TRABAJO DE FIN DE GRADO

Doble Grado en Economía y Estadística

Título: Inteligencia Artificial en los mercados financieros. Consecuencias y aplicaciones.

Autor: Arnau Muns I Orenga

Director: Salvador Torra Porras

Departamento: Econometría, Estadística y Economía Española

Convocatoria: 1er Semestre del curso 2018/2019.

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

DEDICACIÓN

Este trabajo está dedicado a mi familia.

RESUMEN

El término inteligencia artificial es hoy en día un concepto con el que mucha gente ya está familiarizada. En realidad, empieza a estar presente en todas las facetas del mundo. Ya sea de una manera consciente o no, la sociedad del siglo XXI está ya intrínsecamente ligada a los procesos informáticos y a los sistemas automatizados de inteligencia artificial. Se pueden encontrar numerosas aplicaciones en el mundo real de sistemas de inteligencia artificial y machine learning en campos como la medicina, los deportes, la economía o los negocios. La implantación de estos sistemas y, en especial, su auge, se debe al gran crecimiento en cuanto a la disponibilidad de datos que ha experimentado nuestra sociedad de la información. Este tipo de aplicaciones de IA aprovechan este gran volumen de datos para extraer toda la información posible de ellos y, de esta manera, ayudar a elaborar tareas que requieren de una cierta inteligencia humana pero que se han conseguido trasladar a los sistemas informáticos. Desde el proceso de obtención de datos personales que hacen las compañías detrás de las redes sociales, hasta los sistemas de perfilado que tienen los bancos o las compañías de seguros, los sistemas de IA afectan a la vida de cada ser humano en la sociedad globalizada en la que vivimos actualmente, seamos conscientes de ello o no.

En el presente trabajo se pretende elaborar un estudio sobre un sector concreto como lo es el sector financiero. Este sector es uno de los pilares fundamentales de los sistemas económicos actuales y juegan un papel clave en todas las esferas sociales, desde las grandes compañías hasta los principales agentes de consumo, las familias. A causa del papel fundamental e indispensable en la sociedad del siglo XXI de este sector, en el presente trabajo se pretende analizar el impacto que tienen las aplicaciones actuales de inteligencia artificial y sistemas basados en técnicas de machine learning en el sector financiero. En primer lugar se analiza la evolución de la IA desde una perspectiva histórica para posteriormente detallar ejemplos de aplicaciones actuales de IA en el sector financiero. Con el contexto actual en mente se pretende indagar en las consecuencias directas (y que ya están teniendo lugar) de estas aplicaciones, así como analizar desde un prisma económico los potenciales efectos que puede tener la aplicación de la IA en el mundo de las finanzas de continuar su implantación con los ritmos actuales. Anticipando un crecimiento casi exponencial, se proponen distintos problemas que pueden aparecer en un futuro desde perspectivas micro y macro económicas. Posteriormente se desarrollan distintas aplicaciones de machine learning relacionadas con el sector financiero, en especial con la inversión en mercados de activos. Se construye un modelo conceptual de predicción de la dirección de movimiento del precio de cierre de un stock y se analizan distintos tipos de modelos/herramientas de ML con los que construirlo. También se elabora otro tipo de aplicación con redes neuronales recurrentes o profundas (tipo LSTM) sobre los precios de cierre de los activos financieros. Con el objetivo de comparar las distintas herramientas con las que se construyen los modelos de predicción de la dirección de movimiento, se analizan los resultados y se desarrollan las conclusiones a las que se llega, tanto respecto al futuro del desarrollo de la IA en el sector financiero como respecto a los modelos creados.

Palabras clave

Stock, Inteligencia artificial, Aprendizaje automático, Machine Learning, Sector financiero, aprendizaje automático supervisado, aprendizaje automático no supervisado, modelo predictivo, Trading automático (negociación bursátil) precio de apertura, precio de cierre, precio máximo y mínimo.

Clasificación AMS

La clasificación AMS de este trabajo es: 91G99 *Mathematical finance, None of the above but in this field*

ÍNDICE

LISTA DE TABLAS

LISTA DE FIGURAS

CAPÍTULO I

INTRODUCCIÓN

Hoy en día palabras como machine learning o inteligencia artificial, o conceptos relacionados con aplicaciones de los mismos, nos suenan relativamente familiares porque forman parte indiscutible del mundo actual. Aunque no lo parezca, y no sea de conocimiento general, este tipo de aplicaciones que intentan trasladar el pensamiento o razonamiento humano a un lenguaje que los ordenadores sean capaces de entender y reproducir fueron creadas de manera teórica en la segunda mitad del siglo XX, así que en realidad tienen ya sus décadas de vida. Lo sorprendente es que vienen teniendo un auge muy grande en la última decada. Esto se debe a que las limitaciones prácticas que tenían estos modelos en el momento de ser creados se han podido superar en los últimos tiempos. Éstas limitaciones prácticas a las que se hace referencia son básicamente la enorme capacidad computacional requerida para poner en práctica todos estos algoritmos de IA. Ésto es lógico ya que, si en realidad se intenta trasladar el razonamiento humano a un orderador, éste tiene que ser capaz de procesas grandes cantidades de información y realizar operaciones a una alta velocidad. En la época de su creación las máquinas tenían una capacidad computacional muy inferior a la actual, ya que el crecimiento que esta capacidad ha tenido en las últimas décadas ha sido exponencial. La potencia computacional existente hoy en día ha permitido el desarrollo de modelos que previamente no se podían desarrollar a causa de las restricciones físicas en la maquinaria. Además, este aumento de capacidad computacional viene asociado con una reducción de los costes de las máquinas, cosa que potencia la aparición de nuevos elementos de IA. El objetivo del presente trabajo es el de analizar las el contexto actual de la IA en el mundo de las finanzas, empezando con la evolución histórica que viene teniendo la implantación de la IA en el sector financiero hasta la actual situación y aplicaciones. En ese sentido se analiza la situación actual para poder analizar cuáles han sido las consecuencias de la implantación de la inteligencia artificial en los mercados financieros. La motivación presente detrás de este trabajo está clara: en un mundo donde los conceptos relacionados con la IA artificial están de moda, personalmente creía necesaria la elaboración de un estudio riguroso que detallara las aplicaciones actuales y las consecuencias de las mismas.

En esta línea una de las tareas dentro del sector de las finanzas que más se ha intentado desarrollar es la de intentar predecir el precio de un stock, que ha sido una materia objeto de estudio desde los últimos años. Los *brokers* llevan años intentando implementar y mejorar la manera en la que se predicen los precios futuros de los stocks. Este esfuerzo se debe a la potencial utilidad de un modelo preciso en cuanto a la predicción del precio ya que éste

sería indispensable para enriquirse haciendo *trading*. Debido a la complejidad de las series temporales de los precios una nueva rama de investigación se está desarrollando rápidamente. Ésta rama de investigación concentra los esfuerzos en intentar predecir la *dirección* del stock en cuestión en vez de intentar predecir el precio exacto. Estos modelos de predicción de la dirección del movimiento de los stocks, llamados SMD por sus siglas en inglés (Stock Movement Direction), consiguen alcanzar un rendimiento predictivo suficiente como para utilizarlos como complemento al análisi fundamental a la hora de tomar una decisión de inversión. Sin embargo, gracias a la aparición de nuevos modelos de ML tales como las redes neuronales recurrentes, la predicción del precio de cierre en está empezando a parecer plausible, y los modelos que se construyen en esta línea, capaces de ser útiles ayudando en un proceso de toma de decisiones de inversión.

En el trabajo que se desarrolla a continuación se proponen distintas aplicaciones utilizando modelos de IA y machine learning en un sub-sector concreto del sector de las finanzas como es el de los mercados inversiones en activos. De una manera general se pretende mostrar como el abanico de posibilidades en este campo, en cuanto a los distintos tipos de aplicaciones que se pueden realizar, es muy amplio. De una manera más concreta las aplicaciones que se desarrollan a continuación se pueden clasificar en dos grandes grupos. La característica que los distingue es el objetivo final que tiene cada uno de ellos. De una manera técnica se podría decir que los dos grupos de experimentos se distinguen al ser la variable aleatoria que se pretende modelizar de una naturaleza distinta. El primero consiste en el desarrollo de un tipo de modelos concretos, los llamados SMD o modelos de predicción de la dirección de movimiento del precio de un *stock* utilizando un conjunto de indicadores técnicos como variables predictoras. Usando este tipo de modelos se elaboran distintos experimentos los cuales se agrupan en 3 sub-tipos. Se realiza un primer acercamiento detallado utilizando 4 empresas como ejemplo y con un tipo de modelos llamados Random Forest. Posteriormente se aplica el mismo marco conceptual sobre un conjunto más elevado de empresas utilizando computación en paralelo sobre arquitectura distribuida para aumentar la capacidad de cálculo. Finalmente se propone un ejemplo de aplicación de estos modelos conceptuales construyendo otro tipo de modelos de *machine learning*, distintos al Random Forest, utilizando un algoritmo automático de construcción de modelos, concepto conocido en inglés como *automatic machine learning*. El segundo gran grupo de experimentos explora la posibilidad de predecir dos de los más importantes valores a tener en cuenta en un mercado de inversión como son, por un lado, el precio de cierre diario de una determinada acción y, por otro, la rentabilidad diaria asociada a esa misma acción. Este proceso de predicción se elabora utilizando un tipo de modelo concreto de *deep learning* llamado Long-Short Term Memory Recurrent Neural Network. Este tipo de modelos permite conservar dependencias temporales lejanas en el tiempo por lo que se considera un modelo potencialmente útil a la hora de predecir tanto el precio de cierre como la rentabilidad diaria de una acción en un mercado de inversión.

Cabe destacar que todo el conjunto de experimentos que se desarrollan en esta tesis, incluyendo el considerar dos tipos de predicciones conceptuales (dirección y valor), está en línea con la idea de poder utilizar todo este tipo de herramientas a la hora de considerar una decisión de

inversión real. Es decir, todos los experimentos se elaboran teniendo en mente que el objetivo de su elaboración es formar parte, por si mismos o al combinarse con otras herramientas, de una plataforma que permita obtener la información necesaria para poder tomar una decisión de inversión fundamentada. El poder predecir tanto la dirección como el valor mismo que el precio de una acción puede tener dentro de un mercado de inversión de una manera precisa sería sin duda una ayuda fundamental para cualquier persona, profesional o no, que quisiera plantearse una decisión de inversión en un mercado de acciones.

CAPÍTULO II

METODOLOGÍA

En el siguiente apartado se procede a detallar la metodología que se sigue en esta tesis. Para hacerlo se aborda desde 3 perspectivas. En primer lugar se describe la metodología seguida en la redacción del presente documento. En segundo lugar se explica el procedimiento o metodología seguidos en el análisis económico de la situación actual y sus consecuencias. En tercer y último lugar se detalla la metodología seguida durante el proceso de modelado de la dirección de movimiento del precio de cierre.

En cuanto a la redacción del presente documento se ha decidido redactarlo utilizando el software R Markdown. Se ha procedido a descargar una plantilla en LaTeX para la redacción de tesis o artículos académicos y se ha adaptado a las necesidades de la presente tesis. Se ha decidido proceder así para facilitar la interacción entre la escritura de la tesis y el desarrollo de la modelización propuesta, elaborada en R. Utilizando este marco de escritura con R Markdown, controlando la forma y estilo con LaTeX y documento tipo .cls, se consigue una interactividad completa entre el código estadístico escrito en R y la redacción en formato .pdf del presente documento. Técnicamente, la metodología de escritura de este trabajo con R Markdown consiste en separar en un documento a parte cada una de la piezas que conforman la tesis, de manera que se cargan en un documento general que hace las veces de esqueleto vertebrador de la escritura de la tesis. De esta manera se consigue tener cada apartado por separado, potenciando la estructuración y simplicidad de un documento académico que podría ser, de no seguir esta metodología, más difícil de escribir con un software estadístico que con un editor de texto. Por lo que respecta a la definición del estilo de esta tesis, cabe destacar que se ha hecho acorde con las normas definidas por el documento oficial para los trabajos del grado de Estadística.

Todos los documentos y archivos necesarios para compilar y generar la presente tesis en formato .pdf se pueden encontrar en el proyecto "TFG" de mi perfil personal de Git Hub https://github.com/ArnauMunsOrenga/TFG

En cuanto a la metodología que va a seguir el presente trabajo en el análisis económico del marco actual es el siguiente: la investigación académica se elabora a partir del análisis de tesis doctorales, trabajos de final de máster y artículos académicos para servir de base del análisis propuesto. En primer lugar se aborda la evolución histórica de la inteligencia artificial,

poniendo especial atención en las aplicaciones financieras creadas durante el desarrollo de este campo de estudio. En segundo lugar, se utiliza el análisis de documentos de instituciones oficiales y artículos académicos para repasar los *use cases* o aplicaciones actuales de IA en el sector financiero, así como las consecuencias asociadas a los mismos. Finalmente se discuten los posibles efectos a nivel micro y macro económico a partir del análisis de artículos elaborados por instituciones relacionadas con la actividad y estabilidad económicas.

Por último, se detalla la metodología propuesta en cuanto a la modelización que se lleva a cabo. En la figura II.1 se encuentra detallado el proceso de modelado propuesto en el presente trabajo. La variedad y el proceso de investigación llevado a cabo en este trabajo corresponde con la parte de la elección de los modelos, que se desglosa en el capítulo V. Es por eso que los pasos iniciales de la metodología propuesta se comparten entre los distintos procesos de modelado siguientes.

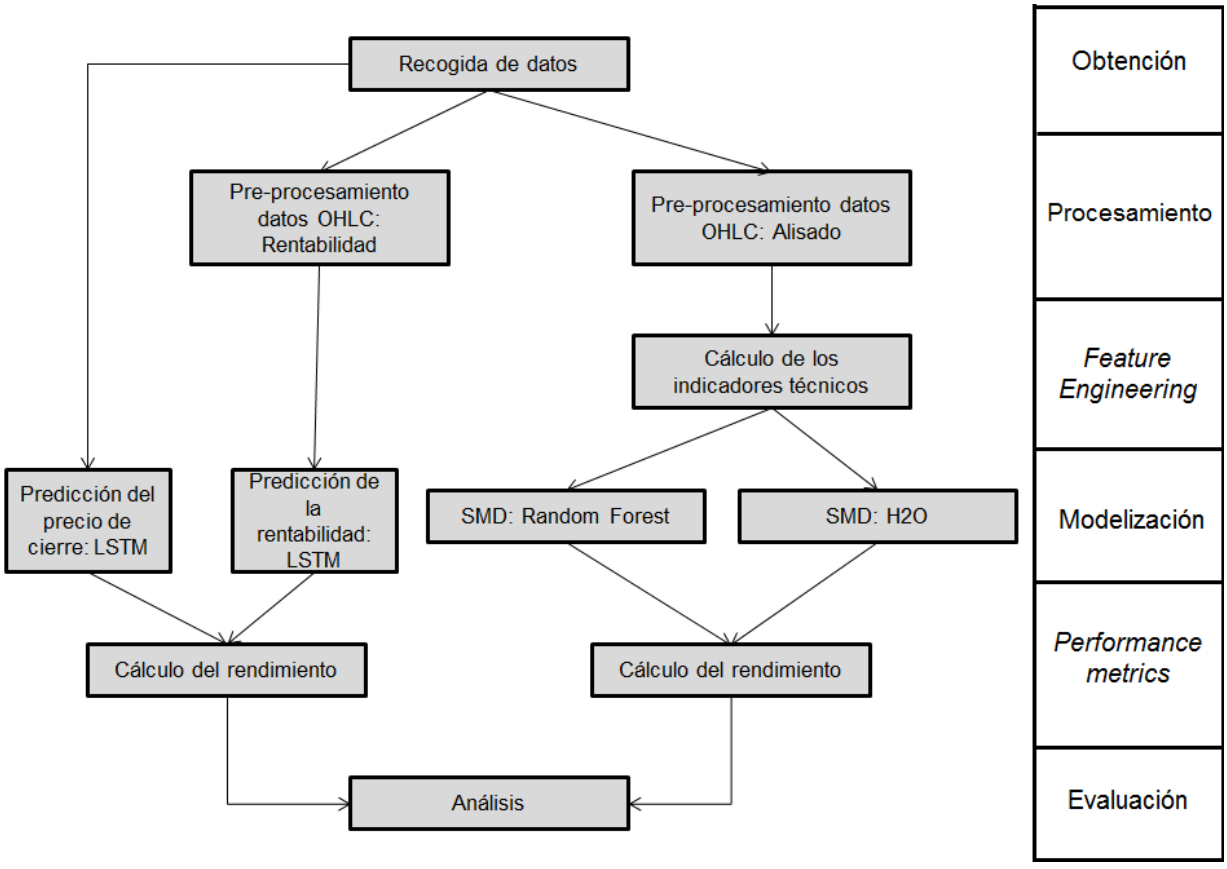


Figura II.1: Metodología propuesta. Fuente: elaboración propia

CAPÍTULO III

CONSECUENCIAS DE LA IMPLANTACIÓN DE LA IA EN LOS MERCADOS FINANCIEROS

En el presente capítulo se procede a analizar las consecuencias de la implantación de la inteligencia artificial en los mercados y el sector financiero. En primer lugar se analiza la proyección histórica y la evolución de la inteligencia artificial como campo de estudio, así como el desarrollo paralelo a lo largo de la historia de las aplicaciones sobre el mundo de las finanzas de la misma. Posteriormente se estudian las aplicaciones actuales de los modelos de machine learning e IA para observar las consecuencias actuales y discutir las posibles consecuencias futuras.

III.1 Evolución histórica y contexto actual

La Inteligencia Artificial, más conocida como AI por sus siglas en inglés, es un campo de estudio que se ha venido desarollando por oleadas los últimos 70 años. Sin embargo, este nuevo campo de la ciencia tiene como fundamentos ideas y técnicas tomadas de otros campos de estudio lárgamente establecidos. Estos otros campos son la filosofía; tomando las ideas de razón, lógica y mente; las matemáticas, la cuál aportó teorías sobre la lógica, deducción e inducción, probabilidad, toma de decisiones y cálculo; la psicología, la linguística y la ciencia computacional (Stuart Russel, 1995).

Nacimiento, primera oleada expansiva y primer invierno

El nacimiento de la inteligencia artificial se puede datar a principios de la década de 1950, en la cual los científicos de la época empezaron a plantearse por primera vez la posibilidad de crear máquinas que pensaran. En este sentido, se empezaron a plantear la idea de crear un cerebro artificial. Este primer periodo de la inteligencia artificial culminará el año 1956 con la conferencia de Dartmouth la cual se puede considerar el nacimiento de la IA al reunir a 11 matemáticos y científicos en lo fué una gran lluvia de ideas alrededor del campo (workshop, 2019).

En este sentido destaca, por ejemplo, la aportación que hizo Alan Turing en la década de los 50. Turing escribió un artículo en el cual especulaba con la posibilidad de crear máquinas que "pensaran". En este sentido se dió cuenta de que "pensar" era un concepto difícil de definir y por ello creó su famoso Test de Turing. Éste era una prueba de la habilidad de una máquina de mostrar un comportamiento inteligente, equivalente o indistiguible, del comportamiento inteligente de un humano. La imagen siguiente ilustra el Test de Turing.

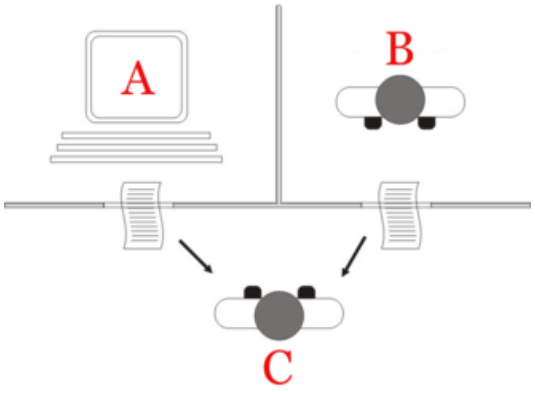


Figura III.1: Diagrama del Test de Turing. Fuente: Wikipedia, adaptado de "Turing Test: 50 Years Later", Saygin, 2000

El Test de Turing consiste básicamente en que el agente C, el interrogador, tiene la tarea de intentar descubrir cuál de los dos interrogados es una máquina, basándose sólamente en respuestas escritas a ciertas preguntas. Si el interrogador no es capaz de distinguir cuál de los dos agentes, A o B, es la máquina, se dice que esta máquina ha pasado el test ya que es capaz de generar respuestas que son muy parecidas a las respuestas que daría un humano. La idea que tenía Turing es que si la máquina es capaz de pasar el test entonces es razonable afirmar que la máquina estaba "pensando", al ser sus respuestas indistinguibles a las que daría un humano.

La época que siguió a la conferencia de Dartmouth 1956 fue una era de descubrimientos en el nuevo campo recién creado. Se desarrollaron los primeros programas capaces de demostrar teoremas de geometría y aprender los primeros idiomas como el Inglés. Sin embargo esta etapa de desarrollo de los primeros programas de inteligencia artificial llegó a su fin a mediados de los años 1970s. La limitada capacidad computacional y potencia de procesamiento de las máquinas de la época dificultó que la expansión continuara, al ser el límite material a las ideas de la época. Pese a que las limitaciones computacionales fueron el principal impedimento para que esta primera ola de desarrollo de la IA continuara, otros problemas derivados como la pérdida de la financiación obtenida durante la primera expansión o las críticas recibidas por parte de otros científicos de distintos campos, como la filosofía, contribuyeron a la finalización de esta primera etapa de expansión (Stuart Russel, 1995).

Segunda oleada

La segunda oleada la podemos ubicar en los años 1980s. Esta etapa está definida por tener a los **sistemas expertos** como centro gravitatorio. A principios de los años 80 este tipo de programas de inteligencia artificial se empezaron a utilizar a nivel empresarial y se popularizaron. Un sistema experto es un tipo de programa de inteligencia artificial que resuelve cuestiones o problemas relacionados con un campo de conocimiento muy específico,

basándose en normas y reglas lógicas derivadas del conocimiento de los expertos en esa materia. Este tipo de programas intentan emular el comportamiento de tendría un experto en un determinado campo de estudio al intentar resolver el problema. Intentan crear, en definitiva, poder computacional "inteligente" que permita suplir al poder cerebral humano (Leondes, 2002). En este sentido fueron los primeros programas de inteligencia artificial que se podían considerar útiles al tener un diseño relativamente sencillo cuyo mantenimiento y desarrollo era relativamente asequible. El alza de los sistemas expertos puso en el centro de la inteligencia artificial el concepto de **conocimiento** y empezaron a plantearse la idea de que la inteligencia podía derivarse del uso intensivo de una gran fuente de conocimiento y la capacidad de utilizarlo e interconectarlo de distintas maneras (Stuart Russel, 1995).

Fue llegados a este punto, con el auge de los sistemas expertos, cuando aparecieron las primeras aplicaciones en el mundo de las financias utilizando este tipo de sistemas computacionales. Uno de los primeros programas que se propuso en el campo de la predicción financiera fue el sistema experto llamado Protrader. Este sistema, desarrollado por Ting-peng Lian y K.C Chen, fue capaz de precedir 87 puntos de caída del índice Dow Jones Industrial Average en 1986. Sus funciones principales eran las de determinar una estrategia de inversión óptima, ejecutar transacciones cuando eran necesarias y modificar la base de su conocimiento mediante un mecanismo de aprendizaje. En la figura 3.2 se puede observar un esquema de la arquitectura que tenía este sistema experto. Más detalles pueden ser encontrados en (Chen, 1989).

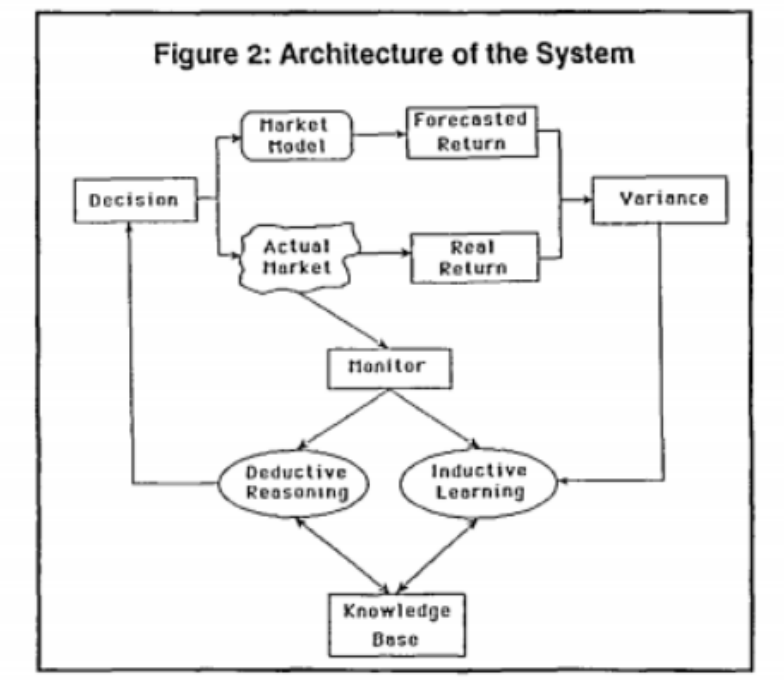


Figura III.2: Esquema de funcionamiento de Protrader. Fuente: Chen, Liang (1989). Protrader: an Expert System for Program Trading

En esta época se desarrollaron y crearon otro tipo de aplicaciones financieras de los sistemas expertos. Podemos encontrar programas en el campo de la auditoría, así como en el de la planificación financiera o los planes de inversión, ahorro y jubilación. A su vez se empezó a explorar la posibilidad de utilizar la inteligencia artificial en el campo de la detección del fraude, especialmente en la decada de 1990. Uno de los programas que fue patrocinado por el departamento del tesoro de Estados Unidos se llamaba FinCEN Artificial Intelligence system (FAIS). Este sistema se puso en funcionamiento en el año 1993 se podía utilizar para determinar casos de blanqueo de capitales (Golberg, 1995). En el diagrama 3.3 se muestra la arquitectura del FAIS. Este sistema era capaz de realizar más de 200.000 transacciones por semana, en una época en la que las transacciones eran transcritas a mano. Por más de dos años el FAIS fue utilizado para detectar 400 casos potenciales de blanqueo de capitales por un total de 1$ billón (Golberg, 1995).

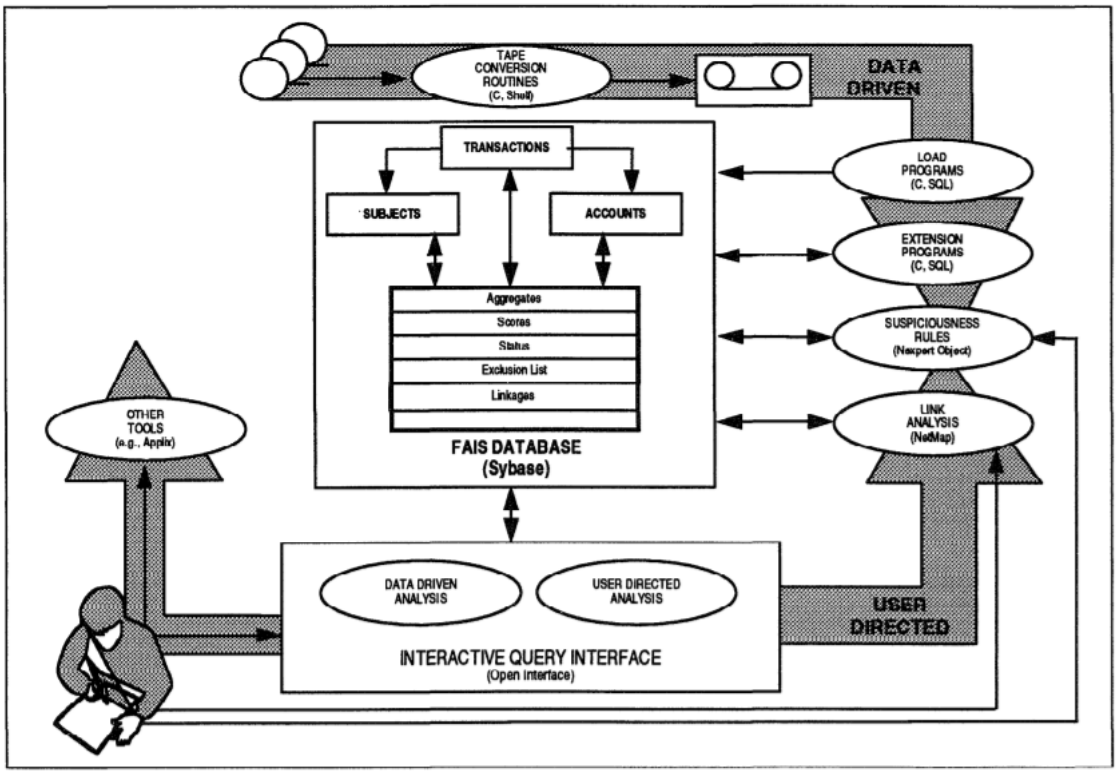


Figura III.3: Esquema de funcionamiento del FAIS. Fuente: informe de la AAAI sobre FinCEN FAIS (1995)

Tercera oleada y actualidad

La tercera expansión de la inteligencia artificial es la que estamos viviendo hoy en día. Empezó a principio de la década de los 90 con la unión de la inteligencia artificial con ideas económicas. La definición de agente racional proveniente de la teoría de la decisión casó con ideas de la ciencia computacional y apareció la definición de agente inteligente. Los

agentes inteligentes son sistemas que perciben el entorno en el que están y toman acciones que maximizan sus probabilidades de éxito. Son programas que vehiculan su actividad en base a la obtención de objetivos dentro de un entorno concreto. Otro ejemplo de los programas desarrollados a principios de los 90 es el sistema conocido como Deep Blue, que fue el primer ordenador entrenado para jugar al ajedrez que pudo batir al campeón mundial de ajedrez, Garry Kasparov (McCorduck, 2004).

Estos avances a principios de la década de los 90 facilitaron la enorme expansión de la inteligencia artificial a principio del siglo XXI. Actualmente, la situación está dominada por el incremento en la potencia computacional de los ordenadores, que permite procesar una cantidad muy elavada de información de distintos tipos (Big Data), así como la utilización de técnicas avanzadas de aprendizaje automático aplicadas de manera exitosa en distintos campos de negocio. Es indudable que la inteligencia artificial forma parte de nuestra vida diaria ya que sus distintas aplicaciones se pueden apreciar de manera clara en la sociedad de hoy en día. En general, los campos exitosos que se están desarrollando en la actualidad son los siguientes:

i) **Deep Learning**: Es un campo del machine learning que permite modelar altos niveles de abstracción en los datos con la construcción de redes neuronales más complejas. Los campos más desarrollados dentro del deep learning son las redes convolucionales profundas y las redes neuronales recurrentes (como las Long Short Term Memory). Su implantación en el mundo real es más que notoria ya que se utilizan de una manera satisfactoria, por ejemplo, en problemas de reconocimiento de imágenes.

ii) **Big Data**: El término Big Data hace referencia al tipo de información que no puede ser capturada, tratada y procesada con los medios de software tradicionales, por lo que es necesario un nuevo paradigma para el tratamiento de este tipo de datos. A los datos Big Data se los suele caracterizar con 5 V's (aunque se pueden encontrar artículos que postulan incluso 8). Éstas son: Volúmen, Velocidad, Variedad, Valor y Veracidad. La V de volúmen hace referencia al tamaño de los datos Big Data, que suele alcanzar magnitudes superiores al TeraByte de capacidad. La Velocidad hace referencia a la alta frecuencia de generación de este tipo de datos, que junto con su elevado volúmen, hace necesaria una nueva manera de capturar estos datos de una manera frecuente y rápida. Variedad hace referencia al hecho de tener distintos tipos de información, todos ellos relevantes. Desde datos en formato tabular hasta imágenes o secuencias de texto. Valor hace referencia al hecho de que sean datos capaces de ayudar a solventar un problema y aportar valor añadido. Por último, la V de veracidad hace referencia al hecho de que sean datos en los que realmente se puedan basar ciertas conclusiones.

La figura 3.4 que se muestra a continuación permite obtener una visión más general de la situación en la que se encuentra la inteligencia artificial actualmente. El gran volúmen de datos disponibles para ser tratados junto con el aumento de la capacidad computacional han

permitido tal evolución que se han desarrollado distintas ramas dentro del mismo. El campo de la inteligencia artificial se puede definir hoy en día como aquél dedicado a desarrollar sistemas computacionales capaces de llevar a cabo tareas que tradicionalmente requerían inteligencia humana para ser llevadas a cabo (Board, 2017). El campo dentro del marco de la inteligencia artificial que más se está desarrollado es el llamado Machine Learning o aprendizaje automático. Se puede definir como el conjunto de métodos y algoritmos los cuales se optimizan a través de la experiencia con intervención limitada o inexistente de un agente humano (Board, 2017), (Jordan & Mitchell, 2015). Estas técnicas, mezcladas con las procedentes del campo del Big Data y el tratamiento masivo de datos, se utilizan para extraer información de valor de conjuntos de datos muy grandes y que suelen incorporar formatos no tabulares de información.

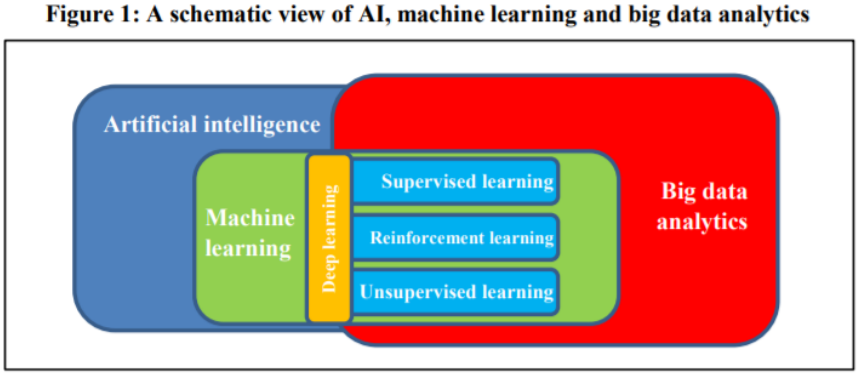


Figura III.4: Esquema AI. Fuente: Artificial intelligence and machine learning in financial services. Financial Stability Board (FSB) 2017

En el mundo actual se pueden encontrar múltiples aplicaciones de inteligencia artificial y modelos de machine learning. En este sentido las empresas llamadas "fintech" ya se encuentran totalmente implantadas en la sociedad actual. Esto es así debido a que son empresas que prestan servicios financieros de una manera más dinámica que el sector bancario o financiero tradicional, apoyándose en las nuevas tecnologías y paradigmas de los sistemas de información de la actualidad. Así mismo, estas empresas están ejerciendo un efecto dinamizador en cuanto al desarrollo y digitalización de los sectores bancarios y financieros más tradicionales. España contaba a finales de 2017 con 300 Fintech y ocupaba la sexta posición en el mundo por número de compañías en el sector. Un informe elaborado por KPMG confirma que el modelo de negocio mayoritario de las Fintech nacionales es el de préstamos (21%), seguido del sector de pagos (19%) y el de inversión (16%) (Funcas, 2017).

III.2 Aplicaciones y consecuencias

En un mundo donde los datos juegan un papel fundamental (hay quién incluso los llama el petróleo del siglo XXI) y el tratamiento de la información está totalmente implantado, todos los ámbitos de la realidad se ven afectados. En el caso del mundo de las finanzas el impacto ha sido profundo. En los últimos años el campo de la inteligencia artificial ha

sido capaz de crear sistemas computacionales y aplicaciones en el campo de las finanzas. En la sección siguiente se examinan los distintos campos del sector financiero en los cuales se aplican actualmente técnicas de machine learning e inteligencia artificial con el objetivo de poder examinar con detalle las consecuencias actuales de su implantación así como las posibles implicaciones futuras. Éstos son: (3.2.1) centrados en el cliente (o *front office*), que incluyen el *credit scoring*, aplicaciones en el campo de los seguros de vida y no vida, y los chat-bots encargados de interactuar con el cliente; (3.2.2) centrados en las operaciones (o *back office*), con casos como los modelos de gestión del riesgo y modelos de impacto de mercado; (3.2.3) gestión de carteras y inversión en mercados financieros; (3.2.4) casos en los que instituciones financieras y empresas privadas aplican la IA y el aprendizaje automático en regulación y supervisión; y (3.2.5) otras aplicaciones.

III.2.1 ***Aplicaciones front office: centradas en el cliente***

Credit scoring

La evaluación del crédito ha sido un método lárgamente usado por los prestamistas a la hora de prestar o no un crédito. La capacidad de evalúar el potencial riesgo del préstamo, otorgándole un valor, de un determinado cliente es esencial para las compañías financieras. La aplicación de técnicas de aprendizaje automático en este sector se está estableciendo lárgamente entre las distintas instituciones encargadas de prestar dinero. Tradicionalmente se han utilizado datos estructurados como transacciones y el historial de pagos para crear modelos tales como regresiones lineales o árboles de decisión para generar un ránking o valoración de crédito. Sin embargo en la actualidad los prestamistas, ya sean bancos o otras comañías, están utilizando ya otras fuentes de datos no estructurados o semi estructurados tales como la actividad en las redes sociales, uso del teléfono móvil o mensajes de texto. Este tipo de datos permite a estas empresas obtener una visión más matizada de la fiabilidad potencial que tendría un préstamo. La aplicación de modelos de IA y ML sobre esta gran variedad de tipos de datos ha permitido a las empresas prestamistas analizar factores cualitativos como el comportamiento del consumo de los clientes o la valoración de la voluntad de pagar el préstamo.

Una de las principales consecuencias de la capacidad de manejar distintas fuentes de información ha sido el hecho de que actualmente es que la segmentación de la calidad de los prestatarios se hace a mayor escala, de una manera más rápida y más barata. En definitiva esto radica en una *decisión de crédito más rápida y acurada* (Stefan Lessmann & Thomas, 2015). Otra de las consecuencias de la aplicación del machine learning en este sector es el hecho de que puede ayudar a garantizar un mayor acceso al crédito. Esto es así ya que los sistemas de evaluación de crédito tradicionales necesitan una cantidad suficiente de datos históricos sobre crédito de esa persona para poder considerarla apta para la evaluación. Si esta información no está disponible, el valor de la evaluación del crédito no puede ser generado y el cliente puede ser potencialmente incapaz de obtener un crédito. Con el uso de fuentes de datos alternativas y la aplicación de modelos de aprendizaje automático los prestamistas son

capaces de llevar a decisiones de crédito que hubieran sido imposibles anteriormente (Board, 2017).

En general se pueden observar ventajas e inconvenientes de utilizar inteligencia artificial o machine learning en el sector de la evaluación del credito. Como ventaja se encuentra el hecho de que la IA permite analizar una cantidad enorme de datos de una manera rápida. Esto implica que el coste de evaluar los riesgos de los potenciales clientes se verá reducido. Además, el hecho de introducir nuevos tipos de datos puede permitir a las compañías el evaluar el riesgo de crédito de individuos para los cuales no se podía evaluar utilizando los datos tradicionales (historial de crédito). Es decir, la falta de historial de crédito o una valoración anterior de crédito ya no serán más un impedimento a la obtención de un crédito ya que otros indicadores de la verosimilitud del pago están siendo utilizados por las compañías.

Sin embargo, el uso de complejos algoritmos de machine learning puede conducir a una falta de transparencia con el consumidor. Cuando se utilizan modelos de machine learning para construir una valoración o puntuación de crédito, es generalmente más difícil el poder ofrecer una explicación sobre esa valoración y la posterior decisión a los clientes, auditores y supervisores. La faceta *black box* que tienen normalmente estos algoritmos lo complica. Además, ciertos autores argumentan que la utilización de fuentes de datos alternaticvas, tales como comportamiento online o fuentes de información financiera no tradicionales, puede introducir **sesgo** en las decisiones de crédito (O'Neil, 2016). En este sentido ciertas asociaciones de consumidores han levantado la voz en cuanto al aspecto moral. Los modelos de machine learning pueden llevar perfiles de prestatario que tengan en consideración la raza o el género. Por ejemplo, estos modelos pueden puntuar a un prestatario de una minoría étnica con un mayor riesgo por defecto sólo porque prestatarios similares han tenido tradicionalmente unas condiciones menos favorables de crédito.

Servicios de seguros

El sector de los seguros es uno de los sectores que más confía en el análisis de datos y la inteligencia artificial para llevar a cabo su negocio. De hecho, el análisi estadístico representa el núcleo fundamental de este tipo de negocio. La capacidad de evalúar el precio de un producto asegurador es esencial para este sector para poder ser rentable. Todas estas técnicas se basan en un análisis masivo de grandes bases de datos que las empresas aseguradoras han venido recolectando desde hace años para evaluar el riesgo de un potencial cliente. De esta manera consiguen ofrecer servicios más baratos a aquellas personas con un menor riesgo potencial o inrcementarlo para aquellas personas con más riesgo. Un claro ejemplo que el lector es posible que haya experimentado es el hecho de ver como incrementa la cuota de su seguro de automóvil después de tener un accidente de tráfico.

Sin embargo, muchas de las aplicaciones actuales de técnicas de machine learning incorporan el análisis de datos desestructurados para mejorar el proceso de suscripción de los seguros, apoyando la asignación de un precio en función de las características del potencial cliente, o para fortalecer estrategias de márketing hacia determinados clientes (segmentación). Ejemplos de estos nuevos tipos de datos son los datos en tiempo real y los datos con alta granularidad. Ejemplos de estos últimos son datos relacionados con el comportamiento de compra online o datos telemétricos provinientes de sensores en aparatos electrónicos. En este sentido, estas empresas empiezan a explorar cómo pueden aplicar la IA y el machine learning sobre datos de sensores remotos, conectados a través del *Internet of Things (IoT)*, para detectar e intentar prevenir accidentes susceptibles de ser asegurados (como accidentes de coche).

Chat Bots

Otra de las aplicaciones actuales de la inteligencia artificial en el campo del *front end* son los llamados ChatBots que se encargan de interactuar con el cliente. Los Chat Bots son programas automáticos que se encargan de asistir y/o ayudar a los clientes en sus transacciones diarias o para resolver problemas. Estos programas utilizan una rama del machine learning llamada NLP (procesamiento del lenguaje natural) para interactuar con los clientes en lenguajes naturales, ya sea por texto o por voz. Este tipo de programas están siendo introducidos por numerosas compañías de servicios financieros, especialmente en sus aplicaciones móviles o las redes sociales, con el fin de agilizar la relación con el cliente y captar nuevas generaciones de clientes.

Actualmente los algortimos Chat Bot que se están utilizando son relativamente sencillos y se centran en informar al cliente y resolver cuestiones sencillas. Sin embargo, los Chat Bots se están moviendo cada vez más hacia las recomendaciones, especialmente en las decisiones financieras importantes. Además, este tipo de modelos permiten a las compañías obtener información de sus clientes gracias a la interacción con estos programas. Ejemplos de sectores que utilizan actualmente los Chat Bots en el mundo financiero son las instituciones financieras y las compañías de seguros. Éstas utilizan los Chat Bots para dar consejos sobre seguros en tiempo real.

Una de las principales consecuencias de la implantación de los Chat bots en el *front office* del sector financiero es algo muy obvio pero a la vez muy importante. Este tipo de programas consiguen reducir las relaciones humanas en el sector financiero. Es posible que en un futuro un gran porcentage de las interacciones entre los clientes y el sector financiero se hagan a través de este tipo de programas, o de otros más complejos que se puedan desarrollar en el futuro.

III.2.2 ***Aplicaciones back office: centradas en las operaciones***

Modelos de gestión de riesgo: validación y stress-tests

El llamado back-testing, o validación de modelos de gestión de riesgo, es importante para el secto bancario porque ha sido utilizado tradicionalmente para evaluar si los modelos de riesgo de los bancos funcionan bien o no. Este tipo de modelos agrupa por ejemplo modelos de gestión de riesgo de crédito, de liquidez, de mercado (tipo de cambio, tipo de interés, cotización…) o riesgo operacional. La validación de modelos se define como el conjunto de procesos y actividades que tienen como objetivo el verificar que los modelos, en este caso de riesgo, están rindiendo como se esperaba, en línea con los objetivos por los cuales se diseñaron y para evaluar su posible impacto (Mitchell, 2016). En este sentido, los bancos están empezando a considerar el machine learning para poder utilizar y hacer que tenga sentido grandes bases de datos estructurados y no estructurados y para analizar el output de los modelos primarios. El hecho de utilizar este gran conjunto de herramientas financieras para realizar el back-testing o validación de modelos permite considerar cambios en el comportamiento de los mercados y otras tendencias, con el objetivo bienintencionado de reducir la potencial infravaloración del riesgo en distintos escenarios financieros (Mitchell, 2016).

Existen algunos ejemplos actuales de bancos que utilizan modelos de aprendizaje automático no supervisados en sus validaciones de modelos. Este tipo de modelos ayudan a los agentes validadores en el monitoreo constante de los test de stress llevados a cabo internamente y de una manera regulatoria, al ser éstos una ayuda para determinar si los modelos de riesgo están rindiendo dentro de los límites aceptables o si se están desviando de su objetivo principal. Además, pueden ofrecer características o variables extra para los modelos de riesgo operacional, tales como la vulnerabilidad de las distintas organizaciones a los ciber ataques (Board, 2017).

A su vez, se está empezando a utilizar la inteligencia artificial y técnicas de machine learning en el campo de los test de estrés bancarios. Estas pruebas de resistencia bancaria consisten en técnicas de simulación que tienen como objeivo determinar la capacidad de estabilidad de una entidad bancaria. Consisten en exponer tanto las carteras de activos como las de pasivos a diferentes situaciones para evaluar las posibles reacciones o consecuencias. Este tipo de pruebas se ha venido utilizando cada vez más después de la crisis financiera global del año 2008. En este caso se utilizan modelos no supervisados de aprendizaje automático para revisar grandes volúmenes de datos con el objetivo de analizar cualquier sesgo en la selección de variables de estos modelos de estrés. La consecuencia directa de la aplicación de la IA en este tipo de pruebas es que conducen inevitablemente a mejores modelos con mayor transparencia.

Modelización del impacto de mercado

El análisis de impacto de mercado consiste en evaluar el efecto que tiene sobre los precios de mercado las acciones de compra/venta (*trading*) que hace una empresa. Para las compañías de *trading* es importante el poder evaluar el impacto que tienen sobre los precios de mercado las operaciones que ejecutan, en especial aquellas operaciones de gran volúmen. En este sentido es esencial para ellas tener una estimación más precisa del impacto que tienen las operaciones que ejecutan de manera que se pueda ajustar la periodicidad de las mismas y minimizar los costes de ejecución de las operaciones. Las compañías financieras están utilizando ya la IA para obtener más información de los modelos que han utilizado históricamente, haciéndolos más fuertes y potentes, así como para ayudar a identificar relaciones no lineales entre las ordenes de compra y venta. Los modelos de machine leaning que se están creando, llamados *trading robots*, se entrenan a ellos mismos para saber cómo reaccionar a los cambios en el mercado (Day, 2017).

Algunos de los ejemplos concretos de herramientas que utilizan el machine learning para modelizar el impacto de mercado son los siguientes. Actualmente se utiliza la IA para identificar grupos de bonos que se comportan de manera similar. De esta manera, las compañías pueden agrupar distintos bonos o activos financieros en grupos utilizando técnicas de *cluster* con el objetivo de poder medir y valorar la liquidez de los bonos de manera individual. Otro de los ejemplos de aplicación de la IA en este campo es el uso que se hace de ella para identificar cómo la sincronización de las operaciones puede minimizar el impacto de mercado. Estos modelos intentar evitar el hecho de programar operaciones muy cercanas en el tiempo con el objetivo de esquivar tener un impacto de mercado mayor que la suma de los impactos individuales. Estos modelos se utilizan para decidir la mejor programación de las operaciones (temporalmente hablando) y para modificar esta programación temporal a medida que la compra venta se va produciendo a tiempo real. Para modelizar estos cambios de utilizan técnincas de aprendizaje automático supervisado.

III.2.3 ***Gestión de carteras e inversión en mercados financieros***

Inversión y trading algorítmico

Otro de los campos en los que se aplica actualmente la IA es en el del *trading algorítmico*. Estos sistemas de machine learning son entrenados con información de grandes bases de datos relacionadas con las condiciones cambiantes del mercado en cuestión y el precio para extraer una decisión de inversión, compra o venta de una posición, y colocarla en el mercado. Aquí entra en juego de nuevo el gran potencial del big data y el machine learning a la hora de procesar un gran volúmen de información de una manera muy rápida, potencialmente en tiempo real. Estos algoritmos están constantemente analizando el mercado y posteando acciones de compra o venta de una posición con una frecuencia muy elevada. Es a causa de la gran velocidad en las interacciones con el mercado generada con este tipo de sistemas que se los mercados tradicionales se están adaptando al llamado *High-Frequency Trading (HFT)*.

El *trading* de alta frecuencia se soporta sobre este tipo de algoritmos de *trading* automático, que permiten alcanzar niveles donde el ser humano no sería nunca capaz de llegar ya que no podemos procesar tal cantidad de información de una manera tan rápida.

Las consecuencias de la implantación de las transacciones bursátiles de alta fracuencia han sido tanto positivas y negativas. En primer lugar una de las principales consecuencias positivas de la aplicación del *trading* algorítmico y la creación del llamado HFT ya ha sido nombrada. Es el hecho de que las transacciones aumentan de velocidad. Al ser transacciones automatizadas que se hacen de una manera rápida en cuanto hay un cambio favorable en el mercado, también aumenta en consecuencia el número de transacciones totales que se realizan en ese mercado, a la par que disminuyen el número de transacciones con un mayor volúmen. En otras palabras, transacciones de menor volúmen y más rápidas. Otra de las consecuencias generales que se pueden apreciar es que el hecho de introducir HFT, y la algorítmica en general, en los mercados financieros de *trading* es que se ha reducido el coste de las transacciones. La razón es sencilla: es más asequible hacer *trading* o negociación bursátil con máquinas que con humanos, ya que las primeras tienes ciertos beneficios tales como no requerir días de vacaciones o no ponerse enfermos (entre otros). Es por eso que el coste el coste de las transacciones se ha venido reduciendo a medida que los mercados se han ido automatizando. El hecho de poder sustituir factor de trabajo humano por factor capital, al reproducir las tareas que anteriormente hacían los *brokers* o *traders*, tiene como consecuencia directa una reducción del coste de las transacciones. En tercer lugar, existen estudios que demuestran que la diferencia entre el precio de compra/oferta y de venta/demanda en un mercado financiero, también conocido como *bid-ask spread*, se ha reducido a causa de la implantación del HFT. Por consecuencia la liquidez, definida como el valor disponible para comprar y vender dentro del rango de precios Bid-Ask, se ha incrementado a lo largo del tiempo (Oliver Linton, 2018, p. 13). Otra de las consecuencias positivas del HFT se puede encontrar en términos de estabilidad y predictibilidad de los mercados. Hay una gran evidencia que sugiere que la eficiencia del precio, en los mercados financieros, se ha incrementado generalmente con el crecimiento de las operaciones bursátiles basadas en computación (Oliver Linton, 2018). Se ha comprovado que los traders de HFT tienden a mover las operaciones en la dirección de los cambios de precios permanentes (eficientes, óptimos) y en la dirección opuesta a los precios transitorios erráticos (Jonathan Brogaard & Riordan, 2013).

Es totalmente cierto que, por otro lado, el hecho de incorporar las transacciones bursátiles de alta frecuencia en los mercados financieros ha traído, y puede provocar en el futuro, ciertas consecuencias no tan deseables. Uno de los principales argumentos que se esgrime en numerosos estudios en contra del HFT es que éste aumenta la volatilidad de los mercados, aludiendo al hecho de que la volatilidad es más elevada en mercados más rápidos. Además, al ser un sistema nuevo, es posible que este tipo de algoritmos, que unitariamente pueden ser estables, terminen funcionando de una manera inconsistente o muy inestable. Sin embargo, existen toda otra serie de estudios, analizados en detalle en (Board, 2017) p.16, que ponen en evidencia una falta de eventos empíricos que soporten esta teoría. En general los datos

no confirman el hecho de que el HFT haga que los mercados sean más volátiles. Otra de las consecuencias negativas que podrían tener graves implicaciones, especialmente en el futuro, es la que se puso de manifiesto el día 6 de Mayo de 2010 en el índice E-mini S&P 500 del mercado de futuros en EEUU. En esta fecha se produjo lo que se llamó un *Flash-Crash*, es decir, un evento intra-día de corta duración en la que se produce un desajuste profundo de los precios que no está generado por ningún cambio en el valor fundamental de los activos que se están comprando/vendiendo. En el caso del *Flash-Crash* del 2010, se ejecutranon en un corto plazo de tiempo muchos algoritmos de HFT con órdenes de venta, motivo por el cual el precio presentó una gran volatilidad durante unos minutos. La figura 3.5 muestra la trayectoria del precio de los futuros en este mercado durante el crash, así como los cambios en los precios entre transacciones consecutivas, medidos en ticks. Gracias a ella podemos observar la gran velocidad con la que cambió el precio y la increíble volatilidad durante este periodo.

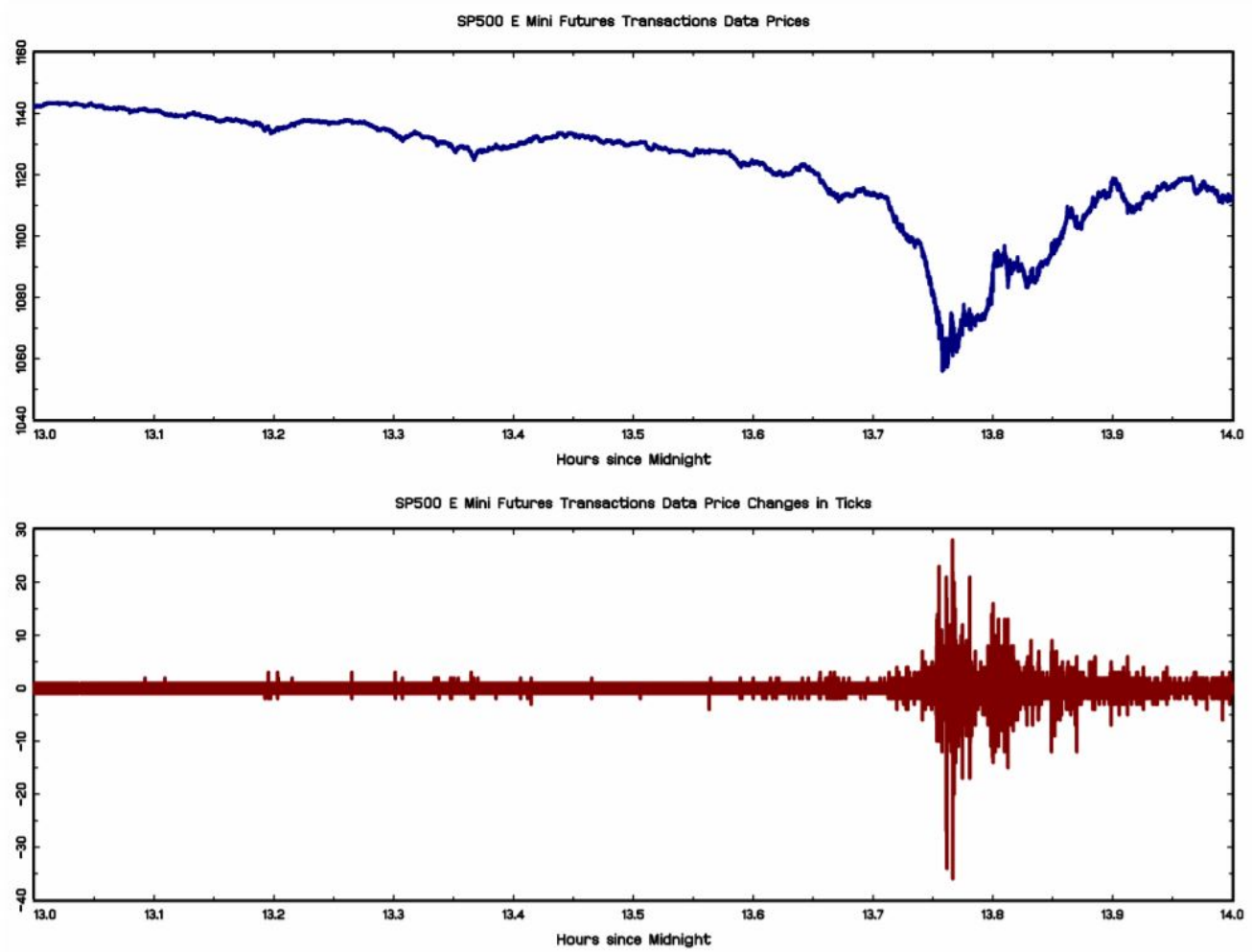


Figura III.5: Nivel de precios de los futuros Emini durante el flash crash y cambios en los precios entre transacciones consecutivas. Fuente: Implications of high-frequency trading for security markets, Oliver Linton, Soheil Mahmoodzadeh 2018

Una de las posibles causas de este *Flash-Crash* se encuentra detalladamente descrita en (Oliver Linton, 2018). En un informe de la SEC/CFTC (2010) se propone como punto inicial una gran orden de venta de 75000 contratos de futuros. Esto provocó que los algoritmos de trading automático empezaran a interactuar entre ellos tratando de absorver el gran volúmen. En general, este informe concluye que los traders de alta frecuencia no fueron los que desencadenaron el *Flash-Crash* pero fueron sus respuestas a la presión de venta inusual generada ese día, ejecutadas a gran velocidad, las que provocaron que la volatilidad del mercado se viera amplificada.

Existen otros ejemplos de *Flash-Crashes* entre los que cabe destacar la comúnmente llamada "Pesadilla en Wall-Street" el 1 de Agosto de 2012. En este caso, la compañia Knight Capital perdió alrededor de 450 millones de dólares en unos pocos minutos a causa de un "error de *trading*" que provocó un un desajuste transversal en el NYSE. Otros ejemplos los podemos encontrar en los mercados de divisas. Es el caso del llamado *Sterling Flash Crash* ocurrido en 2016, donde el par GBP/USD, el tercero más líquido en el mundo, disminuyó el valor un 9.66% en 40 segundos (Oliver Linton, 2018).

En general, las transacciones bursátiles de alta frecuencia pueden mejorar la calidad de los mercados, generando mayor liquidez, reduciendo los márgenes entre precios de oferta y demanda (*bid-ask spread*) y aumentando la eficiencia. No obstante los beneficios que el HFT aporta pueden venir con una serie de costes asociados, como ya se ha destacado. La capacidad de estos sistemas automáticos de interactuar entre ellos de una manera autónoma incrementa el riesgo de que se produzcan eventos extremos que, aun siendo raros y excepcionales, se inician y desarrollan a velocidades muy elevadas. Estas situaciones generan grandes volúmenes de información que requieren largos periodos de tiempo para ser analizados antes de ser entendidos.

Gestión de carteras

El uso de la inteligencia artificial y el machine learning en el sector de la gestión de carteras es considerablemente importante. Estas técnicas se utilizan para sacar provecho de la gran cantidad de datos disponibles en los mercados para identificar nuevas señales en los movimientos de los precios. Sin embargo, estas técnincas se fundamentan en los mismos principios que las técnincas analíticas que ya se utilizaban tradicionalmente. La idea es indentificar señales que permitan hacer predicciones relacionadas con los precios o la volatilidad de los *stocks* para generar rentabilidades más grandes y no correlacionadas. Especialmente se encuentra una gran implantación en los fondos *quant*, la mayoría de los cuales son fondos de cobertura. A su vez se pueden encontrar también ejemplos de compañías que utilizan técnicas de machine learning para recomendar a sus clientes estrategias de inversión personalizadas y automatizadas, de manera que se consigue un servicio personalizado ajustado a las necesidades específicas de cada cliente.

III.2.4 ***Regulación y supervisión***

Detección de fraude

Otra de las aplicaciones actuales es en el campo de la detección de fraudes o anomaías. Los modelos y técnicas de inteligencia artificial ayudan a identificar conductas o comportamientos que se alejan de los patrones regulares. Estos modelos se utilizan por ejemplo para detectar patrones complejos y para hacer hincapié en las transacciones sospechosas que son potencialmente más serias y peligrosas y que por lo tanto requieren una mayor atención y cuidado. Utilizando estas técnicas junto con distintos métodos de machine learning para analizar datos de una manera más desagregada de transacciones, perfiles de clientes y distintos tipos de datos desestructurados, el mundo de la inteligencia artificial se está utilizando para descubrir relaciones no lineales entre los diferentes atributos o características (variables), y para detectar patrones complejos que reflejen potencialmente un blanqueo de capitales (Board, 2017). En este sentido ayuda a agilizar el trabajo de los agentes institucionales encargados de velar por la seguridad financiera. Estos modelos están entrenados con una información que se genera a tiempo real y su utilidad real está en ser capaces de detectar y bloquear una transacción potencialmente sospechosa de fraude para su posterior revisión. Este tipo de sistemas ha permitido ahorrar a las autoridades y empresas un gran volúmen de masa monetaria.

En la actualidad existen numerosos ejemplos de empresas aplicando tecnologías de machine learning. Casos como los de deepsense.ai o feedzai.com son bastante habituales a día de hoy. Estas empresas se dedican a ofrecer soluciones en relación a la detección del fraude a otras empresas. En este modelo B2B (empresa-empresa) permite a los clientes obtener soluciones personalizadas basadas en modelos y técnicas de análisis de datos e inteligencia artificial para poderlos utilizar en su día a día. También existen otros casos de empresas multinacionales que ya han incorporado este tipo de técnicas. Es el caso de Mastercard que lanzó en 2016 su nuevo sistema llamado “Decision Intelligence”. Este sistema intenta resolver la problemática de los falsos positivos de fraude en transacciones genuinas, es decir, verdaderas, utilizando algoritmos y técnicas de machine learning.

Compromiso regulatorio

Otra de las aplicaciones actuales de la inteligencia artificial y el machine learning se puede encontrar en el campo de la regulación financiera donde las instituciones financieras las están aplicando para hacer más potente el compromiso regulatorio. Un ejemplo de las técnicas que éstas utilizan son el procesamiento del lenguaje natural (NLP) para monitorear el comportamiento y la comunicación de los *traders*, con el objetivo de mantener la transparencia y la buena conducta en los mercados. Este tipo de modelos se basan en datos tales como emails, palabra hablada, mensajes intantáneos, documentos y otro tipo de metadatos. A su vez, también se están utilizando los modelos de procesamiento del lenguaje natural para

interpretar, y hacer más comprensibles, regulaciones financieras como el MiFID II. Con esto se consigue el poder automatizar reglas de control basadas en este tipo de regulaciones.

III.2.5 ***Otras aplicaciones***

Text mining y análisis de sentimiento

Otra de las grandes aplicaciones de la inteligencia artificial en general, y en especial en los sectores financieros, es la creación de modelos de procesamiento del lenguaje natural (NLP). Las compañías financieras y empresas FinTech ofrecen servicios basados en sistemas que incluyen el procesamiento de todo tipo de noticias, textos en redes sociales (social media) y arículos. La creción de regresores a partir de información escrita potencia la capacidad predictiva de los modelos, haciendo que sea aplicado hoy en día por numerosas empresas de servicios de inversión. No son extraño casos como las compañías AlphaSense, Kensho o Serimag, que ofrece servicios tales como verificación de documentos de identificación on-line o la detección de cláusulas abusivas en las escritura o contrato de una hipoteca (https://serimag.com/en/case-studies/). Este tipo de modelos consigue procesar una cantidad muy elevada de información, y como se trata de leer documentos escritos, sobrepasan enormemente la capacidad de lectura del ser humano. Es por eso que su aplicación sigue creciendo dentro del sector financiero y se desarrolla en nuevas herramientas o sistemas que utilizan esta técnica.

Otra de las aplicaciones del NLP es el análisis de sentimiento. Este conjunto de técnicas se puede considerar un subtipo de análisis dentro del gran campo de estudio dentro del procesamiento del lenguaje natural. Se utiliza para añadir a los modelos predictivos el punto de sentimiento que se deriva de los textos, en general datos de redes sociales y noticias. De esta manera se pueden construir modelos que procesen datos a tiempo real que se pueden utilizar como proxy del estado de animo o sentimiento de ciertos actores económicos, utilizándolos para la predicción, por ejemplo, de la dirección de los mercados o para definir estrategias de inversión.

III.3 Análisis económico: posibles efectos

Una vez analizadas las diferentes aplicaciones actuales de la inteligencia artificial en el sector de las finanzas, y sus consecuencias más directas, se procede a elaborar un análisis de las posibles efectos o implicaciones de su implantación desde una perspectiva económica. Concretamente se distribuye el análisis en dos partes: por un lado desde una perspectiva micro económica (o micro financiera) y, por otro, desde una perspectiva macro económica.

III.3.1 ***Análisis micro-económico***

En primer lugar se analizan los posibles efectos de la inteligencia artificial en los mercados financieros. Una de los conocimientos globales que se tiene actualmente es el hecho de que la IA y el machine learning tienen el potencial para mejorar substancialmente la eficiencia

en el procesamiento de información. Eso deriva intevitablemente en una reducción de las asimetrías en la información que tienen los agentes económicos y por lo tanto puede reforzar la función de información del sistema financiero. Los mecanismos por los cuales este refuerzo podría ocurrir incluyen los siguientes factores. Por un lado, estos sistemas de IA puede permitir que ciertos agentes en el mercado recojan y analicen datos a mayor escala. Con este análisis los participantes del mercado son potencialmente capaces de entender (mejor) la relación entre distintos factores y la formulación de los precios de mercado. Ejemplos de los distintos factores que se pueden relacionar con el proceso de definición de los precios de mercado son los sentimientos o estádos de ánimo extraídos de las redes sociales o las notícias gracias a técnicas de *sentiment analysis*. Esto podría reducir las asimetrías en la información y de esta manera contribuir a la eficiencia y estabilidad de los mercados (véase (Securities Commissions, 2017) p. 28). Por otro lado los modelos de machine learning e IA pueden beneficiar a los participantes en un mercado al reducir los costes asociados a su participación. Además la IA pude permitir a los agentes ajustar sus estrategias de inversión de una manera casi inmediata (o en tiempo real), como respuesta a una realidad cambiante. De esta manera se reducen en general los costes de las transacciones en el sistema a la par que se mejora el descubrimiento de los precios (Board, 2017). Paralelamente se observan otras posibilidades. Si los participantes en los mercados terminan pot utilizar técnicas similares de machine learning los riesgos correlacionados podrían derivar en riesgos de estabilidad de los mercados. Es decir, si el uso de *traders* basados en modelos de aprendizaje automático se empieza a ser mejor, de una manera masiva, que el *trading* clásico, esto puede provocar que muchos agentes en el futuro decidan también utilizar este tipo de técnicas y esdevengan, en definitiva, un sistema que puede amplificar los *shocks* financieros, tanto positivos como negativos.

En segundo lugar se analizar los posibles efectos de la IA en las instituciones financieras. Por un lado los sistemas de machine learning pueden mejorar los ingresos y reducir los costes para las mismas, ya que pueden mejorar el procesamiento de distintas operaciones que este tipo de insituciones utilizan. Al tener éstos un gran componente computacional, el hecho de incorporar el machine learning para automatizar y mejorar procesos de negocio rutinarios puede trasladarse en unos costes operacionales más bajos y, en general, hacen que estas empresas sean más rentables. Por otro lado, y como ya se ha analizado en la presente tesis, la IA y el machine learning se pueden utilizar y se utilizan para la gestión del riesgo. Por ejemplo, técnicas y modelos de gestión de riesgos de cola (distribuciones de riesgos con las colas pesadas no normales). Además también se pueden utilizar para detectar el fraude bancario o financiero, las transacciones sospechosas y estimando el riesgo de ciber ataques mejorando, en definitiva, el proceso de gestión del riesgo. Sin embargo todas estas herramientas tienen un peligro potencial ya que pueden pasar por alto nuevas fuentes de riesgo al no estar presentes en los datos históricos utilizados para entrenar los modelos. Otro de los posibles efectos sobre las instituciones financieras se analiza seguidamente. El hecho de que el desarrollo en el campo de la inteligencia artificial sea de carácter *open source* y con un uso intensivo de datos puede animar a las instituciones financieras a colaborar con otras e incluso con otros sectores como el comercio on-line y las economías colaborativas. En este sentido fomenta incrementar y potencial la colaboración inter-sectorial.

En general, en el caso de las instituciones financieras reguladoras, la aplicación de la IA puede beneficiar al sector en general ayudando a mejorar los sistemas actuales de control, gestión y regulación. No obstante todos estos efectos positivos vienen con unos riesgos potenciales asociados. La creación de modelos de toma de decisiones automáticas, que normalmente se convierten en cajas negras, puede hacer dificil para las instituciones financieras el captar cómo se formulan las decisiones, especialmente en casos de inversión y *trading* (véase (Knight, 2017) para una descripción concisa de los problemas de las *black boxes* en la IA y la toma de decisiones). Esto puede derivar en un serio problema cuando estos algoritmos causan situaciones de *Flash Crush* o cuando están relacionados con eventos extremos. Esto sin duda significa una falta de transparencia en torno a los sistemas de IA que puede ser problemático para las instituciones a la hora de entender cómo y por qué han ocurrido los eventos exremos o no deseados derivados de la aplicación de los mismos.

Otro de los potenciales riesgos de la IA es la falta de transparencia en cuanto a las responsabilidades derivadas de las posibles pérdidas económicas en los intermediaros que estos modelos puedan causar. Por ejemplo en el caso de que una aplicación de IA desarrollada por un tercero provoque grandes pérdidas. En este caso, es la institución que provoca las gestión o movimiento la única responsable de las pérdidas? O serán las instituciones capaces de gestionar las potenciales demandas en contra de los desarrolladores de la aplicación? En este sentido el debate está abiero, y habrá que seguir en el futuro las noticias relacionadas con este tema. En este sentido, es posible que la aplicación transversal de la IA en el sector financiero provoque cambios en la manera en la que se regula este sector. Por último, cabe destacar las grandes dependencias sobre terceros agentes que la aplicación de la IA en finanzas tiene. El desarrollo de estas aplicaciones se sostiene en general gracias a un gran número de empresas tecnológicamente avanzadas que son las proveedoras de estos servicios. Si las grandes instituciones financieras confían tanto en los terceros agentes, se pueden generar grandes disrupciones en el sistema si estos terceros agentes experimentan situaciones de alto riesgo o quiebras. Especialmente en el futuro habrá que tener presente estos riesgos a la hora de desarrollar herramientas de IA encargadas de misiones o tareas consideradas críticas (Board, 2017).

En tercer lugar se analizan los potenciales efectos de la IA en los consumidores e inversores. En general, ya se ha visto que la IA y el machine learning tienen el potencial de reducir costes y de aumentar la eficiencia de los servicios financieros y, por tanto, los consumidores pueden verse beneficiados de distintas maneras. Uno de los beneficios directos de la reducción de costes es que ésta puede verse trasladada a los consumidores en forma de menores tasas y costes de transacción. Además los consumidores e inversores pueden tener acceso a una mayor variedad de servicios financieros. A su vez también pueden facilitar unos servicios financieros mucho más customizados y personalizados ya que la IA aplicada al big data (*big data science*) permite a las compañías analizar las características de los clientes y, así, poder ofrecer servicios ajustados a diferentes perfiles de cliente. Sin embargo, la utilización de estos datos de los consumidores puede llevar a problemas de privacidad de datos y seguridad de la información (véase (Kuroda, 2016) para más información acerca de los problemas relacionados

con la privacidad de los datos y IA). Gracias al uso intensivo de este tipo de datos privados los modelos de machine learning pueden llegar a ser discriminatorios por razones de raza, sexo, religión, etc. Evitar este tipo de modelos discriminatorios en ambientes como la evaluación del riesgo de crédito, modelos de acceso a crédito o calculadores de cuotas de servicios de seguros.

III.3.2 ***Análisis macro-económico***

En segundo lugar en el presente apartado se procede a analizar las posibles consecuencias y efectos generados de la aplicación de la IA en el sector financiero pero desde una perspectiva macro económica.

Para empezar se analiza el potencial impacto desde la perspectiva del crecimiento económico. Se puede observar que los servicios financieros apoyados en técnicas de machine learning y de IA tienen el potencial de potenciar la eficiencia de la economía y de contribuir al crecimiento económico a través de los siguientes mecanismos (Wilkins, 2017):

i) Fortaleciendo la eficiencia de los servicios financieros, gracias a una gestión más eficiente del riesgo que puede beneficiar al sistema de una manera agregada. A su vez, gracias a la ayuda que brinda a la hora de procesar información para el cálculo del valor fundamental de los activos, el machine learning puede ayudar a alinear mejor los fondos hacia los inversores y los proyectos de una manera más efectiva. Además el hecho de que el machine learning aplicado a los servicios financieros puede reducir el coste e incrementar la velocidad de las transacciones puede estimular las transacciones para las actividades económicas reales.

ii) Facilitando la colaboración entre servicios financieros y distintas indústrias, que puede llegar a crear nuevos sectores o economías/servicios que, a su vez, ayude al crecimiento global de la economía. Un claro ejemplo de este punto es la colaboración entre el sector del comercio electónico y las economías basadas en el concepto de compartir (*sharing economies*).

iii) Estimulando la inversión en el propio sector de la inteligencia artificial y las áreas relacionadas. El hecho de que muchas compañías, no sólo del sector financiero, estén dispuestos a aplicar este tipo de herramientas o sistemas de IA genera mayor inversión en el sector que a su vez se traslada en una mayor inversión inter-sectorial y, así, en mayor crecimiento.

Desde otras perspectivas macro económicas se destacan los siguiente. En primer lugar, con la aplicación de la IA en el mercado financiero es posible que afecte a los tipos y grados de **concentración del capital** en los distintos mercados financieros. Por ejemplo, el hecho

de que todos estos sistemas de IA estés intrínsecamente ligados al desarrollo tecnológico puede provocar que el acceso a las más nuevas tecnologías en, por ejemplo, el sector de la informática o la infraestructura Big Data, quede restringido a empresas de gran capital al ser las únicas capaces de poder afrontar el coste. Otro ejemplo podría ser que la aparición de un grupo reducido de compañías ofreciendo servicios avanzados de machine learning podría incrementar la concentración en el sector de cierto tipo de servicio financiero. Por último se destaca el hecho de que las aplicaciones de inteligencia artificial tienen el potencial para fortalecer la interconectividad de los mercados y las instituciones financieras con consecuencias inesperadas. Por ejemplo, el uso cada vez más frecuente de las instituciones del Big Data puede generar mayores dependencias en variables macroeconómicas que antes no se relacionabas, extraídas a partir de sectores no financieros como el e-commerce. Sin embargo, si un sector crítico de las instituciones financieras utiliza las mismas fuentes de datos y las mismas estrategias algorítmicas, entonces es posible que, bajo ciertas conficiones del mercado, un problema en esas fuentes o algoritmos pueda afectar al mismo sector como si fuera un único nodo, actuando como catalizadores del *shock*.

CAPÍTULO IV

CASOS PRÁCTICOS: MARCO

Una vez analizada la situación actual de la aplicación de la IA en los mercados financieros, en cuanto a los *use cases*, consecuencias y posibles efectos de la misma, se proceden a elaborar distintas aplicaciones basadas en modelos estadísticos y de machine learning. La motivación para la parte práctica del presente trabajo es la crear distintos modelos que puedan llegar a tener una aplicación práctica en el sector de las finanzas, especialmente en el campo de la inversión en activos. Los modelos que se desarrollan a continuación son únicamente un ejemplo de la gran variaded de técnincas, y sobretodo de maneras de enfocarlas, que se pueden aplicar en este campo del sector financiero.

Esta sección se estructura de la siguiente manera. En primer lugar se describen teóricamente los modelos estadísticos y de machine learning que se utilizaran posteriormente en los diferentes planteamientos de los usos prácticos, así como las métricas de rendimiento que se utilizan para evaluarlos. Posteriormente, cinco apartados indagan en los cinco tipos de aplicaciones de IA que se desarrollan en el presente trabajo, elaborando las conclusiones extraídas a partir del análisis en el mismo apartado

IV.1 Definiciones de los modelos

Random Forest

Los árboles de decisión CART, llamados así por su nombre en inglés `Classification and Regression Trees`, son un tipo de modelos que se pueden utilizar para distintos tipos de aplicaciones de aprendizaje automático. En resumidas palabras, el método consiste en partir los datos a partir de un valor de cierta variable. Cada nodo padre genera 2 nodos hijos al tratarse de un problema de clasificación donde la variable respuesta tiene 2 clases. Esta partición se hace a partir de un criterio de impureza de los datos de manera que los nodos finales, llamados hojas, tengan la mayor pureza posible. Sin embargo los árboles que se hacen crecer de una manera muy profunda, es decir árboles muy grandes en cuanto al número de *split* y la profundidad que cogen, para aprender patrones altamente irregulares tienden a sobreajustar los datos de entrenamiento (problema conocido en inglés como *overfitting*). Un ligero ruido en los datos puede causar que el árbol crezca de una manera completamente diferente (Luckyson Khaidem, 2016, p. 7). Esto se debe al hecho de que los árboles de decisión tienen poco sesgo pero una alta varianza, al hacer pocas asumpciones sobre la variable

respuesta (sesgo) pero altamente susceptibles a las variables predictoras (varianza). En otras palabras, un árbol de decisión casi no hace asumpciones sobre la variable objetivo (sesgo pequeño) pero es altamente susceptible a variaciones de los datos que se utilizan como input (alta varianza). Seguidamente se muestra una ejemplo sencillo de cómo un árbol de decisión luce. Esta imagen representa el ejemplo en el que se quiere predecir el género de una persona a partir de se altura y su peso.

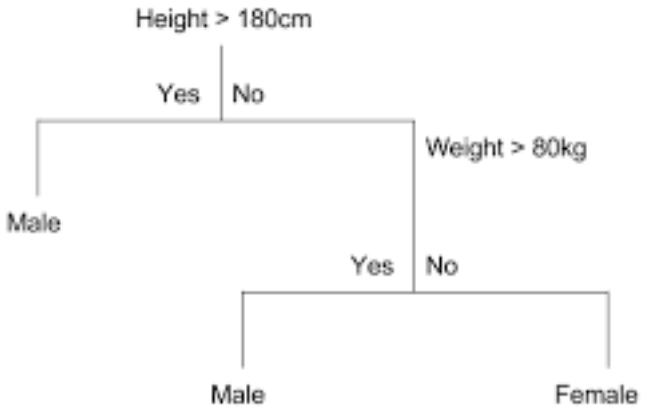


Figura IV.1: Ejemplo de árbol de decisión. Fuente: Classification And Regression Trees for Machine Learning. Jason Brownlee, 2016

En este punto es cuando aparece el modelo de aprendizaje automático llamado `Random Forest`. Este tipo de modelos superan el problema explicado en el párrafo anterior entrenando múltiples árboles de decisión en un subespacio del espacio formado por las variables predictoras/explicativas, asumiendo como coste un ligero incremento del sesgo. Esto significa que ninguno de los árboles del bosque es entrenado con la totalidad de los datos de entrenamiento. Los datos son recursivamente partidos en particiones, de manera que en un nodo particular la partición se elabora haciendo una "pregunta" a una de las variables. Por ejemplo, una partición podría estar hecha "preguntándole" a la variable *Rate of Change 1 day* cuántos datos tienen un valor superior/inferior a un cierto valor X. La elección del criterio de partición de los datos se basa en alguna medida de impureza tales como la Entropía de Shannon o la medida de impureza de Gini. En el presente trabajo se utiliza la función `randomForest` implementada en el paquete de R con el mismo nombre. Esta función utiliza como medida de impureza a partir de la cual se particionan los datos el índice de impureza de Gini (Liaw & Wiener, 2002). Este índice se utiliza como la función para medir la calidad de cada partición en cada nodo. La impureza de Gini en el nodo N se calcula a partir de la fórmula siguiente:

$$g(N) = \sum_{i \neq j} P(w_i)P(w_j)$$

Dónde $P(w_i)$ es la proporción de casos donde la variable respuesta toma la clase i. $P(w_j)$ entonces es la propoción de casos en los cuales la variable respuesta toma la clase j.

La manera heurística para escoger la mejor decisión de particion en un nodo específico se basa en el hecho de conseguir la mayor reducción posible de impureza. Es decir, la mejor partición posible en un determinado nodo viene definida por la mayor ganancia de información (variable que mejor particiona los datos / inclye más observaciones en cada partición) o por la mayor reducción de impureza. La ganancia de información que genera una determinada partición se puede calcular con la fórmula siguiente:

$$\triangle I(N) = I(N) - P_L * I(N_L) - P_R * I(N_L)$$

Dónde $I(N)$ es la medida de impureza (ya sea la impureza de Gini o la entropía de Shannon) de un nodo N. P_L es la proporción de casos que en el nodo padre N van a parar al hijo **izquierdo**. De un modo similar, P_R representa la proporción de casos en el nodo padre N que se van a parar al nodo hijo **derecho** después de realizar la partición. N_L y N_R son los nodos hijos izquierdo y derecho, respectivamente.

Este tipo de modelos de aprendizaje automático son conocidos como modelos *ensemble*. En el núcleo de estos modelos está el *Bootstrap aggregating*, mayormente conocido como *bagging*. Esto significa que la predicción final se calcula como una media de la solución obtenida con cada árbol construido sobre cada remuestra generada con la técnica no paramétrica del *bootstrap*. En otras palabras: utilizando *bootstrap* se calculan remuestras de los datos con las cuales se contruye un árbol. Dentro de cada árbol calculado sobre cada remuestra *bootstrap* cada nodo es partido utilizando la mejor variable dentro de la muestra de variables escogidas aleatoriamente en cada nodo. Al final, la predicción del modelo es una media de los valores obtenidos con todos estos árboles calculados sobre las distintas remuestras *bootstrap* (véase (Liaw & Wiener, 2002) p. 18 y (Dietterich, n.d.)). El método *bagging* mejora la estabilidad y la precisión de los algoritmos de aprendizaje. Al mismo tiempo reduce la varianza y el sobreajuste, los cuales son un problema relativamente común al construir árboles de decisión CART (véase Luckyson Khaidem, 2016, p. 8 para un resumen del algoritmo escrito en pseudocódigo).

Modelos en h2o: AutoML

En el apartado V.3 del presente trabajo se utiliza la función `AutoML` del paquete `h2o` para hacer un análisis automático sobre el modelo conceptual de predicción de la dirección de movimiento del precio de cierre. Los modelos que esta función prueba son los siguientes:

i) *Distributed Random Forest.* Este tipo de modelos incluyen tanto los modelos Random Forest descritos en el presente apartado como los modelos llamados Extremely Randomized Trees (Árboles aleatorizados de manera extrema). La diferencia entre este tipo de modelos (XRT) y los modelos Random Forest clásicos es que se añade otra capa de aleatoriedad en la manera en la que las particiones en cada nodo se calcula. Como en los Random Forest, se utiliza un subconjuto aleatorio de variables predictoras como candidatas en cada partición pero en vez de buscar el umbral que mejor discrimina (o parte) la base de datos, el umbral se construye aleatoriamente para cada variable candidata y el mejor de estos umbrales generados aleatoriamente se coge como la regla de partición. Esto normalmente permite reducir un poco más la varianza del modelo a expensas de un ligero incremento en el sesgo.

ii) *Generalized Linear Models.* Este tipo de modelos es ampliamente conocido. En este caso, al tratarse de un problema de clasificación, el modelo lineal generalizado que se prueba con la función `AutoML` es una regresión logística con variable respuesta binaria.

iii) *Extreme Gradient Boosting (Gradient Boosting Machine).* Este tipo de modelos son considerados como modelos *ensemble*. Consiste en construir árboles de clasificación de manera paralelizada para después hacer un *ensemble* de ellos. Son en esencia modelos que se construyen por árboles de clasificación, cada uno construido con computación en paralelo.

iv) *DeepLearning (Fully-connected multi-layer artificial neural network).* El modelo de deep learning que aplica el paquete *h2o* está basado en una *multi-layer feedforward artificial neural network* que se entrena utilizando el método de optimización llamado *stochastic gradient descent* utilizando *back-propagation*.

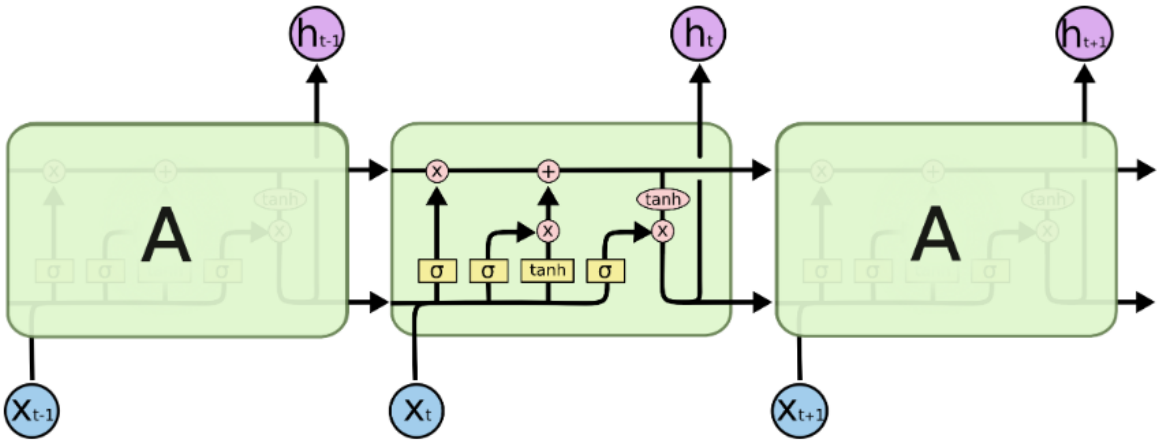
v) *Stacked Ensemble.* El objetivo de los modelos *ensemble* de machine learning es el de utilizar distintos modelos de aprendizaje automático para obtener un rendimiento mayor en cuanto a capacidad predictiva que la que podría obtener ninguno de los

modelos individuales por si solo. De hecho, muchos de los modelos actuales de machine learning que se han vuelto populares son modelos *ensemble*. Por ejemplo el mismo modelo Random Forest o las máquinas de gradientes potenciadas (Gradient Boosting Machine) son modelos *ensemble* que utilizan métodos distintos para hacer este *ensemble*. Los modelos Random Forest utilizan el método *bagging (bootstrap aggregating)* mientras que los GBM utilizan el método del *boosting*. Lo que permiten estos métodos es el coger un grupo de modelos más "débiles" (por ejemplo los árboles de decisión) para crear un único modelo más potente.

Long-Short Term Memory Recurrent Neural Network (LSTM)

Este tipo de redes neuronales recurrentes, llamadas simplemente LSTM, permiten resolver el problema que se planta con las redes neuronales recurrentes tradicionales en cuanto a las dependencias temporales alejadas en el tiempo. Las redes tradicionales presentan el inconveniente de que no son capaces de recordar largas dependencias temporales. Este tipo de redes, las LSTM, fueron introducidas por Hochreiter & Schmidhuber en 1997 y permiten solventar el problema de las dependencias temporales grandes al permitir recordar información durante largos periodos de tiempo. Para ayudar a la explicación de cómo funcionan este tipo de redes neuronales se utilizan los gráficos creados por (Olah, 2015), que utilizan una ilustración muy clara que permite una explicación intuitiva de este tipo de modelos sin centrarse totalmente en la formalidad matemática.

En primer lugar se presenta el esquema básico de una RNN LSTM. Como todas las redes neuronales recurrentes se puede representar el esquema como una cadena de módulos de red neuronal. La diferencia entre las RNN clásicas y las LSTM es que su estructura interna es diferente ya que tienen 4 redes neuronales internas en cada módulo en vez de una simple capa.



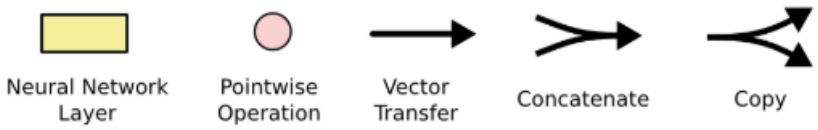


Figura IV.2: Estructura general de una red neuronal recurrente LSTM. Fuente: (Olah, 2015)

La idea fundamental de este tipo de redes neuronales radica en el llamado *cell state* que en este caso está representado con la línea que recorre la parte superior de la figura IV.3. Este *cell state* es el flujo de datos entre el módulo previo de la LSTM y el módulo siguiente, y se ve afectado por la red LSTM que quita o añde información a este flujo regulando este proceso a través de las estructuras llamadas puertas o *gates*. Estas puertas son redes neuronales que deciden que cantidad de información previa va a afectar a este flujo de datos o *cell state*. Un módulo de LSTM tiene 4 puertas: 3 sigmoidales, que se encargan de decidir que porcentage de información se mantiene y cuánto se olvida y una tanh que transforma el rango de los valores entre -1 y 1.

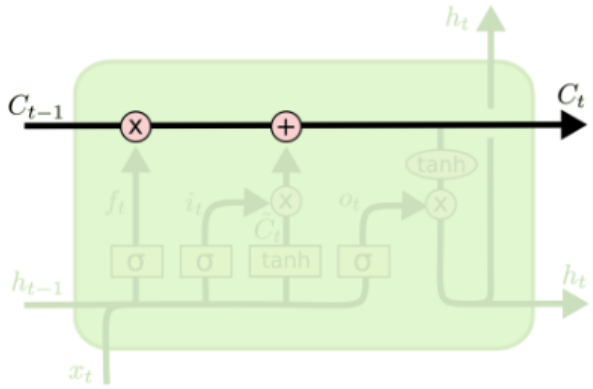
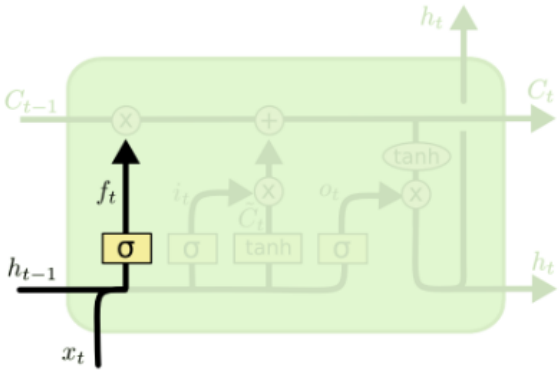


Figura IV.3: Cell state. Cantidad de información que fluye a través de un módulo de la LSTM. Fuente: (Olah, 2015)

Primer paso

El primer paso en la LSTM es decidir que información de C_{t-1} se olvida, basado en h_{t-1} y x_t. A esta puerta se la conoce como la "puerta del olvido" y el proceso se elabora a través de una capa sigmoidal, que decide qué partes de la información de C_{t-1} hay que olvidar. Esta capa utiliza h_{t-1} y x_t para sacar como output un valor entre 0 y 1 por cada número en el *cell state* C_{t-1}. Un 1 representa "recuerda totalmente este elemento" mientras que un 0 representa "olvida totalmente este elemento".

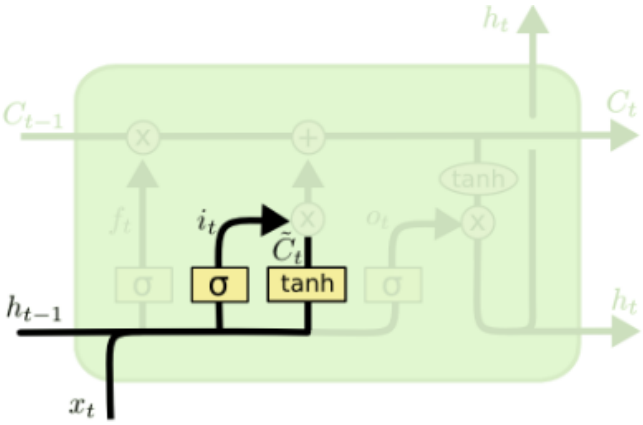


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Figura IV.4: Primer paso o puerta en la LSTM. Fuente: (Olah, 2015)

Segundo paso

En el segundo paso se decide qué nueva información se procede a guardar en el *cell state*. En primer lugar se encuentra una capa sigmoidal llamada la "puerta de entrada" o "*input gate layer*" que decide qué valores se procede a actualizar. A continuación una capa con la función **tanh** crea un vector con los nuevos valores candidatos $\tilde{C}_t$ que pyede ser añadido al *cell state*. En otras palabras el proceso que tiene lugar en el segundo paso es el de decidir que valores se van a actualizar y con que valor concreto lo van a hacer. En el tercer paso se combinan estas dos puertas o capas para actualizar el *cell state*.



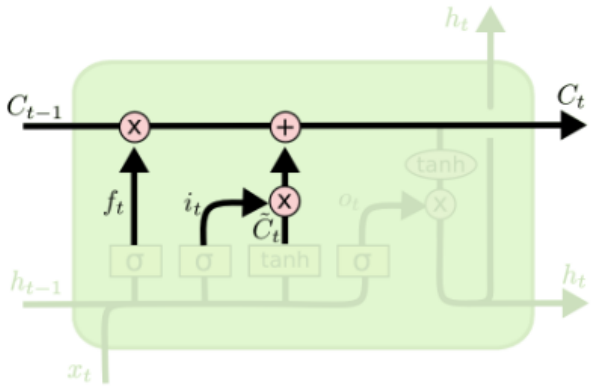
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$

$$\tilde{C}_t = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$$

Figura IV.5: Segundo paso en la LSTM. Puertas 2 y 3. Fuente: (Olah, 2015)

Tercer paso

El tercer paso consiste en actualizar el antiguo *cell state*, o C_{t-1}. En los pasos 1 y 2 ya se ha decidido qué información olvidar y qué valores actualizar, junto con el valor concreto que deben tener. Queda por tanto el trabajo de aplicar dichos cambios. Se multiplica el antiguo C_{t-1} por f_t, olvidando las partes que se ha decidido previamente olvidar. Posteriormente se añade $i_t * \tilde{C}_t$, lo que crea los nuevos valores candidatos escalador por el valor que indica cuánto se ha decidio actualizar cada valor del *cell state*.

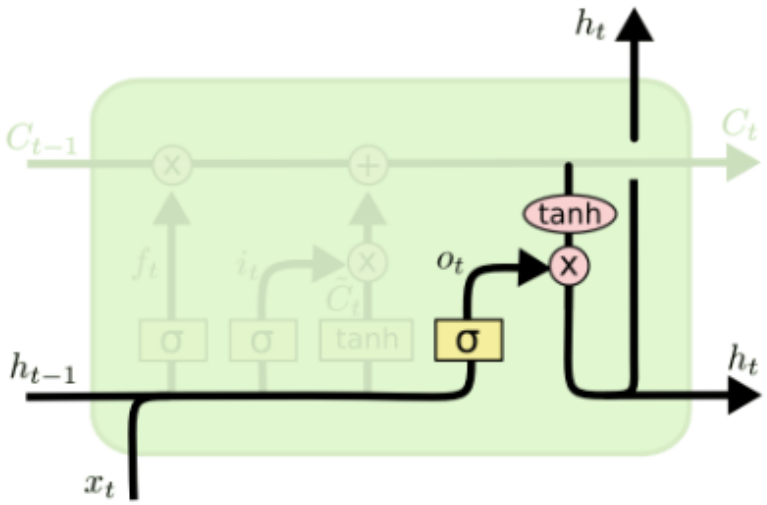


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figura IV.6: Tercer paso en la LSTM. Aplicación de las puertas 1, 2 y 3. Fuente: (Olah, 2015)

Cuarto paso

El cuarto y último paso consiste en decidir qué se va a generar como *output*. Este *output* o resultado estará basado en el *cell state* pero será una versión filtrada del mismo. En primer lugar se utiliza una capa sigmoidal la cual decide qué partes del *cell state* se sacarán como *output*. Posteriormente se hace pasar el *cell state* a través de una capa **tanh** para transformar los valores y ponerlos entre -1 y 1 y se multiplican por el *output* de la capa sigmoidal, de manera que sólo se sacan como *output* las partes que se deciden. De hecho, el proceso de multiplicar la cantidad de memoria presente en el flujo que es seleccionada por la capa **tanh** por la capa sigmoidal está definiendo el volúmen de información que sale de la LSTM como output hacia el siguiente módulo.



$$o_t = \sigma\ (W_o\ \cdot\ [h_{t-1}\ ,\ x_t] + b_o)$$

$$h_t = o_t * tanh(C_t)$$

Figura IV.7: Cuarto y último paso en la LSTM. Elección de la salida de la LSTM. Fuente: (Olah, 2015)

Finalmente cabe comentar que se construye una LSTM de tipo *STATEFULL*. Esto significa que todos los estados de la red son propagados al siguiente *batch*, de manera que no existe un proceso de barajado interno sino que el estado de la muesta X_i será utilizado en la computación de la muestra X_{i+bs} donde bs es el tamaño del *batch*.

IV.2 Métricas de rendimiento

Modelos de clasificación

Los modelos construidos están pensados para ser de ayuda a la hora de tomar una decisión de compra o venta de un *stock*. Si la predicción es +1 (Up) se espera que el precio al cabo de $d = 30,\ 60\ y\ 90$ días sea superior al actual y entonces la acción lógica del inversor sería **comprar**. Al contrario ocurriría si la predicción fuera -1, lo cual significaría que se espera que el precio al cabo de d días sea inferior. En este caso la decisión razonable a tomar por parte del inversor sería **vender**. Una predicción errónea puede causar grandes perdidas de dinero para el inversor. Por lo tanto, se deben definir métricas que ayuden a evaluar la potencia predictiva de los modelos construidos. Las métricas que se utilizan en el presente trabajo para evaluar el rendimiento de los modelos de clasificación que se crean a continuación son las siguientes:

Accuracy

Esta métrica hace referencia a la proporción de casos clasificados correctamente entre el total de casos con los que se prueba el modelo. Se calcula a partir de la fórmula siguiente:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \qquad (M.1)$$

Recall/Sensitivity

Esta métrica mide la habilidad de un modelo clasificador de identificar correctamente los casos positivos.

$$Sensitivity = \frac{tp}{tp + fn} \qquad (M.2)$$

Specificity

Esta métrica mide la habilidad de un modelo clasificador de identificar correctamente los casos negativos.

$$Specificity = \frac{tn}{tn + fp} \qquad (M.3)$$

Dónde,

tp = número de verdaderos positivos ≡ número de veces en las que el caso era positivo y el modelo predice positivo

tn = número de verdaderos negativos ≡ número de veces en las que el caso era negativo y el modelo predice negativo

fp = número de falsos positivos ≡ número de veces en las que el caso era negativo y el modelo predice positivo

fn = número de falsos negativos ≡ número de veces en las que el caso era positivo y el modelo predice negativo

Area Under the Curve (AUC)

Directamente derivado de la curva ROC se define este estadístico llamado Área bajo la curva. Como su nombre indica, se trata de calcular el área que queda por debajo de la curva ROC. El objetivo de esta métrica es el de calcular los distintos ratios de verdaderos positivos y de falsos positivos para distintos umbrales definidos, no sólo para el umbral de 0.5 de probabilidad. La idea es que de un clasificador aleatorio se puede obtener tantos verdaderos positivos como falsos positivos, que es lo que indica un AUC de 0.5 o la recta y=x en la curva ROC. Por lo tanto, el AUC indica la capacidad del modelo clasificador de obtener más verdaderos positivos que falsos positivos para todos los umbrales posibles. Como se ha indicado, un AUC de 1 indicaria un modelo clasificador perfecto, capaz de predecir con un 100% de acierto los verdaderos positivos y de no dar ningún falso positivo mientras que un AUC de 0.5 indicaria el mismo número de falsos positivos que de verdaderos positivos. El objetivo es, pues, el de obtener un modelo clasificador que tengo un AUC entre 0.5 y 1 cosa que significa que puede mejorar la predicción aleatoria. Otro de los usos comunes de esta métrica es la de poder comparar modelos clasificadores de manera que un mismo modelo será mejor que otro si su AUC es mayor.

La diferencia básica respecto a la *accuracy* definida en M.1 es que para calcular la *accuracy* el usuario tiene que definir un umbral a partir del cual se define la clasificación como "1" o "0" (caso positivo o negativo) mientras que el AUC está midiendo el rendimiento del modelo **mientras el umbral varía sobre todo los posibles valores**. En este sentido el AUC es una métrica más general que no depende (no es función de) el umbral que se defina.

Modelos de regresión

Para la evaluación de la LSTM sobre los precios se utilizan las siguientes métricas:

MAPE

Conocido por sus siglas en inglés, el Mean Absolute Percentage Error representa el error medio percentual que obtenemos al hacer una predicción. Se calcula de la siguiente manera:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Actual_i \ - \ Predicted_i|}{Actual_i} \qquad (M.4)$$

A su vez también se propone una medida alternativa de MAPE, que ofrece una visión más compensada de los errores:

$$MAPE_2 = \frac{\sum_{i=1}^{n}|Actual_i \ - \ Predicted_i|}{\sum_{i=1}^{n} Actual_i} \qquad (M.5)$$

Paralelamente a la medida de MAPE, cuyo rango de valores está restringido entre 0 y 1, se presentan las siguientes dos métricas:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Actual_i \ - \ Predicted_i| \qquad (M.6)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual_i \ - \ Predicted_i)^2} \qquad (M.7)$$

CAPÍTULO V

CASOS PRÁCTICOS: EJECUCIÓN

V.1 Dirección de movimiento del precio de cierre: análisis detallado con Random Forest

En el apartado siguiente se elabora la primera aplicación práctica de un sistema de inteligencia artificial sobre el sector de la inversión en activos financieros. En este apartado se procede a desarrollar un sistema clasificador que sea capaz de predecir la dirección del movimiento de un determinado *stock*. Concretamente, el objetivo de este tipo de aplicación es el de desarrollar un sistema que sea capaz de predecir si el precio de cierre de un activo será mayor o inferior al del momento actual al cabo de una determinada ventana de tiempo.

Se estructura la sección de la manera siguiente: en primer lugar se explica la procedencia y se describe la base de datos utilizada. En segundo lugar se explican los tratamientos de procesamiento de los datos que se aplican sobre los datos sucios extraídos previamente. A continuación se describen detalladamente los procesos de creación de variables que se elaboran para poder utilizarlas como *inputs* del modelo. En última instancia se propone una partición sobre los datos, previa a la realización de los experimentos y al análisis de los resultados.

V.1.1 ***Base de datos: Obtención y descripción***

Obtención

Los datos con los que se va a realizar la modelización descrita en este apartado se obtienen a través del paquete `quantmod`. Este paquete se diseñó para assistir a los *traders* quantitativos en el desarrollo, evaluación y puesta en funcionamiento de modelos de *trading* basados en la estadística. Concretamente, se utiliza la función `getSymbols` para obtener los datos ya que permite cargar y manejar `Symbols` en un ambiente especificado. Las posibles fuentes de datos son: Yahoo; Google, aunque actualmente no funciona; Alphavantage, con la ventaja de que podemos obtener datos intra-dia; MySQL, FRED, etc. Para obtener los datos se procede a utilizar *Yahoo Finance* como la fuente.

La metodología para la búsqueda de los distintos stocks con los cuales entrenar los modelos predictivos empieza con la idea de querer encontrar empresas representativas dentro de

distintos sectores estratégicos y con amplio impacto en la economía real. Además se pretende que los stocks utilizados tengan rentabilidad financiera con el objetivo de llevar el presente trabajo lo más cerca posible de una situación real de inversión. Para ello se utiliza el portfolio a fecha 01/01/2019 del gran *gurú* de las finanzas Warren Buffet (*Warren buffett*, n.d.). Dentro de las numerosas empresas presentes en este portfolio se escogen los 3 *stocks* del NYSE (bolsa de nueva york) y una del NASDAQ.

Simbolo	Nombre	Bolsa
AAPL	Apple Inc.	NASDAQ
WFC	Wells Fargo & CO	NYSE
KO	Coca-Cola Company	NYSE
AXP	American Express CO	NYSE

Tabla V.1: Stocks utilizados

Al hecho de que los *stocks* que forman la base de datos formen parte del portfolio de uno de los *gurús* del mundo de las finanzas y la inversión, se le suma también el hecho de que las empresas escogidas son empresas multinacionales con un gran capital que ocupan una posición destacada sus respectivos mercados, siendo representativas de cada uno de los sectores en los cuales operan. Seguidamente se describen brevemente las actividades de las cuatro compañías, siendo éstas el sector retail, sector tecnológico y sector de los servicios financieros o banca.

Coca - Cola Company

The Coca-Cola Company es una empresa multinacional de bebidas estadounidense. Tiene su sede en Atlanta, Georgia y es conocida por comercializar el refresco más consumido del mundo: la Coca-Cola. Esta bebida azucarada se creó en 1886 y forma parte de la historia moderna del mundo. Actualmente esta compañía está considerada una de las más grandes corporaciones de EEUU.

Apple INC.

Apple, Inc. es una empresa estadounidense del sector tecnológico que diseña y produce equipos electrónicos, software y hardware y ofrece servicios en línea. Es una compañía multinacional de gran renombre y trascendencia en los últimos años al ser los creadores del iPhone, el famoso teléfono inteligente, así como el ordenador personal Mac o el reproductor de música iPod. También ofrece el desarrollo de su propio sistema operativo que está presente en todos sus productos.

American Express

American Express Company (NYSE AXP) es una empresa dedicada a los servicios de finanzas, entre los cuales destaca la emisión de una tarjeta de crédito que lleva su mismo normbre y que es bastante popular en estados unidos. Esta compañía fue fundada en 1850.

Wells Fargo & Company

Wells Fargo & Co. es una compañía de servicios financieros que opera a nivel internacional. Se trata de uno de los bancos más importantes en los Estados Unidos, considerándose entre los 4 más potentes. Tiene su sede en California y es el resultado de una adquisición por parte de Norwest Corporation de Californa Wells Fargo & Co. en el año 1998.

Descripción

Después de utilizar la función `getSymbols` se obtiene una tabla para cada stock con un formato estandarizado. Se obtienen 5 series temporales con periodicidad diaria que hacen referencia a los precios de **apertura**, **cierre**, **máximo**, **mínimo** y **volúmen**. Además se incluye el precio ajustado pero esta variable no será utilizada en este trabajo. Todas las variables presentes en la base de datos utilizada son numéricas.

Los datos utilizados en este trabajo compreden el período 2000 - 2018, siendo el día `2000-01-01` la primera observación de cada serie temporal. La última observación de la base de datos es `2018-12-31`.

La función `str` permite visualizar fácilmente la estructura y formato que presentan en R las distintas tablas de la base de datos inicial. En el siguiente output se muestra como ejemplo la empresa Coca-Cola Company.

```
## 'data.frame':    4778 obs. of  6 variables:
##  $ KO.Open    : num  29 28.2 28.2 28.5 28.9 ...
##  $ KO.High    : num  29 28.4 28.7 28.8 30.4 ...
##  $ KO.Low     : num  27.6 27.8 28 28.3 28.9 ...
##  $ KO.Close   : num  28.2 28.2 28.5 28.5 30.4 ...
##  $ KO.Volume  : num  10997000 7308000 9457400 7129200 11474000 ...
##  $ KO.Adjusted: num  12.3 12.3 12.4 12.4 13.2 ...
```

Tablas descriptivas

Se explora descriptivamente los datos analizando los estadísticos descriptivos . Para cada empresa, se obtienen los distintos estadísticos de cada una de las variables descritas previamente. Para ello se elaboran 4 tablas que hacen referencia a los distintos estadísticos, calculados sobre los distintos precios y el volúmen, para una misma empresa.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Open	18.55	23.49	28.75	31.43	40.4	50.82
High	18.8	23.72	29.01	31.66	40.6	50.84
Low	18.5	23.26	28.47	31.19	40.15	50.25
Close	18.54	23.5	28.77	31.44	40.39	50.51
Volume	2.147M	9.821M	12.972M	14.692M	17.49M	124.169M

Tabla V.2: Estadísticos descriptivos para los distintos precios de Coca-Cola Company

Como se puede apreciar en la tabla V.2 el rango que han tomado los precios de cierre para la empresa Coca-Cola CO. en el período estudiado se mueve entre 18.54 y 50.51. El precio medio de cierre para el período estudiado es de 31.44 dólares.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Open	0.93	4.37	26.13	51.65	90.52	230.78
High	0.94	4.47	26.53	52.13	91.37	233.47
Low	0.91	4.21	25.66	51.12	89.72	229.78
Close	0.94	4.36	26.1	51.64	90.52	232.07
Volume	9.835M	52.133M	93.003M	119.542M	156.041M	1855.41M

Tabla V.3: Estadísticos descriptivos para los distintos precios de Apple Inc.

Como se puede apreciar en la tabla V.3 el rango que han tomado los precios de cierre para la empresa Apple Inc. en el período estudiado se mueve entre 0.94 y 232.07. El precio medio de cierre para el período estudiado es de 51.64 dólares, superior al de la empresa Coca-Cola CO.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Open	9.99	41.14	50.19	55.59	69.94	113.99
High	10.66	41.58	50.89	56.16	70.45	114.55
Low	9.71	40.5	49.72	55	69.56	112
Close	10.26	41.04	50.22	55.59	70.08	112.89
Volume	0.837M	3.85M	5.314M	6.979M	7.965M	90.337M

Tabla V.4: Estadísticos descriptivos para los distintos precios de American Express CO.

Como se puede apreciar en la tabla V.4 el rango que han tomado los precios de cierre para la American Express CO. en el período estudiado se mueve entre 10.26 y 112.89. El precio medio de cierre para el período estudiado es de 55.59 dólares, ligeramente superior al de la empresa Apple Inc..

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Open	8.65	25.95	31.19	35.19	45.82	65.89
High	8.94	26.29	31.48	35.56	46.23	66.31
Low	7.8	25.54	30.84	34.82	45.47	65.66
Close	8.12	25.91	31.2	35.2	45.97	65.93
Volume	1.774M	9.129M	15.235M	23.412M	27.088M	478.737M

Tabla V.5: Estadísticos descriptivos para los distintos precios de Wells Fargo and CO.

Como se puede apreciar en la tabla V.5 el rango que han tomado los precios de cierre para la empresa Wells Fargo and CO. en el período estudiado se mueve entre 8.12 y 65.93. El precio medio de cierre para el período estudiado es de 35.2 dólares, ligeramente superior al de Coca-Cola CO. inferior al de `AAPL` y `AXP`.

Visualización gráfica

Seguidamente se explora gráficamente el precio de cierre para cada una de las empresas, al ser éste la variable a partir de la cual se calculará la variable respuesta sobre la que hacer predicción. El objetivo de este tipo de análisis es poder visualizar el tipo de evolución que han presentado los distintos *stocks* durante el período de estudio.

En el siguiente gráfico se representa el precio de cierre de Coca-Cola Company para el período comprendido entre el 03/01/2000 y 28/12/2018.

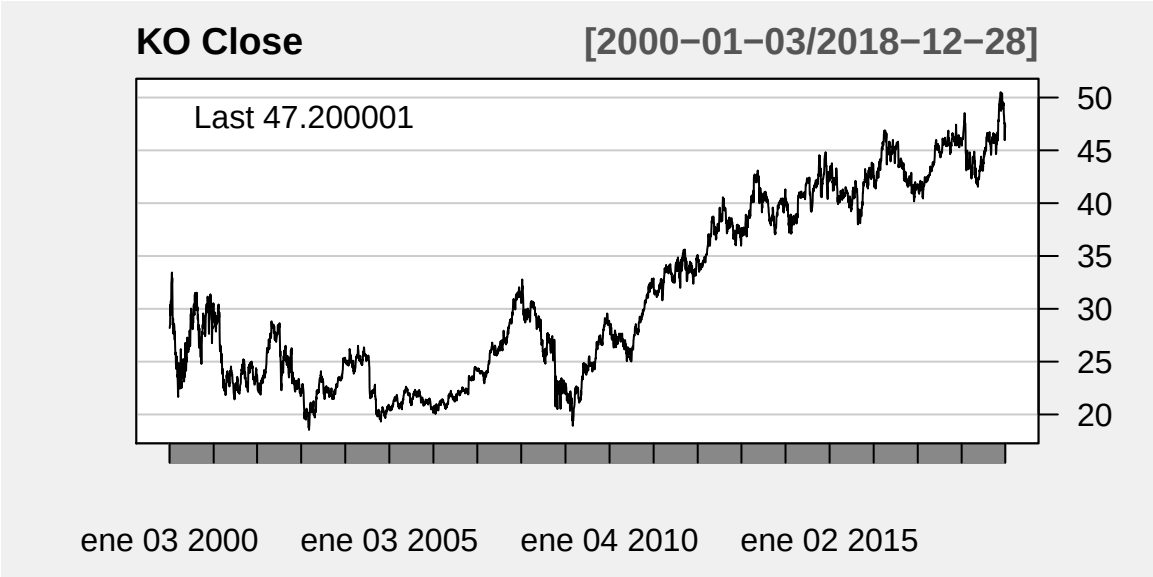


Figura V.1: Precio de cierre de Coca-Cola Company 03/01/2000 - 28/12/2018

En el siguiente gráfico se representa el precio de cierre de Apple Inc. para el período comprendido entre el 03/01/2000 y 28/12/2018.

Figura V.2: Precio de cierre de Apple Inc. 03/01/2000 - 28/12/2018

En el siguiente gráfico se representa el precio de cierre de Wells Fargo and CO. para el período comprendido entre el 03/01/2000 y 28/12/2018.

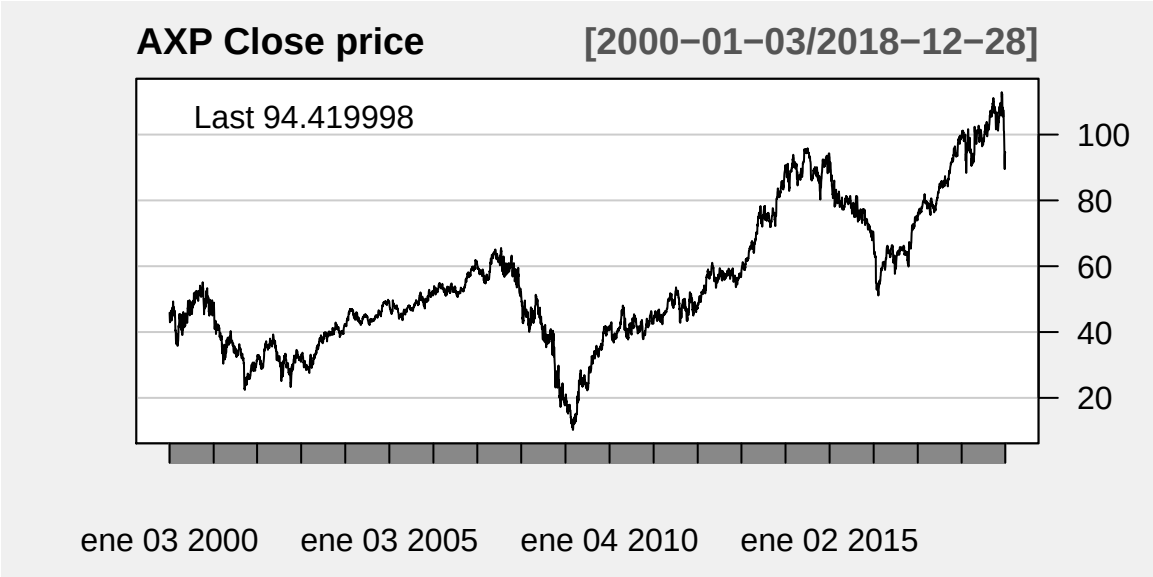


Figura V.3: Precio de cierre de American Express CO. 03/01/2000 - 28/12/2018

En el siguiente gráfico se representa el precio de cierre de Wells Fargo and CO. para el período comprendido entre el 03/01/2000 y 28/12/2018.

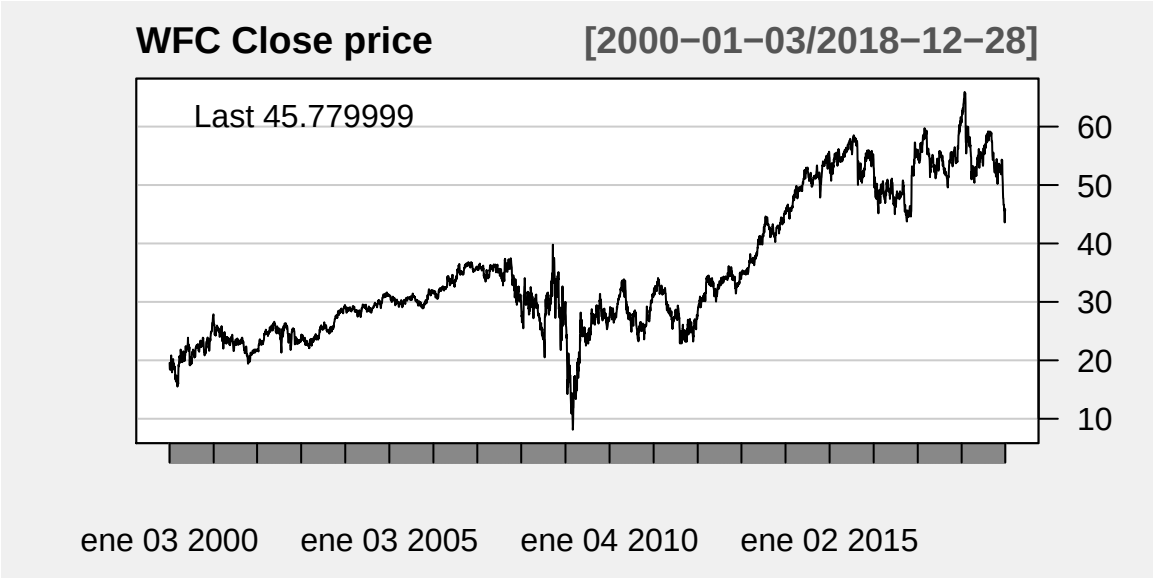


Figura V.4: Precio de cierre de Wells Fargo and CO. 03/01/2000 - 28/12/2018

A partir de esta exploración gráfica se descubre que 3 de las 4 empresas (KO, AXP y WFC) presentan un comportamiento relativamente similar. Como se puede observar en la figura V.1, V.3 y V.4 la evolución de `KO, AXP y WFC` en el período estudiado presenta una tendencia, en general, creciente. Si se analiza en detalle el lector puede detectar 3 etapas claramente

destacada en la evolución de estos stocks, siendo la primera de ellas distinta en el caso de Wells Fargo & CO. Estos 3 períodos claramente diferenciados son los correspondientes a los años 2000-2007, 2008-2010 y 2011-2018.

2000-2006

En el caso de las empresas Coca-Cola CO. y American Express CO. la primera etapa (2000 - inicios de 2006) corresponde a la relajación de los mercados financieros posterior al *boom de las .com*. La segunda mitad de los 90 fueron una época de expansión económica en la cual se produjo un boom financiero en USA (ver Urban Jermann, 2003) que se relajó a partir de finales de milenio.

2006-2008

Esta relajación fué seguida por la segunda etapa, un período de crecimiento económico y de los mercados financieros que se produjo a partir del 2006 y hasta el 2008. Esta etapa de crecimiento fué previa a la gran recesión mundial que tuvo lugar a partir del año 2008 (ver (Andrew K. Rose, 2011), (DeLong, 2009), (P.R. Lane, 2011) pp. 77-110 para más información].

2008-2010

Posterior al crecimiento experimentado a nivel global antes del año 2008, se produjo la última gran crisis financiera que ha tenido lugar a nivel global. Es bien sabido que el problema financiero que se expandió a Europa empezó en USA, a causa de los paquetes de activos financieros de alto riesgo mezclados con otro tipo de activos financieros. Esta crisis se ve representada en los gráficos gracias al socavón que presentan los precios de cierre de las empresas consideradas como ejemplo entre el 2008 y el 2010. Cabe destacar que ,como se puede apreciar a partir de las gráficas anteriores, la empresa Apple Inc. es la única empresa de las 4 consideradas como ejemplo que no fue fuertemente afectada durante este periodo de crisis, experimentando sus precios un ligero decremento durante este periodo (ligero en comparación con las otras empresas).

2010 - Actualidad

La etapa posterior a la gran crisis financiera mundial del 2008-2010 ha sido una etapa de crecimiento, tal como demuestra el hecho de que las 4 empresas consideradas presentan un crecimiento de los precios de cierre. Especialmente en Europa, a partir de las políticas de austeridad creadas después de la crisis, lideradas por Alemania, permitieron una situación

económica favorable para el crecimiento, al mantener el BCE el precio del dinero (el tipo de interés) prácticamente al nivel de 0.

Descriptiva financiera

Seguidamente se muestran las distintas tablas comparativas de los distintos precios obtenidos inicialmente. Éstas incluyen la media, la desviación típica o volatilidad del precio y el coeficiente de variación, el cual es un estadístico utilizado de manera frecuente para medir la volatilidad de un stock al ser una variable que no tiene en cuenta las unidades de medida. Esto significa que permite comparar distintos stocks en cuanto a volatilidad se refiere sin tener en cuenta la magnitud que éstos presenten.

Este tipo de visualización permite analizar más detenidamente cada uno de los precios y poder compararlos fácilmente entre todas las empresas. Es además una buena manera de analizar a priori la predictibilidad de los distintos precios al poder comparar los distintos coeficientes de variación. La idea de evaluar la predictibilidad de una serie temporal con el coeficiente de variación sugiere que es más fácil de predecir una serie temporal con un comportamiento más estable, esto es, con menos unidades de desviación típica por unidad de media (Gilliland, 2009).

Nombre	Media	Desv_std	CoV
Apple Inc.	51.64595	55.925416	1.0828616
Wells Fargo & CO	35.19440	11.806917	0.3354771
Coca-Cola Company	31.42690	8.652774	0.2753302
American Express CO	55.58608	21.112421	0.3798149

Tabla V.6: Estadísticos descriptivos para el precio de apertura (Open).

Nombre	Media	Desv_std	CoV
Apple Inc.	52.13167	56.386960	1.0816258
Wells Fargo & CO	35.56184	11.819827	0.3323739
Coca-Cola Company	31.66443	8.670834	0.2738351
American Express CO	56.16119	21.131090	0.3762579

Tabla V.7: Estadísticos descriptivos para el precio máximo (High).

Nombre	Media	Desv_std	CoV
Apple Inc.	51.12417	55.450009	1.0846143
Wells Fargo & CO	34.82003	11.805231	0.3390357
Coca-Cola Company	31.18907	8.644319	0.2771585
American Express CO	55.00056	21.083165	0.3833264

Tabla V.8: Estadísticos descriptivos para el precio mínimo (Low).

Nombre	Media	Desv_std	CoV
Apple Inc.	51.63773	55.925371	1.0830331
Wells Fargo & CO	35.19502	11.807079	0.3354758
Coca-Cola Company	31.43855	8.656529	0.2753476
American Express CO	55.58724	21.099819	0.3795803

Tabla V.9: Estadísticos descriptivos para el precio de cierre (Close).

Como se puede apreciar en la tabla V.9, las compañías American Express y Apple Inc. son las que presentan un precio de cierro medio mayor, superando en ambos casos los 50 dólares/acción. A su vez son las 2 compañías que presentan una desviación estándard del precio de cierre mayor, siendo la de Apple la mayor de todas con 55.93. Esto se debe al hecho de que Apple presenta una gran tendencia creciente en los precios de cierre desde el año 2000 y el cálculo de la desviación estándard no es capaz de eliminar este efecto de la tendencia. Este hecho también se aprecia en el cálculo del coeficiente de variación, que en el caso de Apple es superior a 1. Por otro lado, la empresa que presenta unos resultados menores es Coca-Cola CO., con un menor precio de cierre medio (31.44dólares/acción) y menores desviación típica y coeficiente de variación.

Nombre	Media	Varianza
Apple Inc.	0.077	7.203
Wells Fargo & CO	0.018	5.699
Coca-Cola Company	0.011	1.679
American Express CO	0.015	5.004

Tabla V.10: Momentos calculados sobre los logaritmos de la rentabilidad (*log returns*)

En la tabla anterior se muestra la media y la varianza de las rentabilidades de las distintas empresas durante el periodo estudiado. La empresa que ofrece una mayor rentabilidad media es Apple, pero a su vez presenta una mayor varianza. En este sentido las empresas Wells Fargo y American Express no parecen tan recomendables al presentar rentabilidades medias de menos del 0.02% y varianzas de más de 5 puntos. Esto significa que estos stocks aportan, de media, unas rentabilidades relativamente bajas en comparación con el riesgo que habría que asumir. En este caso una opción más balanceada en términos de rentabilidad-riesgo es la de Coca-Cola ya que tiene una rentabilidad media de 0.01% con un riesgo de 1.7.

V.1.2 ***Procesamiento de los datos***

Siguiendo la metodología propuesta en la figura 2.1, en primer lugar se procede a elaborar una transformación de todas las variables/precios iniciales. Esta transformación suaviza los datos con el objetivo de remover la variación aleatória y/o el ruido, haciendo que los modelos predictivos de dirección de movimiento elaborados posteriormente sean capaces de detectar más fácilmente una tendencia a largo plazo de los precios dentro del comportamiento de los mismos (Luckyson Khaidem, 2016, p. 4). El objetivo es, por un lado, crear la variable

respuesta utilizando el precio de cierre alisado y, por otro, calcular los indicadores técnicos detallados en la sección 5.3 para, posteriormente, utilizarlos como variables predictoras.

Por un lado se utilizan las medias móviles exponenciales a 30, 60 y 90 días como nuevas representaciones de los precios. De esta manera se obtiene una nueva representación de los precios en la cual se ha removido la variación aleatória y el ruido. Seguidamente se muestran las gráficas para las medias móviles exponenciales del precio de cierre para la empresa Coca-Cola CO. con el objetivo de visualizar el alisado que se obtiene al aplicar este cálculo. A medida que se añaden más datos en el cálculo de las medias se obtiene un mayor alisado.

Por otro lado se aplica el mismo tipo de alisado exponencial aplicado por (Luckyson Khaidem, 2016, p. 4). El objetivo es el mismo que en el caso del cálculo de las medias móviles. El estadístico alisado exponencialmente de un serie temporal Y se puede calcular recursivamente, desde el momento en el que se disponga de dos observaciones, de la manera siguiente:

$$S_0 = Y_o$$
$$S_t = \alpha * Y_y + (1 - \alpha) * S_{t-1} \quad \forall \ \ t > 0 \qquad (1)$$

donde $0 < \alpha < 1$ es el factor de alisado. El método del alisado exponencial otorga más peso a las observaciones más recientes en tanto que hace decrecer exponencialmente el peso de las observaciones más antiguas. (cita articulo, formula internet). Valores elevados de α provocan que el nivel de alisado quede reducido, de manera que en el caso extremo en el que $\alpha = 1$ el estadístico exponencialmente alisado es igual que la observación. Es decir $S_t = Y_t \quad si\ \alpha = 1$. En el presente trabajo se aplica la fórmula anterior sobre cada uno de los precios y el volúmen con un factor de alisado $\alpha = 0.05$ para conseguir un alisado más pronunciado que en el caso de las medias móviles exponenciales.

A su vez se presentan la siguiente tabla para poder visualizar la disminución de variabilidad que se obtiene tras aplicar las distintas medias móviles exponenciales al precio de cierre de las distintas empresas. La volatilidad se evalúa en este caso con el coeficiente de variación para permitir la comparabilidad:

Nombre	CoV_EMA30	CoV_EMA60	CoV_EMA90	CoV.exp.smooth.alpha.0.05
Apple Inc.	1.0799103	1.0746436	1.0686425	1.0852714
Wells Fargo & CO	0.3322648	0.3290268	0.3263450	0.3336280
Coca-Cola Company	0.2735308	0.2716672	0.2700638	0.2722442
American Express CO	0.3763897	0.3725283	0.3686093	0.3746762

Tabla V.11: Comparativa coeficiente de variación entre las distintas medias móviles exponenciales

Una vez alisados los precios de los distintos *stocks* se procede a calcular tanto la variable respuesta como las variables predictoras, en este caso los indicadores técnicos, a partir de éstos.

V.1.3 ***Creación de variables***

Variable respuesta

Después de aplicar ambos tipos de alisado sobre los datos se obtienen 4 bases de datos para cada una de las empresas, las cuales hacen referencia a los precios transformados con las medias móviles exponenciales de 30, 60 y 90 días y utilizando la formula del alisado exponencial. En total la base de datos contiene ahora 16 tablas, 4 para cada empresa.

En este momento se define la variabe respuesta a partir de las distintas transformaciones del precio de cierre de siguiendo la fórmula siguiente:

$$Target_t = \begin{cases} Up & Close_{t+d} > Close_t \\ Down & Close_{t+d} < Close_t \end{cases} \tag{2}$$

donde:

- Close_t es el precio de cierre del activo en el día "t".

La variable respuesta en t toma el valor Up si el precio de cierre a día $t + d$ es superior al precio de cierre del día actual t. Del mismo modo, la variable respuesta en t toma el valor $Down$ si el precio de cierre en $t + d$ es inferior al precio de cierre del día actual t.

En la mayoría de la literatura revisada se construye la variable respuesta para la observación correspondiente al día t comparando el precio de cierre de se día con el precio de cierre

de un día anterior $t - d$ (véase (Yakup Kara, 2011) p. 5313, (Masoud, 2014) p. 610). Económicamente esta variable está midiendo si el precio actual ha tenido una evolución positiva, o no, respecto al valor que presentaba, por ejemplo, hace un mes. Posteriormente se calculan las variables predictoras, esto es, los indicadores técnicos, a partir de los precios OHLC medidos el día t. En este caso los modelos de aprendizaje automático obtienen, en general, buenos rendimientos y son capaces de clasificar una nueva observación. Sin embargo este modo de calcular la variable respuesta no brinda al inversor una herramienta útil para argumentar sus decisiones de compra o venta ya que para predecir la variable respuesta en t, que está calculada a partir del precio de cierre, se utilizan indicadores técnicos calculados, a su vez, con el precio de cierre. Es decir en el momento en el que se pueden obtener los directamente con el precio de cierre anterior para ver si el precio ha subido o no. Es por eso que en esta tesis se ha decidido construir la variable respuesta de una manera distinta, al ser el objetivo del mismo el de construir un modelo predictivo que sea útil a la hora de tomar una decisión de inversión. Por eso se ha decidido construir la variable respuesta como la construida en ((Luckyson Khaidem, 2016), p. 4, (Han, 2012) p. 16 y (Kim, 2003) p. 4). Creando la variable a predecir de este modo, el inversor puede utilizar los precios en t para predecir la dirección que tomará el precio de ese *stock* al cabo de 1, 2 y/o 3 meses.

Se ha decidido definir 3 valores para d a la hora de calcular la variable respuesta. Así se obtienen 3 tipos de variable respuesta a predecir para cada tipo de alisado aplicado a los precios de cada empresa. Estos 3 tipos hacen referencia a los distintos momentos en el futuro para los que se hace la predicción. En este caso, pues, se van a realizar predicciones sobre la dirección que tomará el precio a 1, 2 y 3 meses vista. De este modo se consigue una tabla 4x3 de combinaciones posibles para la definición del modelo entre el periodo de alisado exponencial y el número de días en el futuro sobre el que se quiere predecir. Teniendo en cuenta que en la base de datos sólo se tienen datos de los días laborables, se ha decidido establecer el valor de d acorde con la tabla siguiente para aproximar la predicción extactamente a 1, 2 y 3 meses:

Prediccion	d
1 mes	20
2 meses	40
3 meses	60

Tabla V.12: Periodos de predicción

Las tablas siguientes muestran la proporción de observaciones que representan cada una de las dos clases de la variable respuesta en las distintas empresas para los distintos tipos de alisado aplicados:

Coca Cola CO.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	r1.Down	r1.Up	r2.Down	r2.Up	r3.Down	r3.Up
EMA30	45.44301	54.55699	43.25759	56.74241	41.37343	58.62657
EMA60	43.64758	56.35242	40.69246	59.30754	38.91393	61.08607
EMA90	40.71536	59.28464	38.97612	61.02388	39.25254	60.74746
Alisado exponencial	45.06095	54.93905	42.76066	57.23934	40.46206	59.53794

Tabla V.13: Proporción de la variable respuesta Coca-Cola CO.

Como se puede observar en la tabla anterior la variable respuesta está relativamente bien balanceada en la mayoría de los casos. Cuanto más lisos són los datos, es decir, mayor el periodo de cálculo de las medias móviles exponenciales, mayor es la proporción de días en los que el precio al cabo de 1, 2 y 3 meses ha sido superior (Up). A su vez, para un mismo tipo de alisado sobre los datos, se observa que cuanto más lejos se predice la dirección del precio de cierre, mayor es el porcentage de días en los que el precio de cierre ha sido más elevado. En este caso, esto significa que si el *stock* presenta una tendencia creciente, como es el caso para esta empresa, el hecho de predecir su dirección en un futuro más lejano hace que se la variable respuesta se desbalancee a favor de los días con dirección Up.

Apple Inc.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	r1.Down	r1.Up	r2.Down	r2.Up	r3.Down	r3.Up
EMA30	34.08754	65.91246	31.83266	68.16734	30.30497	69.69503
EMA60	31.92169	68.07831	30.26288	69.73712	29.62009	70.37991
EMA90	28.82844	71.17156	28.26414	71.73586	27.45733	72.54267
Alisado exponencial	33.18621	66.81379	31.21570	68.78430	29.71598	70.28402

Tabla V.14: Proporción de la variable respuesta Apple Inc.

En el caso de la empresa Apple Inc. la variable respuesta no está balanceada en ninguno de los casos. En la tabla 5.13 se puede observar que cuanto menor sea el nivel de alisado en los datos, correspondiente a la EMA de 30 días, más cerca estará la variable respuesta de estar balanceada (34% Down - 66% Up). Esto significa que cuanto más alisados sean los datos y más elevada sea d en la fórmula 2 más desbalanceada estará la variable respuesta para esta empresa, alcanzado un máximo de 27.3% Down - 72.7% Up. Esta proporción tan desbalanceada se explica por la fuerte tendencia creciente de los precios de cierre de AAPL dentro del período de estudio.

American Express CO.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	r1.Down	r1.Up	r2.Down	r2.Up	r3.Down	r3.Up
EMA30	41.48869	58.51131	42.25950	57.74050	38.68629	61.31371
EMA60	39.88083	60.11917	38.85446	61.14554	36.35973	63.64027
EMA90	36.45320	63.54680	36.45945	63.54055	34.13264	65.86736
Alisado exponencial	41.27785	58.72215	41.40988	58.59012	38.53328	61.46672

Tabla V.15: Proporción de la variable respuesta American Express CO.

En el caso de la empresa American Express CO. la variable respuesta está en general ligeramente desbalanceada y presenta el mismo patrón de crecimiento del desbalanceo que las empresas Coca Cola CO. y Apple Inc, alcanzando el máximo balanceo entre las clases de la variable respuesta con la combinación de alisado con EMA 30 días y predicción a un mes vista (41% Down - 59% Up).

Wells Fargo & CO.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	r1.Down	r1.Up	r2.Down	r2.Up	r3.Down	r3.Up
EMA30	40.85430	59.14570	40.34827	59.65173	41.82128	58.17872
EMA60	39.66801	60.33199	39.53836	60.46164	39.83687	60.16313
EMA90	38.63782	61.36218	38.00817	61.99183	37.15705	62.84295
Alisado exponencial	40.87852	59.12148	40.12241	59.87759	41.28868	58.71132

Tabla V.16: Proporción de la variable respuesta Wells Fargo and CO.

En el caso de la empresa Wells Fargo & CO. la variable respuesta está, en general, ligeramente desbalanceada y presenta el mismo patrón de crecimiento del desbalanceo que las empresas Coca Cola CO., Apple Inc y American Express CO.. La variable respuesta calculada para este *stock* presenta un máximo balanceo con la combinación de alisado con EMA 30 días y predicción a un mes vista (40.8% Down - 59.2% Up). El caso de esta empresa es distinto al de las anteriores en el hecho de que el balanceo entre las clases de la variable respuesta aumenta conforme se predice más en futuro para el caso de alisado exponencial de los datos. Cuando se analiza este tipo de alisado, la variable respuesta está más balanceada cuando se predice a 3 meses vista que cuando se predice a 1 mes vista.

Variables predictoras / regresores

En esta tesis se utilizan distintos indicadores técnicos como variables predictoras de los modelos que se construyen posteriormente. Los indicadores técnicos son estadísticos que se calculan a partir de los precios OHLC que presenta un *stock*. Muchos managers de inversión e

inversores en el mercado de *stocks* aceptan y utilizan generalmente ciertos indicadores técnicos como señales de las tendencias futuras del mercado (véase Kim, 2003). Algunos indicadores técnicos son efectivos en mercados con tendencia y otros rinden mejor en mercados no cíclicos y sin tendencia (véase Ray Tsaih, 1998). En mucha de la literatura revisada en torno a la previsión del precio de los *stocks* con indicadores técnicos queda probada la utilidad de éstos como variables predictoras en modelos de predicción tanto del precio como de la dirección de movimiento de los mismos (An-Sing Chena, 2003). En este trabajo se han seleccionado distintos indicadores técnicos para utilizarlos como variables predictoras a partir de la revisión de artículos de expertos en la materia como ((Kim, 2003) (Han, 2012) (Yakup Kara, 2011) (C.L. Huang, 2009) (Manish Kumar, 2006) (Y. Nakamori, 2005)). Seguidamente se detallan los indicadores técnicos utilizados y las fórmulas necesarias para calcularlos.

Para facilitar el cálculo de los siguientes indicadores se ha creado una función de `R` llamada `feature_extraction_finance`. Esta función recibe un `data.frame` con los precios OHLC y el volúmen de un *stock* y devuelve otro `data.frame` en el cual cada observación representa una fecha y cada columna una variable, siendo la primera de ellas la variable respuesta.

Aroon

Aroon es un indicador que puede descubrir el inicio de tendencias. Consiste en tres índices entre los cuales se escogen los dos primeros para utilizarlos como variables predictoras. Éstos están definidos en las fórmulas TI.1 y TI.2.

$$AroonUp = 100 * (\frac{n - PeriodSinceHighestHigh}{n}) \qquad (IT.1)$$

$$AroonDown = 100 * (\frac{n - PeriodSinceLowestLow}{n}) \qquad (IT.2)$$

Media móbil simple 10 días

La media móbil simple, SMA por sus siglas en inglés, calcula la media de los precios en un período de tiempo.

$$SMA_{10} = \frac{Close_t + Close_{t-1} + ... + Close_{t-10}}{10} \qquad (IT.3)$$

Momentum

Este indicador técnico mide la cantidad que el precio de un *stock* ha variado respecto d días en el pasado. Se procede a utilizar como variables predictoras los indicadores Momentum con $d = 1, 2, 3, 4, 5, 10\ y\ 15$ calculados a partir de la fórmula (IT.4).

$$Mom_d = Close_t - Close_{t-d} \qquad (IT.4)$$

Rate of Change

El ratio de cambio, ROC por sus siglas en inglés, calcula la variación de un precio respecto d días en el pasado expresado en porcentage (véase ROC en Ulrich, 2018). En este trabajo se calcula el ratio de cambio respecto $d = 1\ y\ 9$ días en el pasado utilizando la fórmula (IT.5).

$$ROC_d = (\frac{Close_t - Close_{t-d}}{Close_{t-d}}) * 100 \qquad (IT.5)$$

Stochastic oscillator: fast %K, fast %D & slow %D

El indicador técnico llamado Stochastic oscillator relaciona la localización del precio de cierre de cada día en relación con el rango de valores que ha tomado en los últimos d días. En esta disertación se utilizan como variables predictoras los estadísticos fast %K, fast %D y slow %D. El hecho de que el indicador sea llamado *fast* hace referencia a que los precios con los que se han calculado esos indicadores no han sido alisados. De manera contraria, el hecho de que el indicador sea llamado *slow* se refiere a que se ha calculado a partir de precios alisados. En este caso, al calcular los distintos indicadores a partir de precios ya

alisados, el hecho de que un indicador sea *slow* significa que se ha calculado a partir de precios doblemente alisados. Los períodos utilizados para el cálculo de estos indicadores técnicos son $dFastK = 14, nFastD = 3, nSlowD = 3$, correspondientes a los valores por defecto de la función `stoch` del paquete `TTR`. Para el cálculo de estos indicadores se han utilizado las fórmulas siguientes:

$$fast\ K = (\frac{Close_t \quad - LowestLow_{t-d}}{HighestHigh_{t-d} - LowestLow_{t-d}}) * 100 \qquad (IT.6)$$

Dónde $LowestLow_{t-d}$ y $HighestHigh_{t-d}$ hacen referencia al precio de cierre mínimo más pequeño y al máximo más grande de los últimos $t - d$ días.

$$fast\ D = MovingAverage(fastK) = \frac{\sum_{i=0}^{n-1} fastK_{t-i}}{n} \qquad (IT.7)$$

$$slow\ D = MovingAverage(slowK) = \frac{\sum_{i=0}^{n-1} slowK_{t-i}}{n} \qquad (IT.8)$$

Stochastic Momentum Index

Este indicador técnico calcula el valor relativo del precio de cierre respecto el punto medio del rango de precios de los pasados d. Es parecido al oscilador estocástico pero con respecto al punto medio del rango de precios y no respecto al rango total. En otras palabras mide cuán cerca está el precio de cierre respecto el centro del rango de precios de los pasados d días. El cálculo de este indicador se desarrolla en las fórmulas siguientes:

$$cm = Close_t - \frac{(HighestHigh - LowestLow)}{2}$$
$$hl = HighestHigh - LowestLow$$
$$cmMA = EMA(EMA(cm))$$
$$hlMA = EMA(EMA(hl))$$

$$SMI = 100 * \frac{cmMA}{\frac{hlMA}{2}} \qquad (IT.9)$$

En la presente disertación se utilizado como característica de los datos el estadístico calculado con la fórmula (IT.9). Los períodos utilizados para el cálculo de las medias móbiles exponenciales son los valores por defecto (véase SMI en Ulrich, 2018).

Relative Strenght Index

El índice de fuerza relativa, llamado RSI por sus siglas en inglés, es un indicador que pretende medir la fuerza del *stock* en el movimiento que presenta actualmente. Calcula la ratio de los recientes precios crecientes respecto el movimiento absoluto de los precios (véase RSI en Ulrich, 2018). Se ha calculado a partir de la fórmula siguiente:

$$RSI = 100 - \frac{100}{1 + RS} \qquad (IT.10)$$

$$RS = \frac{ganancias\ medias}{prdidas\ medias} = \frac{\frac{\sum_{i=0}^{n-1} Up_{t-i}}{n}}{\frac{\sum_{i=0}^{n-1} Dw_{t-i}}{n}}$$

Dónde Up_t representa el cambio de los precios al alza y Dw_t el cambio de los precios a la baja en el momento t

Indicador Williams Accumulation/Distribution

Este indicador técnico pretende identificar la tendencia del mercado y medir la presión del mismo. Es llamado Williams AD por sus siglas en inglés y se calcula utilizando las fórmulas (IT.11), (IT.12) y (IT.13) (véase williamsAD en Ulrich, 2018):

Si $Close_t > Close_{t-1}$ *entonces*

$$AD_t = AD_{t-1} + Close_t - min(Low_t, Close_{t-1}) \qquad (IT.11)$$

Si $C_t < C_{t-1}$ *entonces*

$$AD_t = AD_{t-1} + max(High_t, Close_{t-1} - Close_t) \qquad (IT.12)$$

Si $C_t = C_{t-1}$ *entonces*

$$AD_t = AD_{t-1} \qquad (IT.13)$$

Indicador Williams Percentage Range

Este indicador técnico se calcula de un modo parecido al indicador estocástico fast %K. Es un indicador *momentum* que mide niveles de sobreventa y sobrecompre. Se ha calculado a partir de la fórmula (IT.14).

$$WPR = \frac{HighestHigh_n - Close_t}{HighestHigh_n - LowestLow_n} * 100 \qquad (IT.14)$$

Indicador Moving Average convergence divergence

Este indicador es más conocido por sus siglas en inglés: MACD. Este indicador es un oscilador que calcula la diferencia entre dos medial móbiles exponenciales, una "rápida" (n=12) y un "lenta" (n=26), las cuales pueden reflejar tendencias en el movimiento de los precios (véase MACD en Ulrich, 2018). Ha sido calculado a partir de la fórmula (IT.15)

MACD = EMA(stockPrices,12) − EMA(stockPrices,26) (IT.15)

Indicador Commodity Channel Index

Este indicador es comúnmente conocido como CCI por sus siglas en inglés. Se utiliza para descubrir los principios y finales de tendencias en *securities* (véase Lambert, 1980). Se calcula a partir de las fórmulas siguientes. El valor utilizado como variable predictora en este trabajo es el calculado con la fórmula (IT.16)

$$TP_t = \frac{High_t + Low_t + Close_t}{3}$$
$$MATP_t = MovingAverage(TP, n)$$

$$MDTP = \frac{\sum_{i=1}^{n} |TP_{t-i+1} - MATP_t|}{n}$$

$$CCI = \frac{TP_t - MATP_t}{0.015 MDTP_t} \qquad (IT.16)$$

Indicador On Balance Volume

Este indicador se llama OBV por sus siglas en inglés y mide el flujo de dinero dentro y fuera de un *stock*. Es una medida del flujo monetario que presenta un *stock*. Su cálculo viene dado por las fórmulas (IT.17), (IT.18) y (IT.19):

Si $Close_t > Close_{t-1}$ *entonces*

$$OBV_t = OBV_{t-1} + Volume_t \qquad (IT.17)$$

De otro modo si $Close_t < Close_{t-1}$ *entonces*

$$OBV_t = OBV_{t-1} - Volume_t \qquad (IT.18)$$

De otro modo

$$OBV_t = OBV_{t-1} \qquad (IT.19)$$

Indicador Average True Range

El indicador rango medio verdadero, ATR por sus siglas en inglés, es un grupo de estadísticos que estima la volatilidad de una serie temporal (Wilder, 1978). Su cálculo viene dado por las fórmulas siguientes:

$$TrueHigh = max(High_t, Close_{t-1})$$
$$TrueLow = min(Low_t, Close_{t-1})$$

TR_t= TrueHigh_t − TrueLow_t (IT.20)

$$ATR = \frac{TR_{t-1} * (n\ menos\ 1) + TR_t}{n} \qquad (IT.21)$$

En el presente trabajo se utilizan como variables predictoras las fórmulas del verdadero rango y el verdadero rango medio definidas en las ecuaciones (IT.20) y (IT.21). Además se utiliza también una variante del verdadero rango calculada en la fórmula (IT.22)

TR2_t = (Close_t − TrueLow_t)/(TrueHigh_t − TrueLow_t) (IT.22)

Indicador Trend Detection Index

Este índice de detección de tendencias se llama TDI por sus siglas en inglés. Se utiliza para descubrir el inicio y el final de tendencias móbiles (véase TDI en Ulrich, 2018). Su cálculo viene dado por las ecuaciones siguientes:

Mom = precio − precio_{periodo}

$$MomAbs = |Mom|$$
$$DI = \sum_{i=1}^{n} Mom_i \qquad (IT.23)$$
$$DIAbs = |DI|$$
$$DIAbsSum = \sum_{i=1}^{n} DIAbs$$
$$DIAbsSum2 = \sum_{i=1}^{2n} DIAbs$$

TDI = DIAbs−(DIAbsSum2−DIAbsSum) (IT.24)

Se utilizan como variables predictoras los indicadores DI y TDI definidos en las ecuaciones (IT.23) y (IT.24)

Indicador Welles Wilder's Directional Movement Index

Este indicador se conoce generalmente como ADX. ADX consiste en 4 indicadores llamados Índice Direccional Poisitvo (DIp), Índice Direccional Negativo (DIn), Índice Direccional (DI) y el Índice direccional Medio (ADX) (véase Wilder, 1978).

HiDiff = High_{t-1} − High_t

LoDiff = Low_{t} − Low_{t-1}

$Si\ HiDiff < 0\ y\ LoDiff < 0\ \ o\ \ HiDiff = LoDiff\ entonces$

$$DIp = 0$$

$$DIn = 0$$

$Si\ HiDiff > LoDiff\ \ entonces$

$$DIp = HiDiff$$

$$DIn = 0$$

$Si\ HiDiff < LoDiff\ \ entonces$

$$DIp = 0$$

$$DIn = LoDiff$$

$$DX = \frac{DIp\ DIn}{DIp + DIn} \qquad (IT.25)$$

$$ADX = \frac{ADX_{t-1} * (n\ menos\ 1) + DX_t}{n} \qquad (IT.26)$$

Se procede a utilizar las ecuaciones (IT.25) y (IT.26) como variables predictoras en el modelo. Por añadidura también se utiliza como variable predictora la ratio dada en la ecuación (IT.27)

$$PN_{ratio} = \frac{DI_p}{DI_n} \qquad (IT.27)$$

Indicador Bollinger Bands

Este indicador llamado Bandas de Bollinger, más conocido por sus siglas en inglés BB, es utilizado para medir la volatilidad de un *stock* y compararla con el nivel para un período de tiempo (véase BBands en Ulrich, 2018). Se ha calculado a partir de la fórmula siguiente (IT.28).

$$TP_t = \frac{High_t + Low_t + Close_t}{3}$$

$$BandWidth_t = 2 * F * (TP_t) \qquad (IT.28)$$

En este trabajo se utiliza la ecuación IT.28 como variable. Las bandas superior e inferior están a F desviaciones típicas por encima y por debajo de la banda intermedia. En este caso $F = 2$.

En todos los casos:

- $Close_t$ es el precio de cierre del stock en el momento t
- Low_t es el precio mínimo del stock en el momento t

- $High_t$ es el precio máximo del stock en el momento t

V.1.4 ***Experimentos***

Partición muestras de entrenamiento, validación y test

Una vez calculada la variable respuesta y las variablex explicativas/predictoras se procede a construir los modelos predictivos. En primer lugar se particionan los conjuntos de datos en las muestras de entrenamiento, validación y test. En la muestra de entrenamiento se encuentran los datos con los que se entrarán los modelos, es decir, son los datos a partir de los cuales los modelos van observar el fenómeno de estudio y van a aprender sobre él. La muestra de validación se utiliza para obtener una optimización de los hyper parámetros. En inglés este proceso se llama *hyper-parameter fine tunning*. Finalmente, la muestra test la forman los datos sobre los cuales se va a evaluar el rendimiento de los modelos entrenados con los datos *train + validation* después de tunear los hyper parámetros.

Las series temporales de precios necesitan un tratamiento especial al momento de hacer las particiones entre muestras de entrenamiento, validación y test ya que se debe mantener la estructura temporal inherente en los datos.

- Muestra de entrenamiento: 2000-2015
- Muestra de validación: 2016
- Muestra de test: 2017-2018

Los modelos que se construyen requieren una muestra de entrenamiento considerable al tener que captar largas tendencias en los precios. Además los hyper parámetros que se deben optimizar, en el caso del Random Forest, no son demasiados y por eso se decide utilizar 17 años de entrenamiento y sólo 6 meses para la muestra de validación y test.

Modelización y resultados

Random Forest

Para realizar los experimentos utilizando este modelo descrito en el apartado 6.1 del presente

trabajo se optimiza el parámetro `mtry` (véase Philipp Probst & Boulesteix, 2018). Acorde con la documentación de la función `randomForest` que se puede consultar en (Liaw & Wiener, 2002), el parámetro `mtry` controla el número de variables aleatóriamente muestreadas que son candidatas en cada *split* del árbol. Es decir, es el número de variables a tener en cuenta en cada *split* a partir de las cuales se selecciona la que mejor particiona los datos. Esta optimización se hace sobre la muestra de validación utilizando como métrica el % de casos mal clasificados. Esto significa que se selecciona el valor de `mtry` que propocione un menor porcentage de casos mal clasificados (1-accuracy).

Seguidamente se muestra una tabla resumen con el valor óptimo del hyper parámtero `mtry` para cada tipo de alisado de cada empresa. Además se incluye en la tabla el valor de *accuracy* obtenido con cada modelo sobre la muestra de validación utilizando el parámetro óptimo `mtry` para cada tipo de alisado.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	mtry1m	accuracy1m	mtry2m	accuracy2m	mtry3m	accuracy3m
EMA30	13	63.09524	2	54.76190	2	60.71429
EMA60	2	63.49206	2	56.34921	2	49.60317
EMA90	2	58.33333	4	75.00000	2	67.46032
Alisado exponencial	2	63.09524	16	66.26984	3	63.49206

Tabla V.17: Coca Cola CO.: valores optimizados para mtry y accuracy obtenida

Como se puede observar para la empresa Coca-Cola CO. el mejor resultado, es decir la *accuracy* obtenida más alta, para la predicción a 1 mes se obtiene con el alisado correspondiente a una media móbil exponencial a 30 días con un valor de 63.5%. Para la predicción a 2 meses el mejor resultado es de 75% de accuracy y se obtiene alisando los datos con una EMA de 90 días (el mejor resultado global para la muestra de validación). Por lo que respecta a la predicción a 3 meses el mejor resultado se obtiene alisando los datos con una EMA 90 días con un valor de 67.46%. Para esta empresa no se aprecia un patrón de crecimiento de la *accuracy* sino más bien unos resultados repartidos más o menos uniformente entre los distintos experimentos realizados.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	mtry1m	accuracy1m	mtry2m	accuracy2m	mtry3m	accuracy3m
EMA30	29	79.36508	11	67.46032	14	66.66667
EMA60	26	80.15873	2	66.66667	23	72.22222
EMA90	28	87.30159	2	69.84127	18	73.80952
Alisado exponencial	27	78.96825	11	68.65079	11	63.09524

Tabla V.18: Apple Inc.: valores optimizados para mtry y accuracy obtenida

Como se puede apreciar en la tabla 6.2, para la empresa Apple Inc. la situación es diferente. Se obtienen, en general, mejores resultados que en el caso de la empresa Coca Cola. Para

la predicción a 1 mes el mejor resultado se obtiene con un alisado utilizando EMA de 90 días con una *accuracy* del 87.30%. Para la predicción hecha a 2 meses el mejor resultado es un 69.84% de accuracy obtenido con un alisado EMA 90 días. Para el caso en el que la predicción es a 3 meses el mejor resultado es de un 73.81% de *accuracy* con un alisado de EMA 90 días. Es decir, en el caso de la empresa Apple Inc el mejor resultado global se obtiene prediciendo si la media móbil exponencial a 90 días del precio de cierre estará más alta o no a 1 meses vista. En este caso los mejores resultados se obtienen con una predicción hecha a 1 mes vista con los datos fuertemente alisados. El hecho de predecir si al cabo de un mes la media móbil exponencial del precio de cierre calculada con 90 días será más alta o no, es de ayuda para el inversor ya que significa que el precio de cierre al cabo de un mes será más alto. Esto se debe al hecho de que si la EMA 90 días será más alta al cabo de un mes significa que los precios de ese último mes han sido más altos.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	mtry1m	accuracy1m	mtry2m	accuracy2m	mtry3m	accuracy3m
EMA30	9	79.76190	2	53.96825	2	63.88889
EMA60	2	73.80952	2	54.36508	5	76.19048
EMA90	10	74.60317	31	57.93651	2	80.15873
Alisado exponencial	2	79.76190	3	59.92063	4	77.38095

Tabla V.19: American Express CO.: valores optimizados para mtry y accuracy obtenida

Los resultados obtenidos para la empresa American Express CO. son globalmente mejores en los casos en los que la predicción está hecha para el siguiente mes y para al cabo de 3 meses. Para el caso en el que la predicción se hace a 1 mes, el mejor resultado obtenido es de 79.76% de *accuracy* en la muestra de validación y se corresponde a un alisado de los datos utilizando la fórmula del alisado exponencial. A su vez se obtiene el mismo resultado con los datos alisados mediante una media móbil exponencial de 30 días. Para el caso en el que la predicción se hace a 2 meses el mejor resultado que se obtiene es de 59.92% de *accuracy* en el caso de alisar los datos con la fórmula del alisado exponencial. Este resultado es ligeramente superior al que se obtendría con una predicción aleatória (50%). El resultado que se obtiene en el caso de la predicción hecha a 3 meses ya que en el mejor de los casos, cuando se alisan los datos con una media móbil exponencial de 90 días, es de 80.16% de *accuracy* (mejor resultado global). Esto significa que, para esta empresa, la mejor clasificación se obtiene prediciendo si la EMA 90 días será más alta o no al cabo de 3 meses. En esencia, este caso de predicción representa que se está analizando si el incremento de los precios ha hecho que la media móbil exponencial haya subido los últimos 3 meses.

	Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
	mtry1m	accuracy1m	mtry2m	accuracy2m	mtry3m	accuracy3m
EMA30	4	67.85714	2	50.00000	2	41.66667
EMA60	3	64.68254	2	50.39683	2	54.36508
EMA90	3	65.87302	2	55.15873	13	64.28571
Alisado exponencial	4	69.84127	3	58.33333	3	55.95238

Tabla V.20: Wells Fargo and CO.: valores optimizados para mtry y accuracy obtenida

En el caso de Wells Fargo & CO. los resultados sobre la muestra de validación se pueden observar en la tabla 6.4. Son, en general, peores que los resultados obtenidos con la empresa Apple Inc. y American Express CO.. Para la predicción hecha a 1 mes el mejor resultado se obtiene utilizando la fórmula de alisado exponencial (2) con una *accuracy* de 69.84% (mejor resultado global para esta empresa). Esto significa que sobre la muestra de validación aproximadamente el 70% de los casos queda bien clasificado. En el caso de la predicción a 2 meses el mejor resultado obtenido es de 58.33% de *accuracy* obtenido alisando los datos con la fórmula de alisado exponencial (2). El mejor resultadao cuando la predicción se hace a 3 meses vista es del 64.28% de *accuracy*.

Resultados sobre muestra test

Una vez optimizados los valores del hyper parámetro `mtry` se entrenan los modelos con las muestras de entrenamiento + validación con el objetivo de testar el modelo con los datos de test. En total se construyen 48 modelos: 4 por cada tipo de variable respuesta y, a su vez, 4 por cada empresa. En las tablas siguientes se muestran los valores de *accuracy*, *sensitivity* y *specificity* de los distintos modelos sobre los datos test.

	Predicción 1 mes			Predicción 2 meses			Predicción 3 meses		
	acc1m	sensi1m	speci1m	acc2m	sensi2m	speci2m	acc3m	sensi3m	speci3m
EMA30	41.16	95.21	17.61	45.12	94.17	27.86	58.05	98.17	44.88
EMA60	64.45	86.51	56.62	77.66	91.67	73.37	75.51	85.71	72.04
EMA90	84.20	92.56	81.39	65.08	93.28	55.26	69.84	91.53	61.92
Alisado exponencial	48.65	90.91	32.66	43.60	93.86	27.09	66.44	89.09	58.91

Tabla V.21: Coca Cola CO.: Metricas de rendimiento sobre muestra test

	Predicción 1 mes			Predicción 2 meses			Predicción 3 meses		
	acc1m	sensi1m	speci1m	acc2m	sensi2m	speci2m	acc3m	sensi3m	speci3m
EMA30	67.15	60.16	69.69	51.41	42.05	53.62	18.37	100.00	9.09
EMA60	64.86	77.17	61.95	55.31	61.54	54.94	59.64	13.04	62.20
EMA90	74.22	63.27	75.46	63.99	27.27	65.83	19.50	100.00	16.86
Alisado exponencial	54.26	56.78	53.44	52.28	97.10	44.39	41.72	62.50	39.65

Tabla V.22: Apple Inc.: Metricas de rendimiento sobre muestra test

	Predicción 1 mes			Predicción 2 meses			Predicción 3 meses		
	acc1m	sensi1m	speci1m	acc2m	sensi2m	speci2m	acc3m	sensi3m	speci3m
EMA30	54.26	89.23	41.31	54.88	97.20	42.09	50.79	68.33	48.03
EMA60	70.27	91.46	65.91	58.57	97.78	54.33	53.51	31.25	55.26
EMA90	70.27	96.97	68.30	59.65	100.00	56.84	66.67	0.00	68.69
Alisado exponencial	66.53	83.33	60.94	47.29	100.00	36.55	47.62	72.55	44.36

Tabla V.23: American Express CO.: Metricas de rendimiento sobre muestra test

	Predicción 1 mes			Predicción 2 meses			Predicción 3 meses		
	acc1m	sensi1m	speci1m	acc2m	sensi2m	speci2m	acc3m	sensi3m	speci3m
EMA30	67.98	68.38	67.61	68.55	71.30	65.80	62.13	77.48	39.66
EMA60	69.85	65.96	73.58	63.99	53.50	75.69	63.27	68.16	57.14
EMA90	78.38	77.08	79.82	67.25	60.82	74.54	65.99	80.93	48.78
Alisado exponencial	67.98	57.87	77.64	71.58	72.29	70.87	59.64	87.01	22.46

Tabla V.24: Wells and Fargo CO.: Metricas de rendimiento sobre muestra test

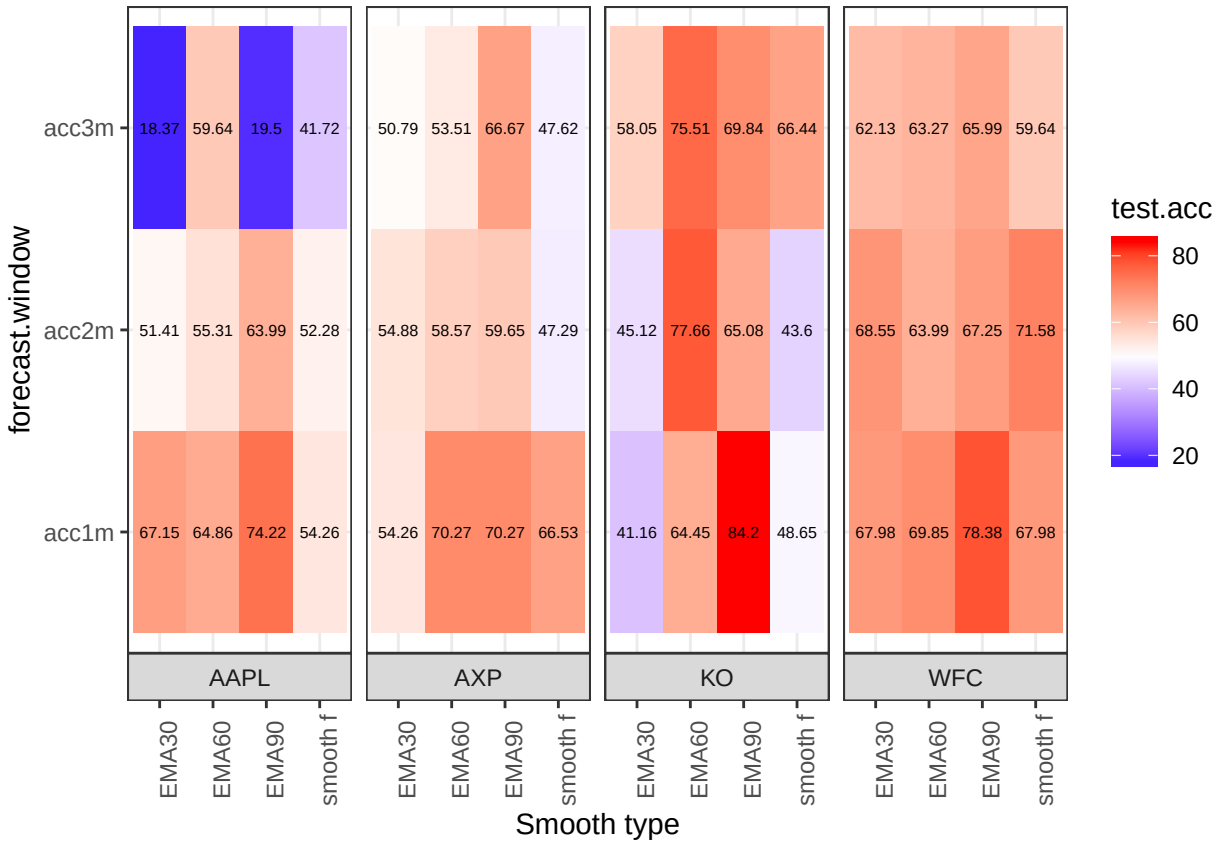


Figura V.5: Heatmap de la accuracy obtenida con los modelos Random Forest sobre muestra test. Fuente: elaboración propia

La representación de la accuracy obtenida sobre la muestra de prueba con los distintos modelos Random Forest permite visualizar de una manera más global. En general los mejores resultados se obtienen prediciendo si el precio de cierre será superior o inferior al cabo de un mes. Este resultado se puede observar de manera general para las 4 empresas estudiadas. Otro resultado que se puede apreciar fácilmente para las 4 empresas es el hecho de que el mejor rendimiento se obtiene a corto plazo (prediciendo a 1 mes) utilizando los datos alisados con una EMA 90 días. Este resultado tiene una fácil interpretación: este tipo de alisado es el más agresivo en cuanto a la fuerza del mismo. Esto significa que se remueven de una manera más clara los movimientos día a día y se conserva de una manera más fuerte la tendencia general.

Sin embargo cuando se analiza el resultado para cada una de las empresas, se puede apreciar que el resultado es totalmente distinto. Para el caso de AAPL se puede observar que el rendimiento del modelo predictivo se reduce a medida que se aleja en el tiempo la predicción. Cuando se intenta predecir si el precio será superior o inferior al cabo de tres meses, el rendimiento que se obtiene es inferior al 50% de accuracy

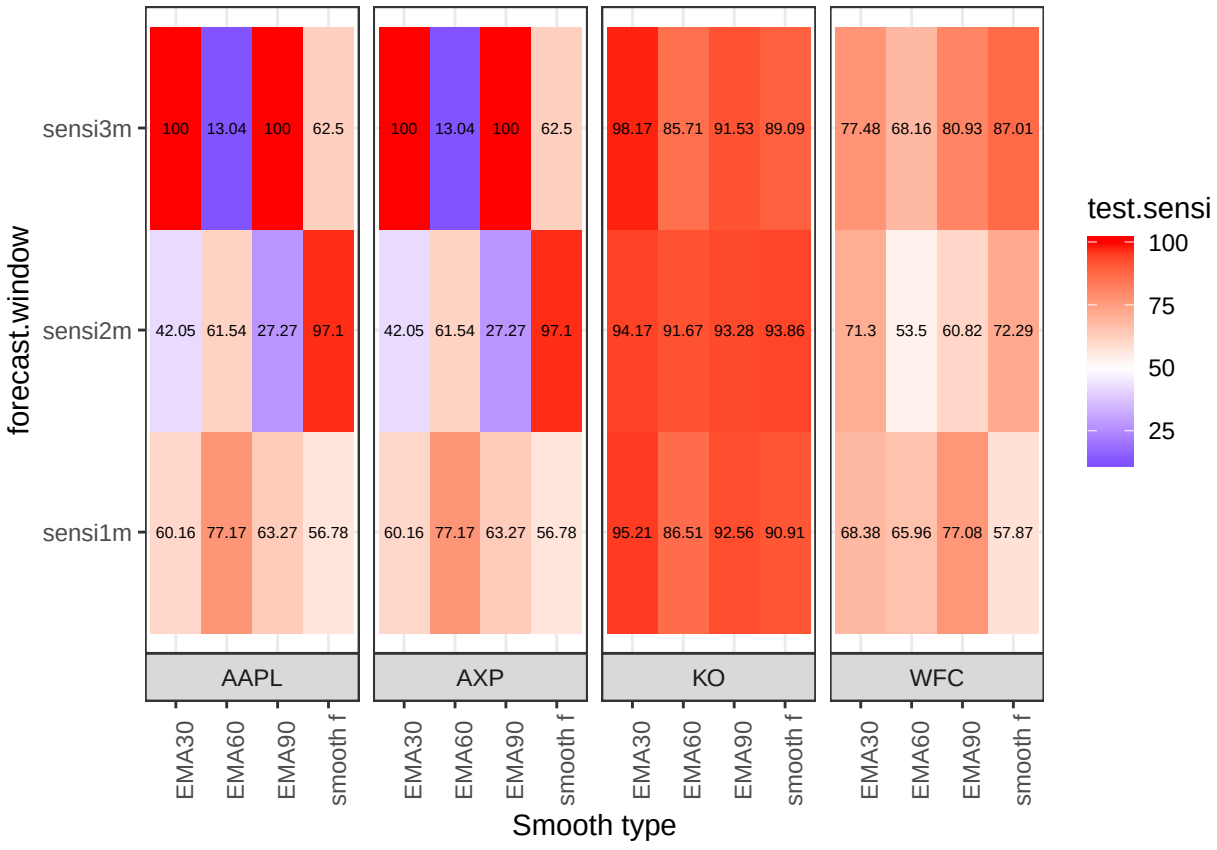


Figura V.6: Heatmap de la sensitividad obtenida con los modelos Random Forest sobre muestra test. Fuente: elaboración propia

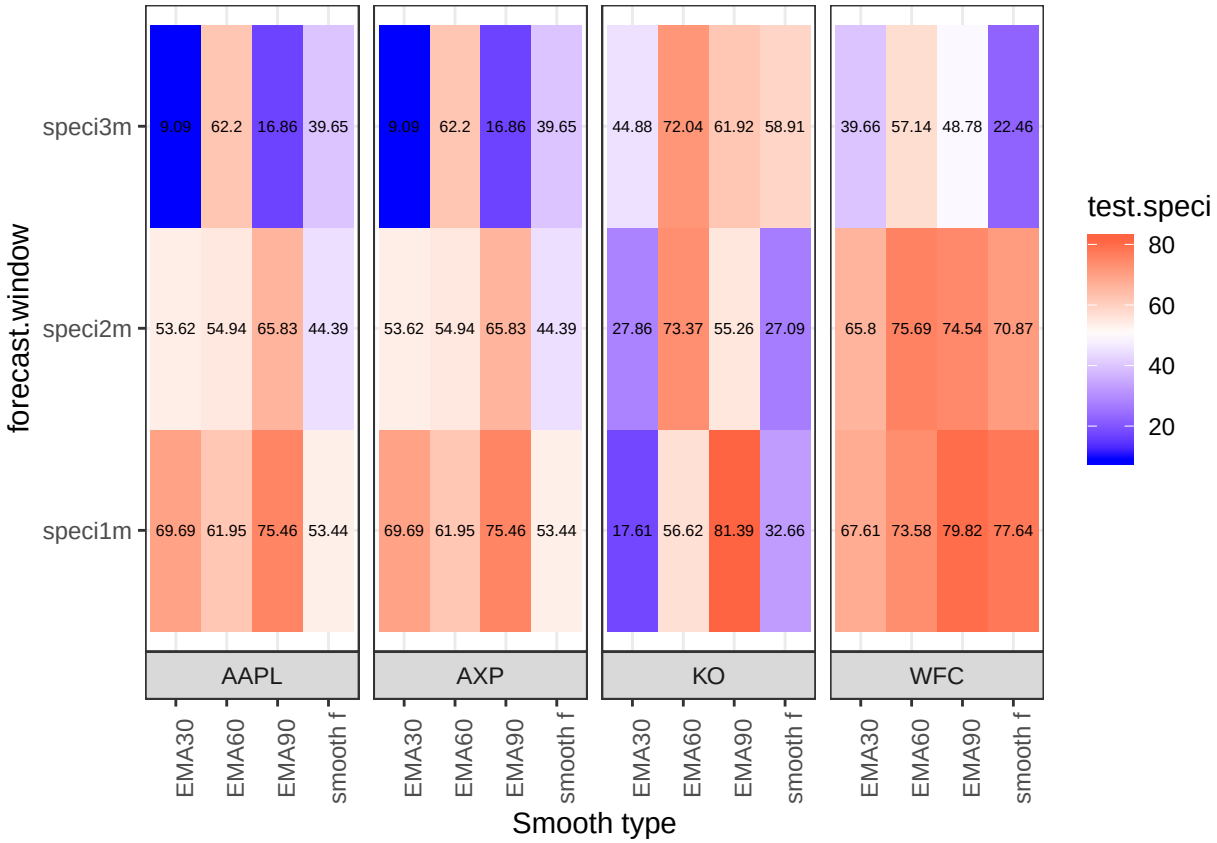


Figura V.7: Heatmap de la especificidad obtenida con los modelos Random Forest sobre muestra test. Fuente: elaboración propia

Importancia de las variables

A la hora de construir los árboles que conforman el bosque aleatorio ciertas variables son más importantes que otras. Éstas son las variables que consiguen partir mejor la base de datos, es decir, son las que consiguen que los nodos hijos sean los más puros posibles. En otras palabras: las variables más importantes a la hora de crear las particiones son aquellas que consiguen una mayor reducción de la impureza. En este sentido en las tablas siguientes se muestran las 4 variables más importantes para cada modelo construido con un alisado distinto. Dichas variables estan ordenadas de mayor a menor decrecimiento medio en la medida de impureza de Gini (Liaw & Wiener, 2002).

	Variables ordenadas de mayor a menor importancia de izq. a dcha.							
	Var1	Var1.n	Var2	Var2.n	Var3	Var3.n	Var4	Var4.n
EMA30 + target 1mes	210.17236	ROC1	161.98600	mom1	150.57163	wAD	146.11927	atr
EMA30 + target 2mes	127.77545	atr	121.04040	sma10	118.66516	wAD	113.56721	OBV
EMA30 + target 3mes	141.33800	atr	125.01709	wAD	122.34676	sma10	121.82152	tr
EMA60 + target 1mes	113.69941	ROC1	102.59613	mom1	96.56371	mom2	85.95530	mom4
EMA60 + target 2mes	94.68379	wAD	91.55005	sma10	90.65286	atr	85.77600	OBV
EMA60 + target 3mes	106.37693	wAD	104.92131	OBV	104.06340	atr	99.49582	sma10
EMA90 + target 1mes	104.72720	ROC1	96.25954	mom2	95.08068	mom1	92.52522	mom3
EMA90 + target 2mes	113.73057	ROC1	102.42608	mom1	101.54729	sma10	101.16146	wAD
EMA90 + target 3mes	101.19810	sma10	99.34100	wAD	93.82961	OBV	85.61232	atr
exp smooth + target 1mes	103.17506	mom1	101.03113	ROC1	89.66250	atr	83.22613	mom3
exp smooth + target 2mes	233.27022	wAD	221.35309	atr	186.85535	sma10	169.24715	OBV
exp smooth + target 3mes	151.05664	atr	148.20306	wAD	138.30116	sma10	134.13610	OBV

Tabla V.25: Coca Cola CO.: Importancia de las variables en los modelos Random Forest

	Variables ordenadas de mayor a menor importancia de izq. a dcha.							
	Var1	Var1.n	Var2	Var2.n	Var3	Var3.n	Var4	Var4.n
EMA30 + target 1mes	512.15204	ROC1	98.93951	mom1	96.63055	tdi	88.19676	di
EMA30 + target 2mes	144.51919	ROC1	121.75086	atr	118.99892	sma10	102.01047	wAD
EMA30 + target 3mes	186.49601	sma10	144.89187	ROC1	134.81112	OBV	134.41703	wAD
EMA60 + target 1mes	681.73270	ROC1	102.46408	mom2	87.15053	mom1	74.47985	ADX
EMA60 + target 2mes	90.74460	ROC1	76.35801	RSI	75.34084	ROC9	67.89531	wAD
EMA60 + target 3mes	362.46061	ROC1	144.70053	wAD	142.20835	sma10	130.94650	OBV
EMA90 + target 1mes	848.39350	ROC1	105.87283	mom1	61.86987	ADX	53.94166	sma10
EMA90 + target 2mes	97.99641	ROC1	85.52830	PNratio	73.11995	mom2	71.66828	mom3
EMA90 + target 3mes	330.79862	ROC1	165.57534	PNratio	128.86718	wAD	111.52754	ROC9
exp smooth + target 1mes	561.39281	ROC1	147.15549	mom1	82.34155	tdi	76.80978	OBV
exp smooth + target 2mes	200.86483	ROC1	113.88077	sma10	112.14525	atr	105.30881	wAD
exp smooth + target 3mes	173.31736	sma10	144.74850	ROC1	141.58538	OBV	132.46565	wAD

Tabla V.26: Apple Inc.: Importancia de las variables en los modelos Random Forest

	Variables ordenadas de mayor a menor importancia de izq. a dcha.							
	Var1	Var1.n	Var2	Var2.n	Var3	Var3.n	Var4	Var4.n
EMA30 + target 1mes	165.25920	mom1	130.46414	mom2	127.02326	ROC1	119.09421	wAD
EMA30 + target 2mes	120.43286	sma10	113.36900	wAD	111.34118	OBV	104.51818	atr
EMA30 + target 3mes	131.57874	atr	116.53550	sma10	115.22161	OBV	110.56381	wAD
EMA60 + target 1mes	118.07516	mom1	102.09144	ROC1	98.61755	mom2	87.88118	mom3
EMA60 + target 2mes	99.28404	atr	97.13480	wAD	94.38705	sma10	93.67309	OBV
EMA60 + target 3mes	181.81561	atr	144.75734	sma10	141.83627	tr	125.10676	wAD
EMA90 + target 1mes	268.16814	mom2	266.96414	mom1	167.93294	mom3	131.28383	ROC1
EMA90 + target 2mes	323.00340	mom1	212.68465	mom2	187.82489	tr	141.22069	ADX
EMA90 + target 3mes	107.27084	tr	103.53848	atr	89.91923	sma10	82.61953	OBV
exp smooth + target 1mes	106.15558	ROC1	99.54442	mom2	94.38816	mom1	82.79137	mom3
exp smooth + target 2mes	130.72584	sma10	128.00214	wAD	123.68470	OBV	110.90271	atr
exp smooth + target 3mes	167.67349	atr	151.07365	sma10	140.54419	wAD	138.35228	OBV

Tabla V.27: American Express CO.: Importancia de las variables en los modelos Random Forest

	Variables ordenadas de mayor a menor importancia de izq. a dcha.							
	Var1	Var1.n	Var2	Var2.n	Var3	Var3.n	Var4	Var4.n
EMA30 + target 1mes	103.91043	ROC1	96.13491	mom1	95.50823	wAD	94.68811	sma10
EMA30 + target 2mes	113.57309	sma10	111.78549	OBV	110.34820	wAD	99.16066	atr
EMA30 + target 3mes	130.20298	sma10	124.84470	OBV	121.70352	wAD	117.96892	atr
EMA60 + target 1mes	117.78181	ROC1	100.53238	mom1	93.98690	mom2	91.03978	mom3
EMA60 + target 2mes	98.24817	atr	97.38317	sma10	95.30748	OBV	92.67769	wAD
EMA60 + target 3mes	121.26242	sma10	118.51177	OBV	114.25443	wAD	107.39861	atr
EMA90 + target 1mes	127.60044	ROC1	112.53211	mom2	111.11757	mom1	85.74894	mom5
EMA90 + target 2mes	86.82101	atr	84.25939	wAD	83.97373	sma10	81.06038	ROC1
EMA90 + target 3mes	208.36783	atr	201.44860	sma10	166.60851	OBV	162.17772	tr
exp smooth + target 1mes	104.95205	ROC1	102.02345	sma10	98.82828	mom1	94.50204	wAD
exp smooth + target 2mes	126.40056	sma10	119.46090	wAD	116.75480	OBV	108.83301	atr
exp smooth + target 3mes	158.43065	sma10	143.52487	wAD	137.79011	OBV	127.47351	atr

Tabla V.28: Wells and Fargo CO.: Importancia de las variables en los modelos Random Forest

Seguidamente se da respuesta a una serie de preguntas en relación a cuáles han sido las variables o indicadores técnicos más importantes en la creación de los distintos Random Forest:

i) Cuál ha sido la variable más importante en el *split* con la frecuencia absoluta más alta de **entre todos los modelos de todas las empresas**?

```
##
##  mom2    tr   wAD  mom1   atr sma10  ROC1
##     1     1     3     4    10    10    19
```

Como se observa a partir del resultado anterior la variable que se escoge más veces como variable más importante a la hora de hacer las particiones es el *Rate of Change*, seguida por la *simple moving average 10 days* y el *Average True Range*.

También se muestra en proporción:

```
##
##  mom2    tr   wAD  mom1   atr sma10  ROC1
##  2.08  2.08  6.25  8.33 20.83 20.83 39.58
```

ii) Cuál ha sido la variable más importante en el *split* con la frecuencia relativa más alta de

entre todos los modelos para cada empresa?

Coca-Cola CO.

```
##
##  ROC1   atr   wAD  mom1 sma10
## 33.33 25.00 25.00  8.33  8.33
```

Apple Inc.

```
##
##  ROC1 sma10
## 83.33 16.67
```

American Express CO.

```
##
##   atr  mom1 sma10  mom2  ROC1    tr
## 33.33 25.00 16.67  8.33  8.33  8.33
```

Wells & Fargo CO.

```
##
## sma10  ROC1   atr
## 41.67 33.33 25.00
```

V.2 Dirección de movimiento del precio: análisis masivo con Random Forest

En el presente apartado se procede a utilizar el modelo de predicción de la dirección de movimiento del precio de cierre desarrollado en el apartado anterior en una aplicación masiva. El hecho de que se considere una aplicación masiva hace referencia a que el modelo SMD definido en la sección V.1 se aplica, después de construir una función que permite generalizar su aplicación, de una manera iterativa sobre un conjunto predefinido de *stocks*, paralelizando el cálculo. En este sentido se puede entender que es una aplicación masiva ya que no se aplica sobre un conjunto específico de empresas sino que se lanza de manera general para un conjunto mucho mayor de empresas. El objetivo es lo que distingue esta sección de la anterior: mientras que en la anterior el centro del análisis consiste en visualizar cómo rinde cada empresa con este tipo de modelos, lo interesante de este apartado es poder analizar más

globalmente (o masivamente) el resultado de este tipo de modelos sobre un conjunto con un gran número de empresas. Esta aplicación no deja de ser un ejemplo adaptado para este trabajo y limitado por la capacidad computacional disponible en el momento de su creación. Cabe decir que la presente lógica de aplicación del modelo SMD se puede extender fácilmente si se dispone de mayor capacidad computacional.

En primer lugar se descargan los datos de 01-01-2000 a 06-05-2019 de 165 empresas con los símbolos correlativos del NYSE. La selección de las empresas del NYSE es consecutiva empezando con el símbolo ETG. Posteriormente se aplica de manera iterativa el modelo de predicción de la dirección de movimiento definido en el apartado anterior utilizando el modelo Random Forest con los parámetros por defecto (número de variables a probar en cada split = sqrt(p) donde p es el número de regresores y número de árboles a crecer). Las particiones de muestra de entramiento y test que se utilizan para evaluar el modelo se construyen con las siguientes proporciones: train = 85% y test = 15%. Para crear estas particiones se mantiene, como en el caso anterior, el factor temporal. Sin embargo, en este caso se definen las particiones en proporción al no tener el mismo número de datos disponibles para todas las empresas. Por lo que respecta al tipo de alisado en los datos que se elabora antes de construir las variables y ajustar los modelos, para este apartado se ha restringido el perímetro y sólo se ha utilizado un alisado exponencial del precio con una EMA a 90 días. Los símbolos o compañías que se analizan en este apartado son las siguientes:

```
##   [1] "ETG"   "ETH"   "ETJ"   "ETM"   "ETN"   "ETO"   "ETR"   "ETV"
##   [9] "ETW"   "ETY"   "EV"    "EVC"   "EVF"   "EVG"   "EVN"   "EVR"
##  [17] "EVRI"  "EVT"   "EXC"   "EXD"   "EXG"   "EXK"   "EXP"   "EXPR"
##  [25] "EXR"   "F"     "FAF"   "FAM"   "FBC"   "FBP"   "FC"    "FCAU"
##  [33] "FCF"   "FCN"   "FCT"   "FCX"   "FDP"   "FDS"   "FDX"   "FE"
##  [41] "FENG"  "FEO"   "FF"    "FFA"   "FFC"   "FFG"   "FGB"   "FGP"
##  [49] "FHN"   "FICO"  "FII"   "FIS"   "FIX"   "FL"    "FLC"   "FLO"
##  [57] "FLR"   "FLS"   "FLT"   "FLY"   "FMC"   "FMN"   "FMO"   "FMS"
##  [65] "FMX"   "FMY"   "FN"    "FNB"   "FNF"   "FNV"   "FOE"   "FOF"
##  [73] "FOR"   "FR"    "FRA"   "FRC"   "FRO"   "FRT"   "FSD"   "FSM"
##  [81] "FSS"   "FT"    "FTI"   "FTK"   "FTS"   "FUL"   "FUN"   "G"
##  [89] "GAB"   "GAM"   "GATX"  "GBAB"  "GBL"   "GBX"   "GCAP"  "GCI"
##  [97] "GCO"   "GCV"   "GD"    "GDL"   "GDO"   "GDOT"  "GDV"   "GE"
## [105] "GEF"   "GEL"   "GEN"   "GEO"   "GES"   "GF"    "GFF"   "GGB"
## [113] "GGG"   "GHC"   "GHL"   "GIB"   "GIL"   "GIM"   "GIS"   "GJH"
## [121] "GJO"   "GJP"   "GJR"   "GJS"   "GJT"   "GJV"   "GLP"   "GLT"
## [129] "GLW"   "GM"    "GME"   "GNC"   "GNRC"  "GNT"   "GNW"   "GOF"
## [137] "GOL"   "GOLD"  "GPC"   "GPI"   "GPK"   "GPM"   "GPN"   "GPRK"
## [145] "GPS"   "GPX"   "GRA"   "GRC"   "GRX"   "GS"    "GS-PD" "GSH"
## [153] "GSK"   "GSL"   "GTN"   "GTS"   "GTT"   "GTY"   "GUT"   "GVA"
## [161] "GWR"   "GWW"   "GYB"   "GYC"   "H"
```

Al ser un ejercicio con un coste computacional elevado, para el presente trabajo se ha decidido **paralelizar el cálculo** de los distintos modelos utilizando los núcleos del ordenador con el que se ha trabajado. En este caso el cálculo se ha paralelizado utilizando 3 núcleos, de manera que en cada uno de ellos se ha calculado por separado todo el proceso de modelización SMD: Recogida de datos, procesamiento de alisado con EMA 90, *feature extraction*, modelización y evaluación de los resultados utilizando la *accuracy* obtenida con el modelo de clasificación. El proceso de cálculo de los 3 modelos (uno para cada ventana de predicción en el futuro) con las 165 empresas tomó alrededor de 5 horas aun habiendo paralelizado el cálculo con 3 núcleos.

La evaluación de los resultados se hace desde tres perspectivas. En primer lugar se elabora un *heatmap* con la *accuracy* obtenida sobre muestra de prueba con cada una de las 3 ventanas de predicción hacia el futuro: 1, 2 y 3 meses. El objetivo de esta herramienta gráfica es múltiple. Por un lado permite la visualización global de los resultados, a la vez que aporta una primera imagen de lo buenos que son los resultados. La idea es que si se pinta el *heatmap* de los resultados obtenidos y el resultado es generalmente bueno (alta *accuracy*) el color predominante en el gráfico debería ser el rojo, indicando que la mayoría de los valores están por encima del umbral del 50%. Seguidamente se presenta el *heatmap*:

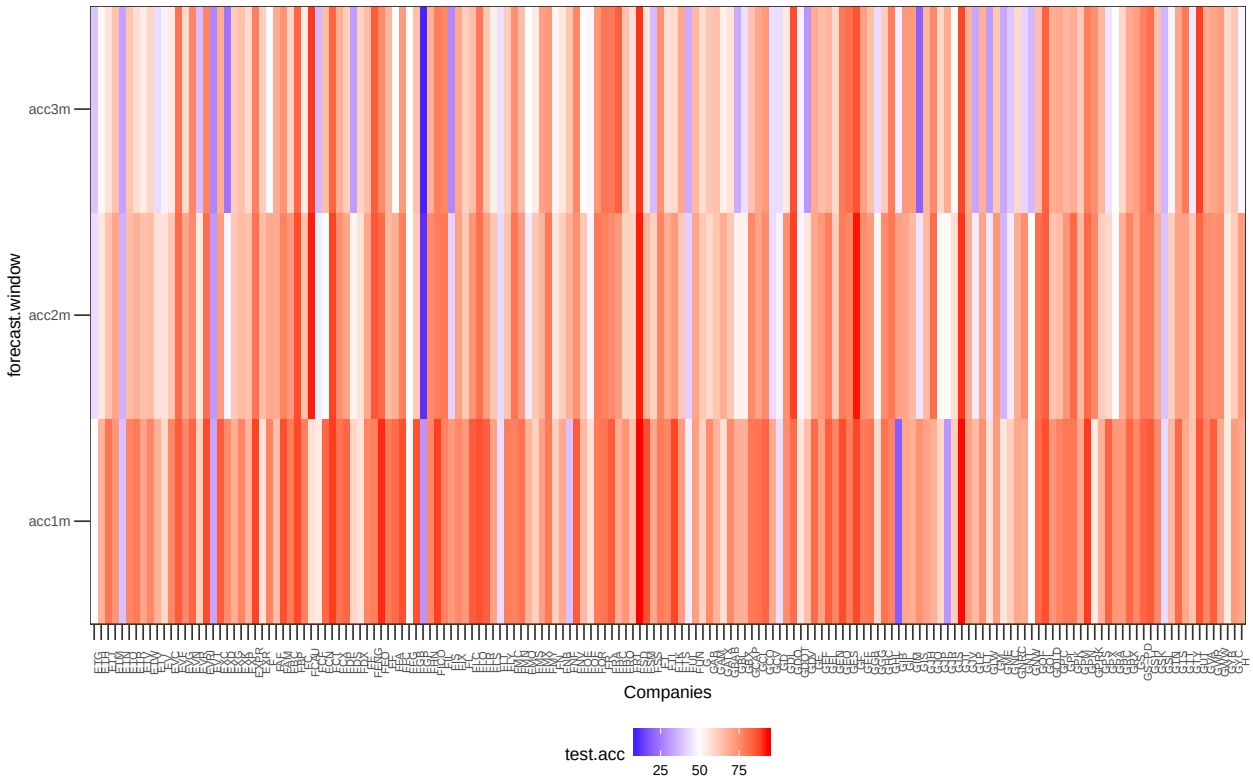


Figura V.8: Heatmap de la accuracy obtenida sobre muestra test SMD masivo para 165 empresas. Fuente: elaboración propia

Como se puede apreciar en la figura V.8 el resultado es, generalmente, positivo. El color que predomina en el *heatmap* es el color rojizo, señal de que los resultados son generalmente superiores al 50% de accuracy sobre la muestra test. Es un buen resultado. Esto significa que para la mayoría de empresas sobre las que se ha probado masivamente este tipo de modelización el resultado es positivo. Sin embargo, también permite observar como para algunas empresas el resultado no es tan positivo en según que ventana de predicción, así como diversas empresas cuyo rendimiento no es generalmente elevado, al presentar unos valores de *accuracy* inferiores al 50% en todas las ventanas de predicción. Estos hechos se aprecian a partir de los valores en azul presentes en el *heatmap*. El hecho de que para algunas empresas el resultado en términos de *accuracy* sean tan malos se puede deber a varias razones. En primer lugar es posible que, para algunas empresas, este tipo de modelo no mejore la predicción base que se podría hacer lanzando una moneda. En otras palabras, para determinadas empresas el hecho de utilizar datos alisados y indicadores técnicos para predecir la dirección de movimiento puede no mejora la predicción *naive*. En segundo lugar, los malos resultados se pueden deber a otro simple hecho: durante el periodo de prueba (test) el comportamiento de los datos es completamente nuevo, es decir, totalmente diferente de los patrones mostrados en la muestra de entrenamiento. Esto explica el hecho de los malos resultados en muestra test ya que el modelo que se entrena con los datos de entrenamientos aprende una serie de relaciones que no se repiten durante el periodo de prueba. En otras palabras: el modelo se evalúa con unos datos que no se corresponden con los datos con los cuales se entrenó el modelo y, por lo tanto, el resultado obtenido en términos de *accuracy* son realmente malos (inferiores a un 50%).

Para ilustrar este último punto se analiza en detalle una de las empresas con las que se obtienen peores resultados. Es el caso de la empresa Ferrellgas (FGP), una empresa americana de distribución de gas natural. Los resultados de *accuracy* obtenidos con los datos de FGP son los siguientes:

```
##    acc1m acc2m acc3m stock
## 49 30.04 12.04  10.1   FGP
```

El gráfico V.9 siguiente muestra toda la serie temporal con una línea que separa las muestras de entrenamiento y test:

Figura V.9: Precio de cierre empresa Ferrellgas (FGP). Fuente: elaboración propia

Como se puede observar en el gráfico anterior, los malos resultados que obtiene el modelo SMD aplicado a los datos alisados con una EMA 30 de la empresa FGP se deben al hecho de que el periodo de test muestra un comportamiento totalmente distinto al periodo de entrenamiento. El potente descenso que presentan los precios de cierre diarios al inicio del periodo de prueba representa un cambio estructural en los datos. Los modelos a 1, 2 y 3 meses que se entrenan con los datos anteriores a este drop no son capaces de predecir correctamente los datos del periodo de test ya que el patrón cambia totalmente. Sin embargo, los malos resultados obtenidos de accuracy no significan que el modelo no hubiera funcionado mejor si el periodo de entramiento no consistiera en un cambio estructural en los datos. Este punto realza el hecho de que, posiblemente, ningún modelo entrenado con los datos históricos de esta compañía sea útil para predecir el futuro de los precios de cierre al existir un cambio estructural en los datos a partir de 2016. En este caso, como es evidente, no se puede en ningún caso utilizar de manera práctica el modelo creado para predecir la dirección de movimiento del precio de cierre de esta empresa.

En segundo lugar se presenta una forma alternativa de visualizar el conjunto de los resultados de *accuracy* obtenidos con las diferentes empresas y ventanas temporales. Al tener 165 empresas se tienen 165 valores de *accuracy* para cada una de las ventanas temporales con lo que se puede graficar la densidad que representa esta muestra de valores con tal de visualizar la distribución de los resultados. En este caso se decide utilizar la función `density` en vez de graficar simplemente el histograma de los datos para conseguir un resultado más suavizado. Además, la densidad de la distribución de los resultados de *accuracy* para cada ventana temporal se presente en un mismo gráfico para facilitar la interpretabilidad y comparación

de los resultados.

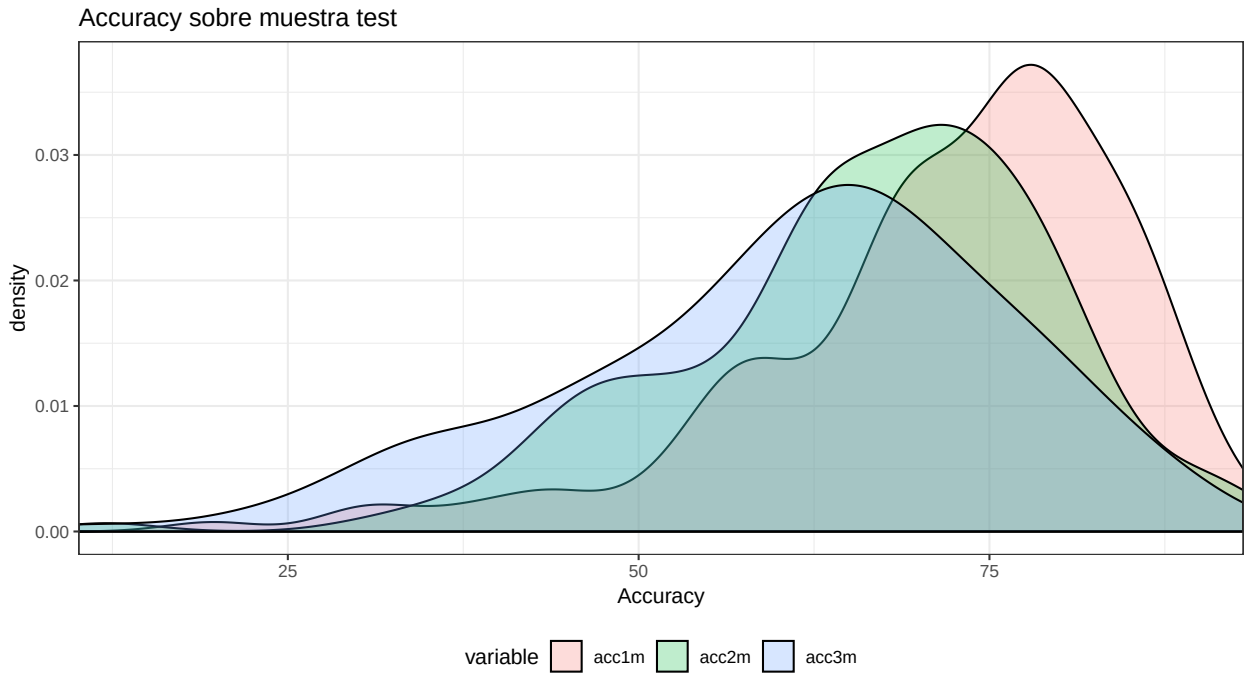


Figura V.10: Distribución de los resultados de accuracy sobre muestra test en las 3 ventanas temporales consideradas. Fuente: elaboración propia

Como se puede apreciar en la figura V.10 los resultados obtenidos son, en general, buenos en el sentido de que la mayoría de modelos mejoran la predicción naive de 50% en cada clase de la variable respuesta. En general las tres distribuciónes, correspondientes a las distintas ventanas de predicción, no son distribuciones simétricas. Al contrario, son distribución con asimetría a la izquierda (negativa) en el sentido de que tiene una cola pesada en los valores bajos. Esto significa que se obtienen en general resultados positivos pero que existen ciertos resultados negativos en las distintas ventanas de predicción que hacen que las distribuciones de los resultados estén sesgadas hacia valores bajos. Si se entra en el detalle, la primera conclusión que se puede extraer de este gráfico es la misma que en el apartado V.1: estos modelos parecen rendir mejor cuando la predicción está más cerca en el tiempo. Es decir, cuando más hacia el futuro se intenta predecir menor es la precisión obtenida. Esto se observa a partir de analizar el gráfico anterior y de ver que conforme se aumenta la ventana de predicción la media de los resultados obtenidos cada vez es más baja. Además, las colas tienden a aumentar cosa que significa que conforme más hacia el futuro se intenta predecir más resultados por debajo del umbral del 50% se obtienen.

Sin embargo la media de *accuracy* obtenida para todas las empresas es mayor del 50% en todas las ventanas de predicción con lo que se demuestra que, en general y excepto casos

excepcionales, los modelos de predicción de la dirección de movimiento del precio de cierre a partir de indicadores técnicos son capaces de aportar valor añadido y mejorar la predicción ingenua del 50% de probabilidad asociada al hecho de subir o bajar. El hecho de que se obtengan mejores resultados cuanto más corta sea la ventana de predicción hacia el futuro no debería sorprender al lector ya que tiene una explicación sencilla. Se predice mejor cuanto más cerca en el tiempo ya que las variables predictoras con las que se define el modelo de predicción de dirección del movimiento son indicadores técnicos extraídos a partir de los datos OHLC. Estos indicadores no dejan de ser estadísticos calculados a partir del precio y su propósito es el de indicarnos diferentes patrones o tendencias que se vienen dando en los precios.

Los indicadores técnicos pueden ayudar a predecir futuros movimientos en el precio pero es natural que los modelos predigan mejor cuanto más cerca en el tiempo. La descripción de la situación actual de un precio, que es en definitiva la información extraída a partir de los indicadores técnicos, es potencialmente útil para predecir el futuro pero es, evidentemente, más útil cuanto más cerca en el tiempo se mire hacia el futuro.

Este hecho se repite de manera habitual en cualquier proceso predictivo que involucre una descripción de la situación actual. En general, si se quiere predecir el futuro en base a cómo están las cosas en el momento actual, es lógico que se prediga mejor una situación cercana en el tiempo (que es consecuencia directa de las cosas que están ocurriendo en este momento) más que una situación lejana en el tiempo que es en parte consecuencia de las cosas que pasan en el momento actual pero también puede incluir otros elementos que pasan entre el momento actual y el momento lejano en el futuro.

Para analizar descriptivamente las distribuciones también se aplica un descriptivo básico de los 4 momentos centrales de estas distribuciones. Con este análisis descriptivo se pueden reafirmar los resultados anteriores: cuanto más alejada en el futuro se hace la predicción, mayor variabilidad se obtiene en el resultado para todas las empresas; cuanto más alejada en el futuro se hace la predicción, menor es la asimetría negativa presente en los resultados (ya que aparecen más frecuentemente resultados malos o menores del 50% de *accuracy*); cuanto más alejada en el futuro se hace la predicción, menor es el coeficiente de kurtosis, indicando una menor concentración de los resultados y, a su vez, una ampliación de las colas.

```
##              acc1m      acc2m      acc3m
## Mean      71.72824  66.165091  60.937394
## Variance 174.20021 173.896540 240.207407
## Skewness  -1.20486  -0.717815  -0.562482
## Kurtosis   1.74246   0.934674   0.028270
```

En tercer lugar se analizan los datos desde una perspectiva más descriptiva. Por un lado se

calcula un descriptivo básico de todas las muestras de *accuracy* obtenidas para cada ventana temporal.

```
##      acc1m           acc2m           acc3m
##  Min.   :19.61   Min.   :12.04   Min.   :10.10
##  1st Qu.:66.85   1st Qu.:60.20   1st Qu.:51.35
##  Median :75.12   Median :68.52   Median :63.44
##  Mean   :71.73   Mean   :66.17   Mean   :60.94
##  3rd Qu.:80.25   3rd Qu.:75.35   3rd Qu.:71.55
##  Max.   :93.09   Max.   :92.35   Max.   :90.27
```

Como se puede apreciar a partir del cálculo anterior, los mejores resultados se obtienen, en general, cuando la ventana de predicción es a 1 mes vista. Esto refuerza la conclusión desarrollada en el apartado anterior. Los modelos de predicción de la dirección de movimiento funcionan mejor cuanto más cerca en el tiempo se intente predecir. Para el caso de la predicción a un mes vista se obtiene una media de 71.73% de *accuracy* en las 165 empresas utilizadas mientras que la media es de 66.17% y 60.94% para los casos en los que la predicción es a 2 y 3 meses, respectivamente. Este descriptivo básico permite ver numéricamente la asimetría hacia la izquierda presente en las distribuciones de los resultados al ser en todos los casos la mediana superir a la media. Esto significa que las distribuciones son asimétricas con cola pesada a la izquierda. Otro hecho característico que se puede resaltar es que para los tres casos se alcanza más del 50% de *accuracy* en el primer quartil. Además, cabe destacar el hecho que de la *accuracy* máxima obtenida para las tres ventanas de predicción es superior al 90% sobre muestra de entrenamiento.

Por otro lado se presentan los ránkings con las 10 empresas que han obtenido mejores resultados en cada una de las ventanas temporales de previsión. Con esta tercera manera de visualizar los resultados se pueden analizar para qué empresas los modelos obtienen un mejor rendimiento sobre muestra test. En este sentido, se trata de encontrar las empresas para las cuales estos modelos funcionan mejor de manera que la confianza al utilizarlos en un caso real de inversión sea máxima. En la tabla V.28 que se presenta a continuación se muestran estos datos:

Predicción 1 mes		Predicción 2 meses		Predicción 3 meses	
acc1m	stock	acc2m	stock	acc3m	stock
93.09	FSD	92.35	GF	90.27	FSD
92.91	GJT	92.03	FSD	89.32	FCAU
91.05	FEO	91.99	FCAU	88.76	GUT
89.40	GPN	90.24	GJT	87.77	GJT
88.77	FCT	88.55	GDO	86.63	GDO
88.72	FICO	88.33	FCT	85.78	GF
88.36	EXPR	86.83	FBP	84.31	FENG
87.78	FTK	85.20	FENG	83.79	FRC
87.59	FBP	83.80	GOL	83.71	GOL
87.59	GF	82.44	GPK	83.05	FCT

Tabla V.29: Top 10 empresas con mejor rendimiento sobre muestra test para cada ventana de predicción.

Gracias a esta tabla se puede priorizar la inversión en las empresas con las que se obtiene un mejor resultado. La idea es que para las empresas dentro del top 10 en *accuracy* queda demostrado que este tipo de modelo tiene un buen rendimiento en términos de capacidad predictiva.. Si en el futuro los precios no experimentan ningún cambio estructural cabe esperar que se puedan utilizar estos modelos como ayuda en el momento de tomar una decisión de inversión o a la hora de crear un algoritmo de trading automático.

V.3 Dirección de movimiento del precio: Automatic Machine Learning

En el siguiente apartado se procede a utilizar el paquete de R **h2o** para construir distintos modelos de machine learning. La compañía **h2o** es una organización fundamentada en el concepto de *open-source* y constituye un sistema preparado para la computación ML. Concretamente en este apartado se procede a utilizar la función `AutoML` que se incluye en este paquete. El procedimiento es el siguiente: se aplica la función autoML con el mismo perímetro que en el apartado V.1 pero reducido, es decir, sobre 1 de las 4 empresas que se toman como ejemplo (KO) y con los alisados utilizando una EMA y ventanas de predicción. Se limita la aplicación de este apartado a una de las 4 empresas tomadas como ejemplo a causa de las limitaciones computacionales que se tienen durante la escritura de la presente tesis. En esencia ambos apartados son lo mismo en tanto que aplican el mismo tipo de modelo conceptual (predicción de la dirección de movimiento del precio de cierre). La diferencia radica en el tipo de modelo que se aplica: mientras que en el apartado V.1 se prueba el modelo Random Forest básico, en este apartado se deja a la función `AutoML` probar todos los modelos que incluye (definidos en el apartado IV.1). El marco para este apartado es el mismo que en el apartado V.1 en cuanto a las empresas utilizadas como ejemplo, los alisados y ventanas de predicción y las particiones de muestra train y test.

La función `AutoML` funciona de la siguiente manera. Con el objetivo de automatizar el proceso de entrenamiento de los modelos de machine learning de aprendizaje supervisado, se utilizan máquinas virtuales java para paralelizar el cálculo de los distintos modelos en los diferentes

núcleos del ordenador con el que se lance, de manera que se prueba un modelo distinto de manera paralela. Una vez calculados todos los modelos se evalúan sobre la muestra de prueba y se elabora lo que los creadores del paquete llaman *leaderboard* o, lo que es lo mismo, una lista con los modelos que han obtenido un mejor rendimiento sobre la muestra de prueba. En el presente trabajo se construye una función que aplica la función `AutoML` para la base de datos y compañía que se deseen. Este función coge el **mejor** modelo, es decir, el modelo que queda primero en esta tabla de modelos con mejores rendimientos. El lector debe notar que, dependiendo de la compañía, tipo de alisado en los datos y ventana de predicción, el mejor modelo escogido puede ser distinto.

En cuanto a los parámetros utilizados para lanzar la función `AutoML` se definen los siguientes. En primer lugar se fija una semilla para permitir la reproducibilidad de los resultados. También se define el parámetro `nfolds=0` para evitar la cross-validación ya que se desea mantener la estructura temporal de los datos. Además se limita el número máximo de modelos a probar a un número de 10 para agilizar el cálculo y se indica a la función que se quiere balancear la variable respuesta. En cuanto a las métricas con las que se evalúa el modelo se utiliza el AUC y la *accuracy*.

Uno de los argumentos en contra de este tipo de funciones es que normalmente son funciones *black box* en el sentido que no muestran lo que están haciendo por dentro. Sin embargo, el objetivo de este apartado es doble: por un lado, el de aplicar distintos modelos de machine learning a los aplicados en el apartado V.1 del presente trabajo sobre el mismo modelo conceptual de predicción de la dirección de movimiento del precio de cierre y, por otro, el de mostrar que en el futuro el valor añadido no estará en el hecho de contruir manualmente los modelos (se está probando que se puede hacer automáticamente) sino más bien el de definir modelos conceptuales que tengan buena capacidad predictiva.

En primer lugar se carga la librería y se inicializa el espacio de trabajo *h2o* con sus respectivas máquinas virtuales. El mensaje informativo sobre las características del clúster que se crea al inicializar la librería se muestran a continuación:

```
> h2o.init()

H2O is not running yet, starting it now...

Note:  In case of errors look at the following log files:
    C:\Users\i0386388\AppData\Local\Temp\RtmpOs2BfZ/h2o_I0386388_started_from_r.out
    C:\Users\i0386388\AppData\Local\Temp\RtmpOs2BfZ/h2o_I0386388_started_from_r.err

java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

Starting H2O JVM and connecting: . Connection successful!

R is connected to the H2O cluster:
    H2O cluster uptime:         3 seconds 740 milliseconds
    H2O cluster timezone:       Europe/Paris
    H2O data parsing timezone:  UTC
    H2O cluster version:        3.22.1.1
    H2O cluster version age:    4 months and 28 days !!!
    H2O cluster name:           H2O_started_from_R_I0386388_ypq345
    H2O cluster total nodes:    1
    H2O cluster total memory:   1.65 GB
    H2O cluster total cores:    4
    H2O cluster allowed cores:  4
    H2O cluster healthy:        TRUE
    H2O Connection ip:          localhost
    H2O Connection port:        54321
    H2O Connection proxy:       NA
    H2O Internal Security:      FALSE
    H2O API Extensions:         Algos, AutoML, Core V3, Core V4
    R Version:                  R version 3.4.4 (2018-03-15)
```

Figura V.11: Información obtenida al inicializar el clúster h2o

Como se puede apreciar en la figura anterior, entre la información que ofrece el paquete h2o ofrece inicializarse se encuentra el mensaje al arrancar las máquinas virtuales Java sobre las que trabaja, así como información relativa a la versión de los clústers que se están utilizando, el nombre, número de nodos y núcleos que se procede a utilizar, la memoria total y datos sobre la connectividad.

Para facilitar la aplicación de la función `AutoML` de h2o se crea una función personalizada con 3 parámetros. Basta con introducir la muestra de entramiento y la muestra de prueba, así como un carácter con el nombre de la empresa (en este caso, combinación de empresa + alisado + ventana de predicción). Esta función devuelve un dataframe con las métricas AUC y *accuracy* (con umbral de 0.5) ya que son las métricas que se utilizan en el presente trabajo para evaluar los modelos de clasificación binaria SMD. A su vez también se incluye la etiqueta, o nombre, del modelo que mejor ha rendido sobre la muestra de entrenamiento de entre todos los que la función `AutoML` prueba. Esta función se puede encontrar en el anexo de esta testis, en el apartado V.3.

Seguidamente se presentan los resultados para todas las combinaciones de alisado y ventana de predicción sobre la empresa Coca-Cola CO, ordenados de mayor a menor AUC.

Company	AUC	Accuracy	Model
KO_60_2m	0.9643794	0.6008677	GBM_4_AutoML_20190527_133104
KO_90_1m	0.9489899	0.7442827	DRF_1_AutoML_20190527_133503
KO_90_2m	0.9266426	0.7548807	GLM_grid_1_AutoML_20190527_133724_model_1
KO_60_3m	0.9196836	0.7596372	GLM_grid_1_AutoML_20190527_133305_model_1
KO_90_3m	0.9184683	0.6598639	GLM_grid_1_AutoML_20190527_133916_model_1
KO_fun_2m	0.9123692	0.8546638	DeepLearning_1_AutoML_20190602_194503
KO_30_3m	0.8804991	0.5374150	GBM_5_AutoML_20190527_132432
KO_fun_3m	0.8796759	0.5736961	GBM_grid_1_AutoML_20190602_194733_model_1
KO_60_1m	0.8794880	0.7879418	GLM_grid_1_AutoML_20190527_132816_model_1
KO_30_2m	0.8525171	0.4295011	GBM_5_AutoML_20190527_131607
KO_fun_1m	0.7841235	0.4989605	GLM_grid_1_AutoML_20190602_194134_model_1
KO_30_1m	0.6883664	0.5093555	GLM_grid_1_AutoML_20190527_125958_model_1

Tabla V.30: Resultados modelo SMD aplicado a la empresa Coca-Cola KO utilizando automatic machine learning en h2o

En general el resultado de los modelos es bueno. Por lo que respecta a la accuracy calculada con un umbral de 0.5 es en general superior al 50% de accuracy. A su vez, en obtienen unos AUC relativamente elevados. El paquete h2o ha demostrado ser efectivo en cuanto a la construcción de modelos con buen rendimiento. Todos menos un modelo aplicados sobre los datos de la empresa Coca Cola obtienen un AUC por encima de 0.85. En este caso el mayor AUC es de 0.96 y se obtiene con un Gradient Boosting Machine sobre los datos de Coca Cola alisados con una EMA a 60 días y prediciendo si el precio de cierre será más elevado o más bajo a 2 meses vista. Seguidamente aparecen los modelos sobre los datos alisados con EMA 90 días a 1 y 2 meses vista con AUC de 0.95 y 0.93 respectivamente. Los resultados obtenidos con el machine learning automático son sorprendentemente buenos en relación al AUC sobre muestra test. El hecho de que el rendimiento del modelo parezca mejor usando el AUC en vez de la accuracy se debe a que la métrica **accuracy** depende de un umbral predefinido (0.5), y posiblemente el umbral óptimo para los modelos definidos no sea el de 0.5.

En cuanto a los distintos modelos escogidos cabe destacar que en 5 de los 9 modelos escogidos para las 9 combinaciones de alisado y ventana de predicción ha sido la regresión logística, etiquetada en este caso como GLM_grid

Finalmente se hace un ejercicio interesante, el de comparar la *accuracy* mostrada en la tabla anterior usando el paquete h2o con la accuracy de los mismos modelos construidos en el apartado V.1. En este caso los modelos son comparables al estar evaluados utilizando el mismo periodo de prueba.

Alisado	acc1m.h20	acc1m	acc2m.h20	acc2m	acc3m.h20	acc3m
Alisado exponencial	49.90	48.65	85.47	43.60	57.37	66.21
EMA30	50.94	41.16	42.95	45.12	53.74	57.82
EMA60	78.79	64.66	60.09	77.66	75.96	75.51
EMA90	74.43	84.41	75.49	65.08	65.99	69.84

Tabla V.31: Comparativa de rendimiento obtenido con Random Forest con parámetros optimizados manualmente y los modelos construidos con ML automático H2O

Como se puede apreciar en la tabla anterior, los resultados de accuracy calculada con un umbral de 0.5 son en general mejores utilizando los modelos construidos automáticamente usando el paquete H2O. Sin embargo sí que existen ciertas combinaciones para las cuales los modelos Random Forest construidos, cuyos parámetros se optimizan manualmente. No se aprecia ningún patrón claro que pueda distinguir para que combinaciones es mejor qué modelo. Sin embargo, lo que sí se aprecia es la diferencia que existe en cuanto a la carga de trabajo que supone la construcción de ambos tipos de modelado. Mientras que la construcción del modelo Random Forest requiere de un procedimiento de optimización de parámetros manual, en el sentido que hay que escribir el proceso que elabore el *grid-search*, el proceso de modelado utilizando el ML automático con el paquete H2O no requiere más que unas líneas de código para elaborar el mismo tipo de optimización sobre muestra de validación, previo al cálculo de las métricas sobre la muestra de entrenamiento. Habiendo observado que los resultados son el general mejores, si más no parecidos, utilizando ambos tipos de modelado, queda claro que el *trade-off* entre carga de trabajo y tiempo versus los resultados lo gana el paquete H2O. Cabe destacar además que el hecho de construir un modelo Gradient Boosting Machine y optimizarlo "a mano" es mucho más laborioso que el de construir un Random Forest en cuanto a complejidad del código, por lo que la opción de modelado con autoML de H2O aparece muy atractiva a nivel usuario para según que tipo de procesos de modelado. Por el contrario aparece como parte negativa el carácter *black-box* que tienen este tipo de sistemas automáticos de machine learning, teniendo que confíar el usuario en los cálculos y resultados que se ofrecen.

V.4 Predicción del precio: LSTM-RNN

En el presente apartado se procede a construir una red neuronal recurrente tipo LSTM para predecir los precios de cierre de la empresa Coca Cola Co.. En hecho de escoger esta empresa está justificado por distintas razones. En primer lugar el hecho de que solo sea una empresa y no las 4 utilizadas como ejemplo en el apartado V.1 se debe a que la capacidad computacional de la que se dispone es limitada. El proceso de entrenamiento de un modelo de estas características requiere de una elevada capacidad computacional para poder realizarse en un periodo de tiempo relativamente razonable. En segundo lugar el hecho de que sea Coca Cola y no una de las otras 3 empresas responde a lo observado en el apartado V.1.1. Esta empresa es la que ofrece una opción relativamente balanceada entre rentabilidad y riesgo. En tercer lugar la tendencia creciente presente en parte de la serie parece ser relativamente fuerte y la LSTM pueda probablemente captarlo.

Para emprezar se elabora el plan de entrenamiento del modelo lstm. Con el procedimiento que se detalla a continuación permite obtener una evaluación del rendimiento de este tipo de modelos en distintos trozos de la serie temporal. En este caso se contruyen 6 submuestras que conforman el plan de entrenamiento, cada una con 996 días en la muestra de entrenamiento y 83 en la muestra de test. El por qué de esta configuració se explica posteriormente en la parte de ajuste de los parámetros. El tercer parámetro que define la partición del plan de entrenamiento es el que controla la separación entre las ventanas móviles. En este caso se define en 650 días. Esto significa que la distancia entre el inicio de las series temporales en cada partición está separado por 650 días. En definitiva el desarrollo que tiene este apartado es el siguiente: por simplicidad, se procede a entrenar una LSTM para cada partición creada, utilizando la misma combinación de hyperparámetros para este modelo. En este apartado no se elabora una optimización exhaustiva de los hyperparámetros de este modelo, que es de hecho donde está la gran complejidad de los algoritmos de machine learning, sino que se define una combinación de los mismos, que se podría considerar como referencia para futuras optimizaciones. Esto es así ya que en el momento de la elaboración de este trabajo no se dispone de la capacidad computacional adecuada para poder elaborar una tarea de semejante magnitud, al ser el periodo de tiempo considerado relativamente elevado (18 años) y al ser el número de combinaciones de hyper parámetros que hay que probar muy elevada. El hecho de entrenar disintos modelos en distintos periodos de tiempo ofrece la posibilidad de analizar el rendimiento de estos modelos con distintas predicciones a lo largo del tiempo. El hecho de reducir el tamaño de la muestra de entrenamiento para cada LSTM hace que el proceso de entrenamiento sea mucho más rápido, con la contrapartida de que los modelos que se crean no están ofreciendo todo su potencial.

En la gráfica que se muestra a continuación se pueden ver representadas las distintas particiones de la serie temporal de los precios de cierre de Coca-Cola sobre los cuales se va a entrenar una LSTM-RNN con el fin de testearla en los 83 días de muestra de prueba, graficados en rojo.

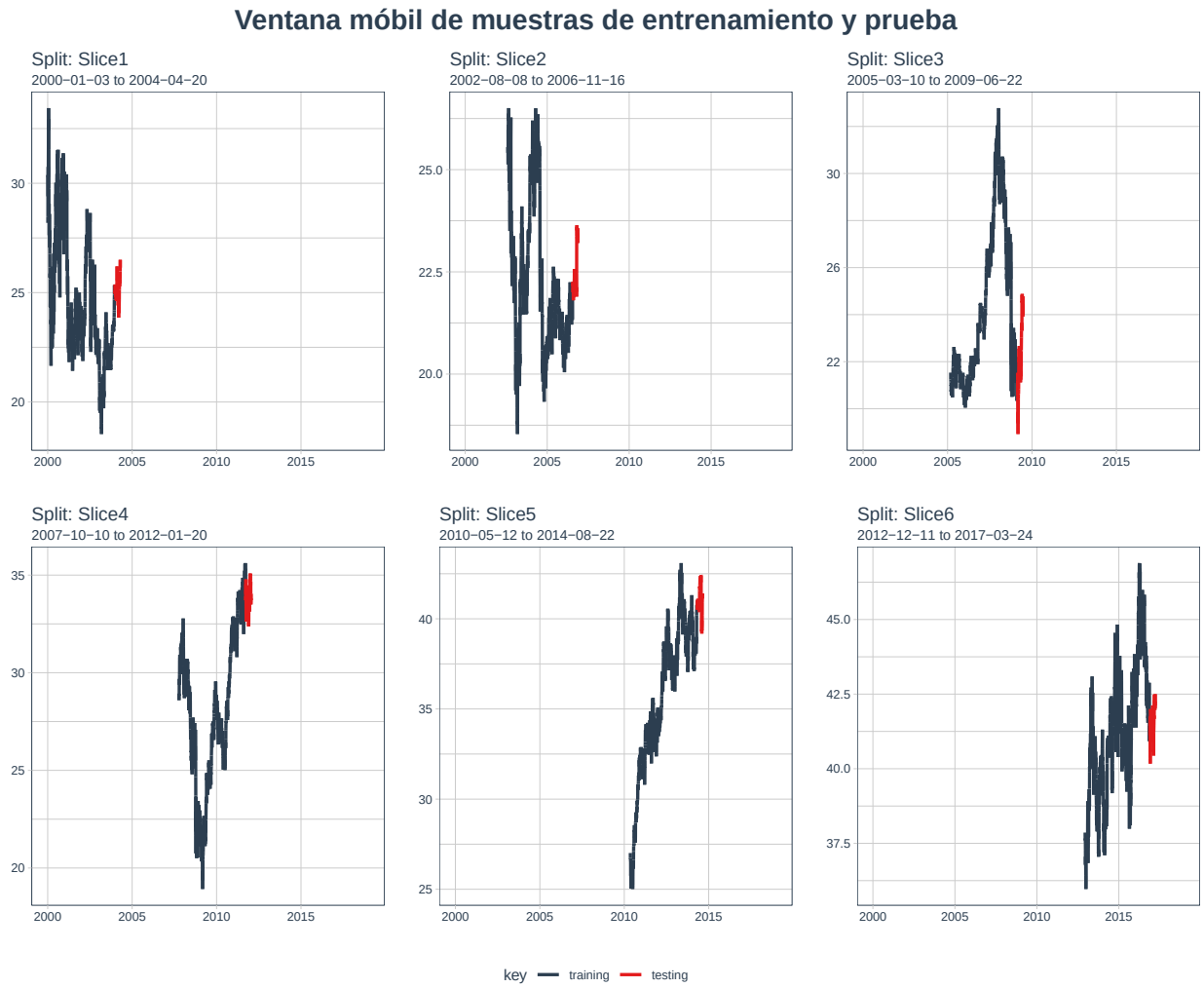


Figura V.12: Plan de entrenamiento de la LSTM para los precios de cierre. Fuente: elaboración propia

También se muestra el gráfico ampliado.

Figura V.13: Plan de entrenamiento de la LSTM para los precios de cierre. Ampliado. Fuente: elaboración propia

Una vez se tienen listas las particiones se procede a la creación de los modelos LSTM sobre cada una de ellas. Cabe decir que la implementación que se hace en este trabajo de los modelos LSTM se soporta en el entorno Keras. En este caso se ha decidido utilizar la implementación del entorno en keras en R, utilizando el paquete `keras`, en vez de hacerlo en Python para mantener la integridad de todo el trabajo en cuanto al software / lenguaje que se utiliza para escribir y realizar los apartados de la presente tesis. Esta decisión tiene implicaciones ya que el paquete `keras`, aunque trabaja "por debajo" con el entorno de programación TensorFlow, no permite tener un control a tan bajo nivel de todos los hyper parámetros que se pueden optimizar en un modelo de estas características.

Como se ha apuntado previamente, en el presente trabajo se elaboran los modelos LSTM con una cierta combinación de parámetros, debido a la alta capacidad computacional que requiere el hecho de hacer el proceso de *rolling origin cross validation* que este tipo de modelos

requieren para optimizar sus hyper parámteros. Los pasos en el proceso de creación de estos modelos se detallan a continuación:

i) En primer lugar se separan los datos de cada partición en sendas muestras de entrenamiento y prueba, teniendo la primera 996 observaciones, es decir, días con valor en el precio de cierre de la empresa Coca Cola, mientras que la muestra de prueba consta de 83 días. Esta separación entre muestras de entrenamiento y prueba responde al hecho que, a causa de la forma requerida del tensor de entrada en el modelo, el número de observaciones en la muestra de entrenamiento tiene que ser divisible entre el número de observaciones de la muestra de prueba, de manera que el resultado sea un número entero.

ii) En segundo lugar se preprocesan los datos. En este caso se aplica en primer lugar la raíz cuadrada a los datos. Este proceso ayuda a reducir la varianza y eliminar los outliers. En segundo lugar se estandarizan los datos, restándoles la media y dividiéndolos por su desviación típica. Este proceso también se conoce como escalar y centrar los datos.

iii) En tercer lugar se transforma la forma de los datos. Éstos necesitan ser transformados a forma de tensor, ya que es la forma requerida por este tipo de modelos. El concepto de tensor se puede pensar como una entidad algebraica que generaliza los conceptos de escalar, vector y matriz. Se podría entender como un vector de matrices, en este caso. En el presente trabajo la forma del tensor se detalla posteriormente en este apartado. Se construyen pues los tensores input y output de las muestras de entrenamiento y prueba.

iv) En cuarto lugar se definen los hyper parámetros y se construye la función que se aplicará a cada partición para entrenar la LSTM. En cuanto a los parámetros, se presenta una restricción añadida a la descrita en el apartado i). El cociente entre el número de observaciones de la muestra de entrenamiento y el parámetro llamado *batch size*, o tamaño del grupo, tiene que resultar también en un número entero. Esta regla también se aplica al número de observaciones en la muestra de prueba. El hyper parámetro *batch size* controla el número de muestras, u observaciones, sobre las que se trabaja antes de actualizar los parámetros internos de la LSTM o pesos da la red. De hecho, es el número de observaciones que se utilizan para hacer las predicciones y así poder calcular el error que permite optimizar los pesos de la red. En el presente trabajo este parámetro se fija en 1 de manera que se está utilizando sólo 1 observación a la vez para actualizar los parámetros internos del modelo. Esta configuración a la hora de utilizar el algoritmo de optimización del descenso del gradiente (Gradient Descent) para optimizar los pesos de la red se llama Stochastic Gradient Descent. Se sabe que fijar el *batch size* en un valor pequeño aportan ruido y añaden un efecto regularizador que permite generalizar mejor. (Dominic Masters, 2018) presentan unos resultados en los que confirman que utilizando un valor pequeño para el parámetro *batch size* se

consigue obtener una estabilidad en el entrenamiento y una mejor generalización, dado una capacidad computacional, a través de un amplio abanico de experimentos.Otro de los parámetros que se fijan durante el entrenamiento de las LSTM es el número de épocas. Este hyper parámetro define el número de veces que el algoritmo va a aprender de todo el conjunto de datos. Es decir, terminar una época significa que cada muestra u observación en el conjunto de entrenamiento ha tenido la oportunidad de actualizar los parámetros internos del modelo o pesos de la red. Las épocas se fijan en 100 para limitar el tiempo de entrenamiento. También se fijan a 100 las unidades, o redes neuronales, internas de la LSTM. Este parámetro también se conoce como neuronas y controla la capacidad que tiene la LSTM de aprender. A más neuronas, más capacidad tiene la LSTM de aprender. Seguidamente se fija para el ajuste de la LSTM es el llamado *time steps*. Este parámetro corresponde a la segunda dimensión del tensor de entrada. En este caso los tensores son de la forma $Input = [996, 1, 1]$, $output = [83, 1]$. Este hyper parámetro controla el número de observaciones en el pasado sobre las de las que la LSTM aprende. Como se ha detallado en el apartado IV.1 la LSTM es un tipo de red neuronal que permite aprender de periodos alejados en el tiempo, ya que es capaz de determinar la cantidad de información del pasado que hay que retener. Por añadidura, se fija el parámetro *unit_forget_bias* a 1. Este hyper parámetro añade 1 al seso de la *puerta de olvido* al inicializarse. Además añade una referencia a Jozefowicz et. al. apuntando a que lo recomienda (Allaire & Chollet, 2019). Finalmente cabe decir que la LSTM que se construye es de tipo *STATEFULL*, tal y como se describe en el apartado IV.1.

v) En quinto lugar se entrena de manera iterativa sobre todas las épocas la LSTM.

vi) Con el modelo entrenado, se calcula la predicción sobre la muestra de entrenamiento y se construye un `data frame` con los resultados reales y los predichos.

Este proceso se aplica de manera iterativa sobre todas las particiones y se grafican los resultados obtenidos sobre las muestras de prueba. En rojo se pueden ver las predicciones, mientras que en negro se dibujan los valores reales del precio de cierre.

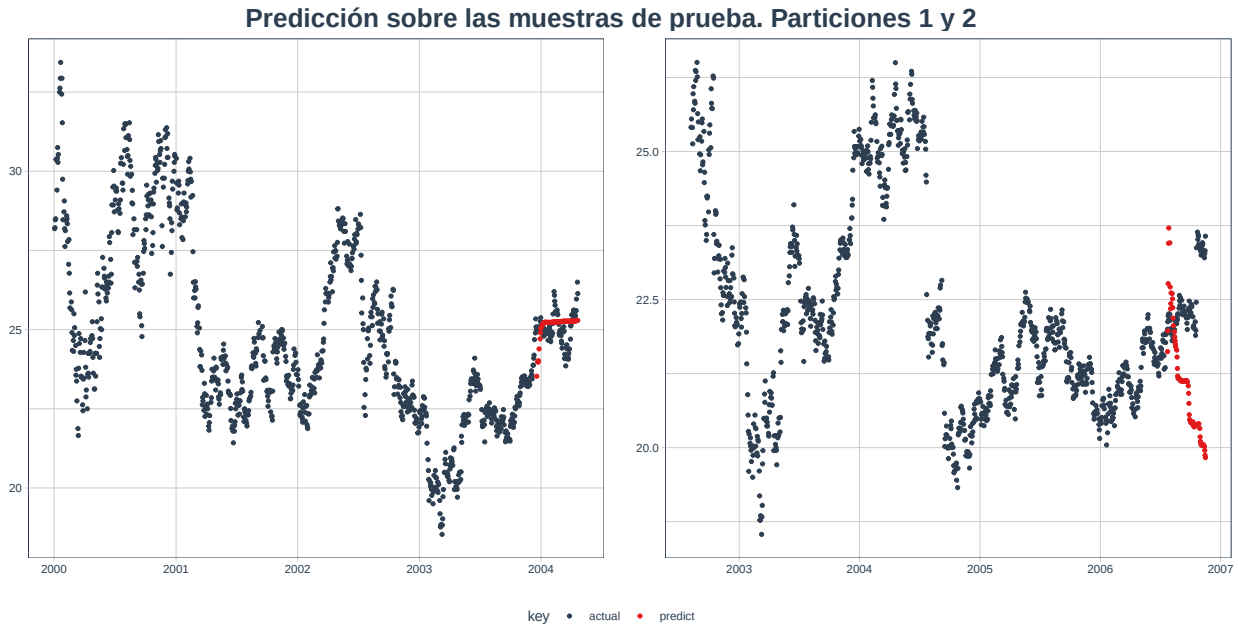


Figura V.14: Resultados sobre muestra de prueba de la LSTM sobre los precios de cierre en particiones 1 y 2. Fuente: elaboración propia

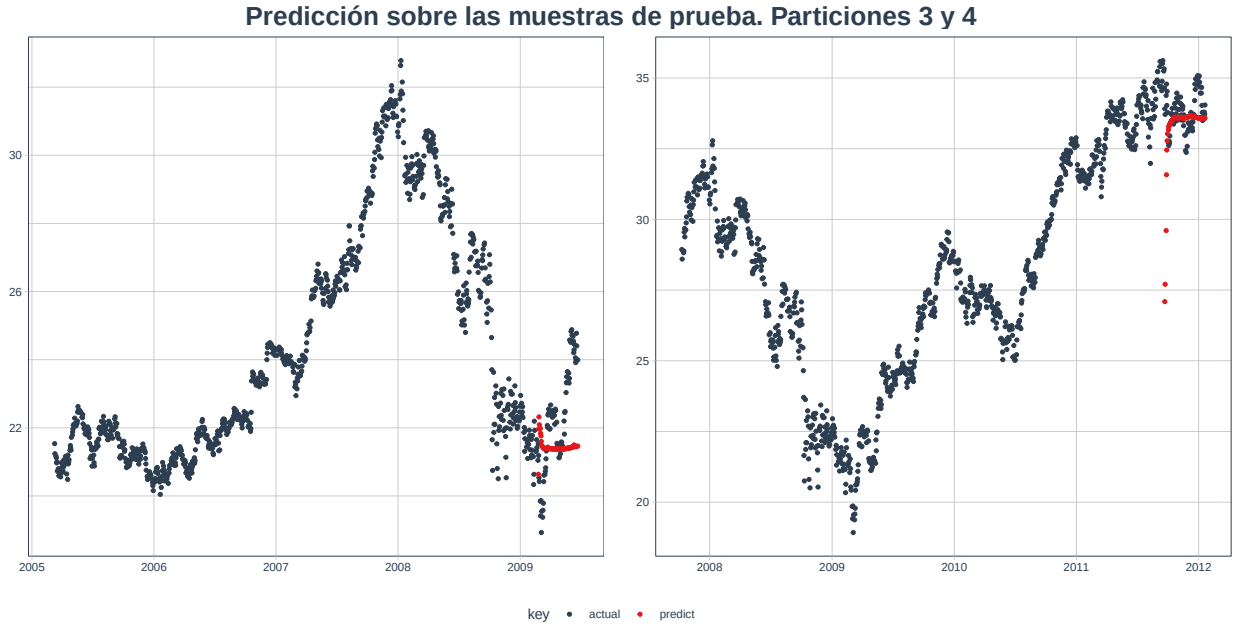


Figura V.15: Resultados sobre muestra de prueba de la LSTM sobre los precios de cierre en particiones 3 y 4. Fuente: elaboración propia

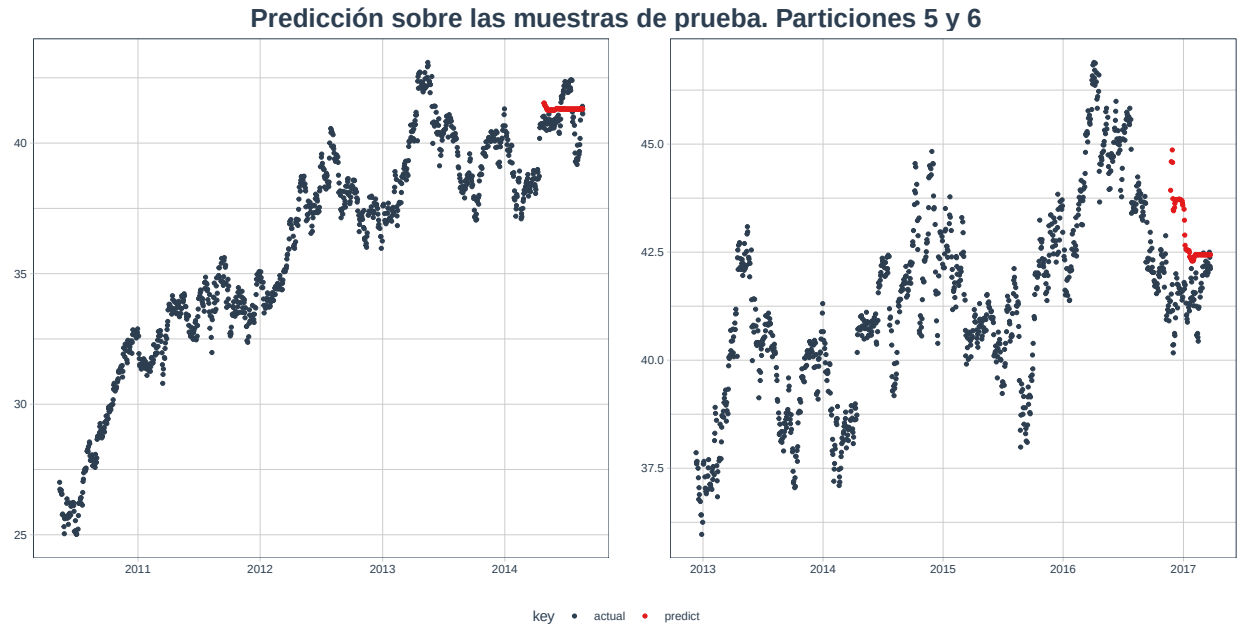


Figura V.16: Resultados sobre muestra de prueba de la LSTM sobre los precios de cierre en particiones 5 y 6. Fuente: elaboración propia

Seguidamente se presenta la tabla que muestra el cálculo de las distintas métricas de rendimiento para las distintas particiones:

```
##     Split     MAPE    MAPE2
## 1 split 1 1.992372 1.980983
## 2 split 2 7.150241 7.244746
## 3 split 3 6.203487 6.340186
## 4 split 4 2.233511 2.249743
## 5 split 5 1.820718 1.807583
## 6 split 6 3.279530 3.258063
```

Como se puede observar en base al cálculo de los dos tipos de MAPE, los resultados parecen bastante buenos. En la peor de las particiones el modelo es capaz de generar un error percentual absoluto medio de alrededor del 7%. El MAPE2, modificado, muestra unos resultados que son ligeramente mejores que la métrica del MAPE original, pero aun así en general se observan buenos resultados. Aun habiendo entrenado la LSTM sin optimizar los hyper parámetros y en el caso más sencillo en el que los *time steps* están fijados a 1 parece que se obtiene un resultado relativamente bueno en términos de MAPE. Sin embargo, una observación más detallada de las gráficas anteriores indica que, aunque los valos predichos están en un nivel razonable, no son capaces de captar los patrones que aparecen rápidamente, como podría ser un crecimiento. Cabe destacar la predicción elaborada en la partición 2. Los valores predichos tienen una tendencia decreciente, mientras que los precios crecían. Esto se debe al gran decrecimiento que presentan los datos en esta partición, en su primera mitad. El

modelo LSTM entrenado con esta partición aprende este partón y es el que intenta replicar al hacer las predicciones.

Para facilitar la comparabilidad de las dos métricas se presenta la siguiente visualización gráfica:

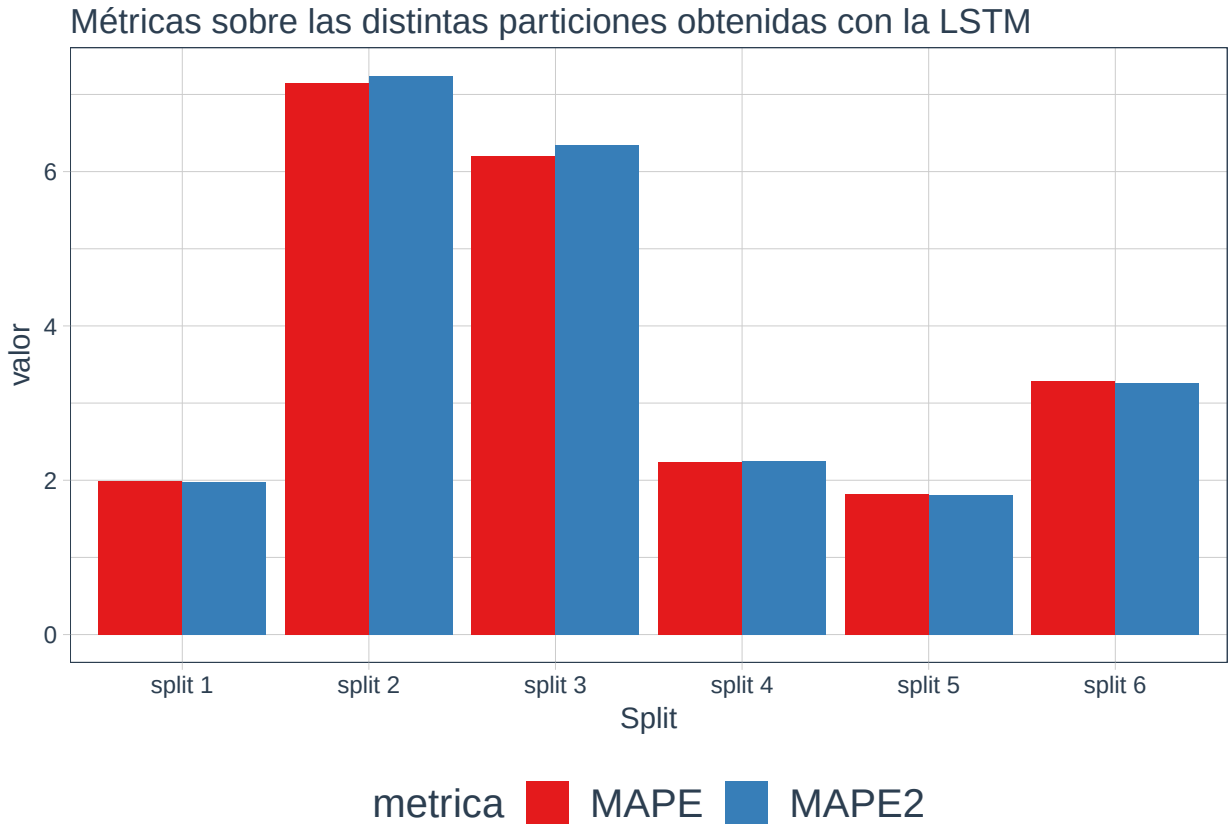


Figura V.17: Métricas de rendimiento de las predicciones elaboradas con LSTM sobre las distintas particiones. Precio de cierre. Fuente: elaboración propia

Gracias a la figura anterior se puede ver fácilmente el punto que se apuntaba previamente. La segunda métrica parece ofrecer ligeramente unos valores inferiores pero en general los resultados obtenidos en términos de MAPE son bastante postivios. Esto induce a pensar que, disponiendo de una mayor capacidad computacional, este modelo se podría mejorar más. La propuesta sería la de utilizar el paquete `tfruns` para elaborar un *grid search* sobre todas las combinaciones de hyper parámetros disponibles. Esta técnica consiste en probar todas las combinaciones de parámetros con tal de escoger la combinación que ofrece un mejor resultado sobre la muestra de validación. En este caso se hubiera propuesto una técnica de cross-validación adaptada a las series temporales llamada *rolling origin cross-validation*, que permite mantener la estructura temporal de la seria en el momento de hacer la optimización

vía *grid search* en muestra de validación.

Seguidamente también se muestran dos métricas alternativas que permiten evaluar la capacidad predictiva de los precios de la LSTM. Éstas métricas son el Error Absoluto Medio (MAE) y el Cuadrado de la Media de los errores al cuadrado (RMSE)

```
##   particiones       MAE      RMSE
## 1     split 1 0.4961335 0.6122746
## 2     split 5 0.7410633 0.8884039
## 3     split 4 0.7601204 1.3616478
## 4     split 6 1.3534583 1.6429215
## 5     split 3 1.4095112 1.7496488
## 6     split 2 1.6291732 1.9298185
```

Como se puede apreciar en los resultados anteriores, ambas métricas también muestran que en general se obtienen unos resultados relativamente buenos. En ningún caso el RMSE pasa de 2 puntos en las particiones, siendo la 1ª y la 5ª partición las que obtienen mejores resultados, siendo su RMSE menor a 1. Sin embargo, las particiones que obtienen peores resultados son las particiones 2 y 3.

V.5 Predicción de la rentabilidad: LSTM-RNN

Una vez aplicada la LSTM sobre los precios de cierre directamente, se explora la posibilidad de construir el mismo tipo de modelos pero utilizando el logaritmo de las rentabilidades como serie temporal. Los *log returns* pueden verse desde una perspectiva económica como la ganancia percentual entre dos precios, puediendo ser su valor positivo o negativo. Desde un punto de vista estadístico lo que busca esta transformación es que la serie temporal pase a ser estacionaria de segundo orden al estar sus valores centrados en 0 y tener (teóricamente) una varianza constante a lo largo del periodo considerado. Sin embargo, este último punto no es del todo cierto en el contexto del sector financiero ya que, al ser las series temporales consideradas muy volátiles, no se consigue estabilizar totalmente la varianza aplicando el logaritmo. Ésto se puede observar en los gráficos que se muestran a continuación en los distintos picos que presenta la rentabilidad, cosa que indica que la varianza no está del todo estabilizada. Esta rentabilidad es la misma que se calcula en la sección V.1 al hacer la descriptiva de las series temporales consideradas.

La estrategia de entrenamiento de los modelos LSTM sobre la rentabilidad es la misma que en el caso anterior sobre el precio. Se elaboran 6 particiones y sus respectivas particiones sobre muestras de entrenamiento y prueba. Los gráficos de las particiones se presentan a continuación.

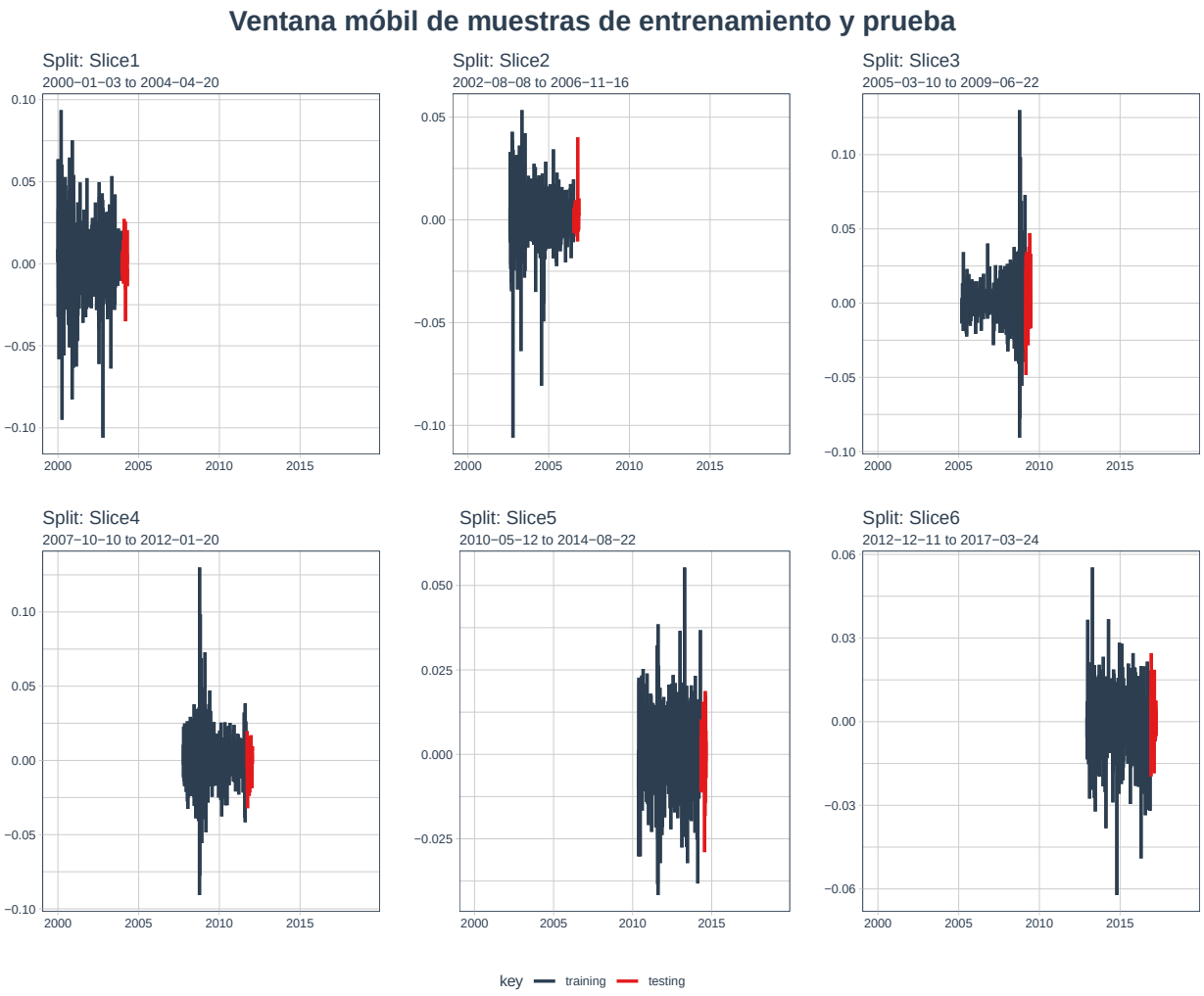


Figura V.18: Plan de entrenamiento de la LSTM para la rentabilidad de Coca Cola. Fuente: elaboración propia

Para facilitar la visualización se presentan también las gráficas ampliadas:

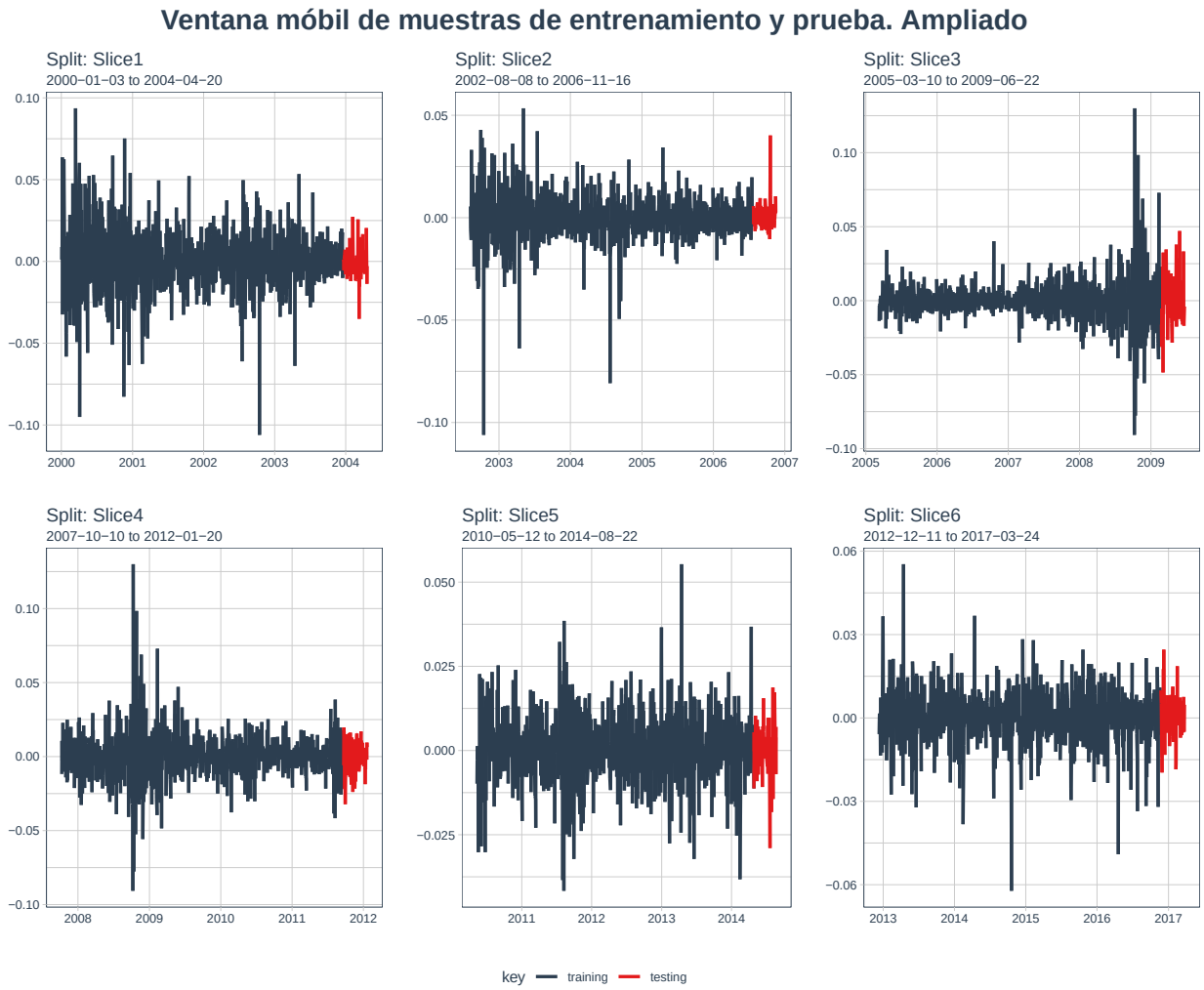


Figura V.19: Plan de entrenamiento de la LSTM para los precios de cierre. Ampliado. Fuente: elaboración propia

La configuración de los hyper parámetros ha sido la misma que en la sección V.4 donde se aplican los modelos LSTM sobre el precio. De nuevo esto responde a un intento de simplificar la complejidad computacional que presenta una optimización y entrenamiento intensivos de estos modelos. A continuación se presentan graficados los valores predichos en rojo encima de los valores reales en gris.

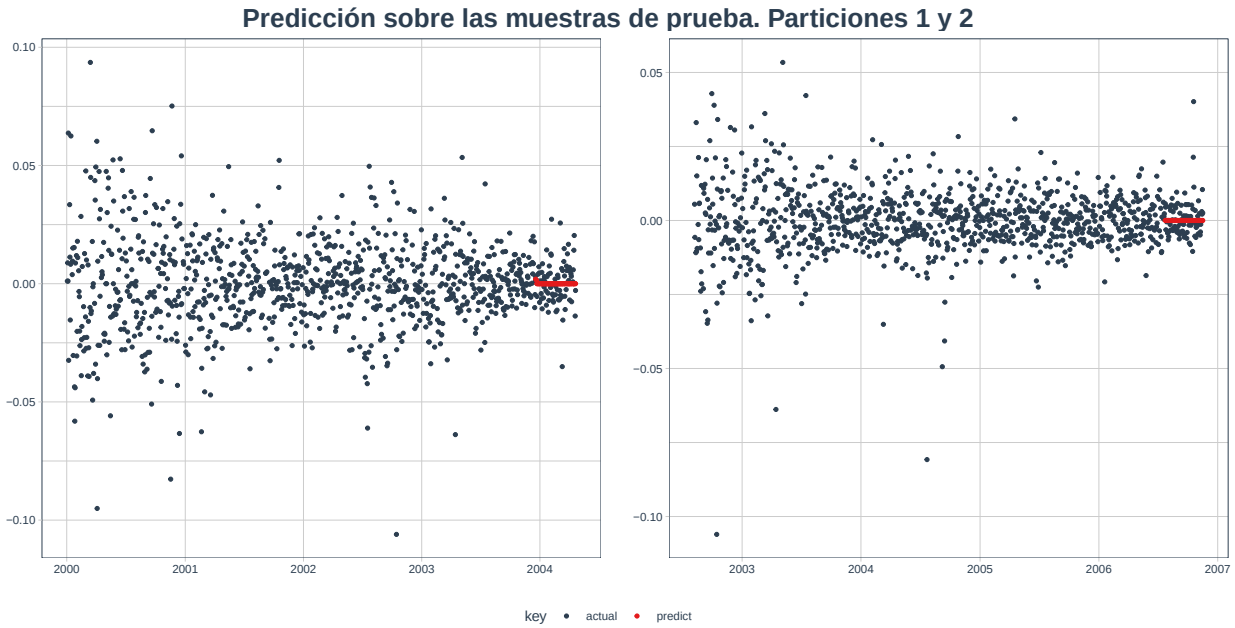


Figura V.20: Resultados sobre muestra de prueba de la LSTM sobre la rentabilidad en particiones 1 y 2. Fuente: elaboración propia

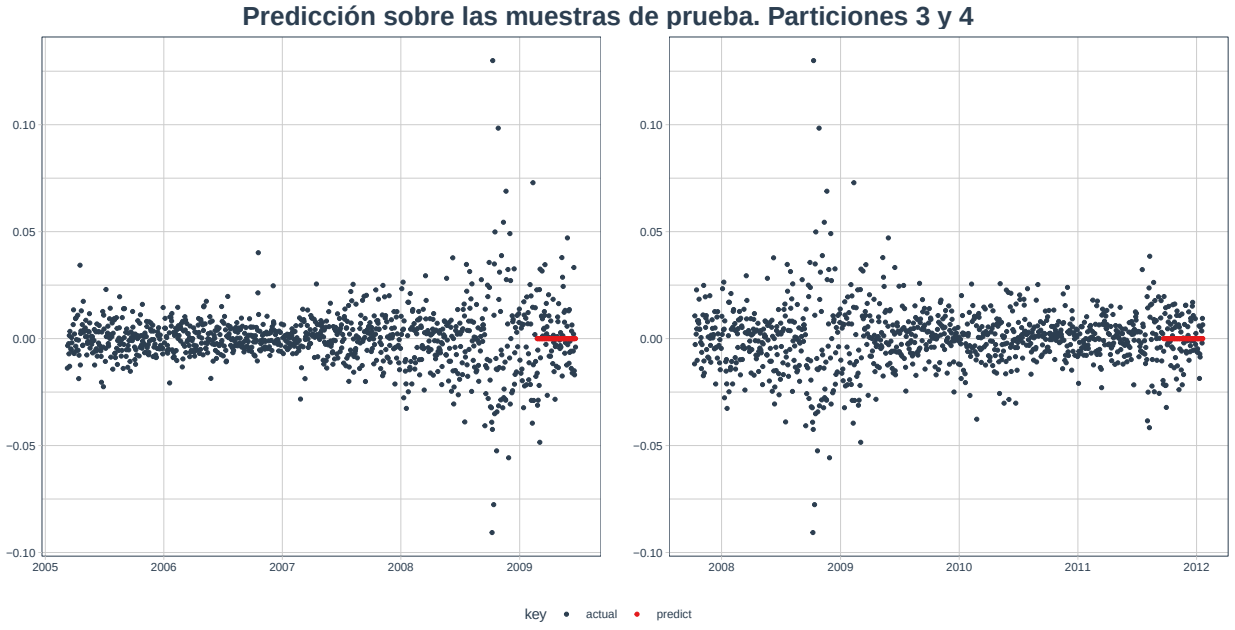


Figura V.21: Resultados sobre muestra de prueba de la LSTM sobre la rentabilidad en particiones 3 y 4. Fuente: elaboración propia

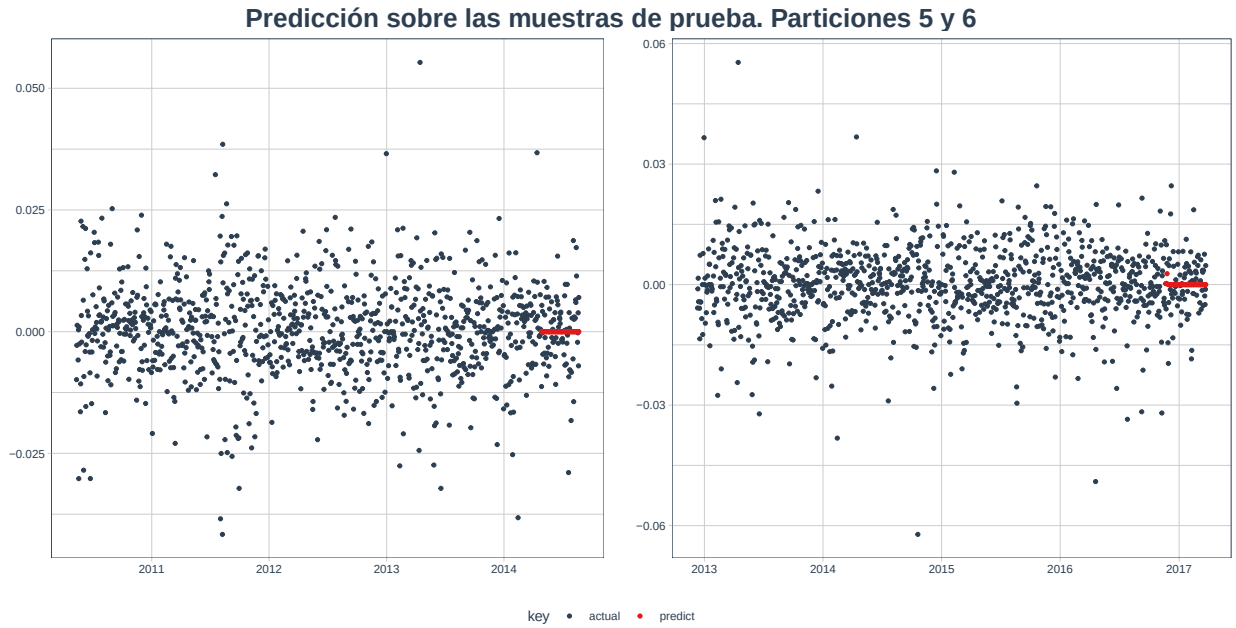


Figura V.22: Resultados sobre muestra de prueba de la LSTM sobre la rentabilidad en particiones 5 y 6. Fuente: elaboración propia

Para evaluar la capacidad predictiva de la LSTM que se acaba de construir sobre la rentabilidad de la empresa Coca Cola se necesita modificar ligeramente la métrica de MAPE presentada en M.4 ya que, al ser la rentabilidad un estadístico que puede presentar valores tanto negativos como positivos, hay que hacer el sumatorio del valor absoluto del cociente, en vez de simplemente el valor absoluto del nominador. Esto se debe al hecho de que los valores actuales, o reales, pueden ser negativos. La nueva métrica de MAPE propuesta para el caso de la rentabilidad es la siguiente:

$$MAPE_{bis} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Actual_i \ - \ Predicted_i}{Actual_i}\right| \qquad (M.4bis)$$

$$MAPE_{2bis} = \frac{\sum_{i=1}^{n}|Actual_i \ - \ Predicted_i|}{\sum_{i=1}^{n}|Actual_i|} \qquad (M.5bis)$$

```
##     Split      MAPE     MAPE2
## 1 split 1 105.68075 100.07356
## 2 split 2 100.00722  99.99388
## 3 split 3  99.92937  99.99064
## 4 split 4 100.00339 100.00138
```

```
## 5 split 5 100.00859 100.01386
## 6 split 6  99.35183  99.35015
```

A continuación se presentan finalmente, siguiendo el formato presentado en la sección V.4, las métricas de rendimiento sobre la predicción que se acaba de generar para la rentabilidad de la empresa Coca-Cola en las 6 particiones. En este caso las métricas que se presentan finalmente para evaluar el rendimiento de las predicciones hechas con la LSTM sobre la muestra de prueba de las rentabilidades son el Error Medio Absoluto (MAE) y el RMSE.

```
##   particiones         MAE        RMSE
## 1     split 2 0.004537274 0.006794484
## 2     split 6 0.005404819 0.007326430
## 3     split 5 0.005718338 0.007525276
## 4     split 1 0.007031099 0.009317196
## 5     split 4 0.008861489 0.010917244
## 6     split 3 0.012347274 0.016499228
```

Los resultados observados anteriormente llevan a pensar que los modelos LSTM construidos con la rentabilidad sobre las distintas particiones no son para nada unos modelos que ofrezcan buenos resultados. Esto se puede ver a partir del cáluclo del MAPE, en contrapartida a las métricas MAE y RMSE. El MAPE de todas las particiones está sobre el 100%. Esto significa que en porcentage, se hace un 100% de error respecto a los valores actuales. Cabe destacar que para considerar un modelo predictivo como de alto rendimiento, el MAPE debería ser inferior al 5%. En este caso, los valores de la rentabilidad son muy pequeños, del orden de 3 decimales, de manera que una diferencia de 1 decimal implica un porcentage de error muy elevado. Por ejemplo, si el valor actual de la rentabilidad es del 0.00019 y el modelo predice un valor de 0.014 para el mismo periodo, uno se encuentra con un error que es 6 veces mayor que el valor real. Es por esto que el cálculo del MAPE ofrece valores tan elevados (alrededor del 100%). Este hecho se remarca ya que el cálculo del RMSE podría engañar al lector. Aunque los cuadrados de los errores sean muy pequeños, esto no significa que los modelos construidos sobre la rentabilidad tengan un buen rendimiento (tal como indica el MAPE). Estos modelos presentan un RMSE tan pequeño a causa del orden de los valores de la rentabilidad. Es decir, al ser la rentabilidad un valor tan pequeño, del orden de 3 o 4 decimales, los cuadrados de los errores que se obtienen son, a su vez, muy pequeños. Esto podría parecer la indicación de que los modelos predictivos sobre la rentabilidad funcionan de una manera sorprendentemente buena cuando, en realidad, no lo están haciendo, como muestra el MAPE. En resumen: parece que los modelos LSTM construidos sobre la rentabilidad no funcionan, ni de cerca, tan bien que los modelos sobre el precio de cierre. El lector debe recordar que los modelos LSTM se han construido con una configuración determinada al disponer de una capacidad computacional limitada. Por ese motivo el rendimiento de la LSTM sobre la rentabilidad se cree que puede mejorar con una configuración optimizada de los hyper parámetros.

CAPÍTULO VI

CONCLUSIONES

Una vez terminadas las explicaciones de la parte práctica, habiendo desarrollado todos los experimentos previamente, se procede a la redacción de las conclusiones de la tesina. En primer lugar cabe destacar que la exposición de las conclusiones se elabora en dos partes, siguiendo la línea general del trabajo. Por un lado las conclusiones que se extraen del análisis elaborado en el apartado III. Por el otro las conclusiones extraídas de todos los experimentos que se elaboran en el apartado V.4. La línea que se pretende seguir en la redacción de las conclusiones es la siguiente. Para poder dar una visión general de todo el trabajo, y de sus implicaciones, todas las conclusiones son extraídas a partir de los objetivos fijados para este trabajo. En base a lo que se pretendía analizar, se extraen las conclusiones en base a lo que se ha elaborado.

Por lo que respecta al análisis de la situación actual de la aplicación de la inteligencia artificial en el sector de las finanzas, se han cubierto largamente las distintas aplicaciones que ésta tiene hoy en día. Este análisis tenía por objetivo el obtener una visión global de las distintas aplicaciones actuales con tal de poder analizar las consecuencias que esta aplicación ha tenido. Las consecuencias transversales que se han podido observar son las siguientes. En todas las aplicaciones se aprecia la misma tónica y es que el aumento que está experimentando en el siglo XXI la generación de datos, en cuanto a volúmen, variedad y velocidad de generación permite la situación perfecta para la proliferación de aplicaciones de IA que utilizan estos grándes volúmenes de datos como fuente de información. Esto unido al aumento de la capacidad computacional que conlleva la evolución tecnológica ha permitido desarrollar aplicaciones nunca antes vistas, que procesan este gran volumen de información, tradicional y nuevos tipos de datos, de una manera muy rápida, y ha motivado la creación de muchas empresas nuevas que se encargan de desarrollar nuevas soluciones soportadas en sistemas de IA.

Desde una perspectiva de impacto positivo, se ha observado que, de una manera transversal en las distintas aplicaciones analizadas, una de las principales consecuencias de la existencia hoy en día de sistemas de IA que reproducen más rápidamente las tareas que previamente hacían los seres humanos es la reducción en los costes asociados. Esto puede permitir en definitiva una mejora en la eficiencia general del sector financiero. Otra de las consecuencias transversales que se observa es la deriva que está tomando el sector financiero, por un lado,hacia la aparición de nuevos agentes en el mismo que remueven a los bancos de su

posición oligopolística, y por otro, hacia la oferta de servicios muchos más personalizados y adaptados a las necesidades específicas de cada cliente. Paralelamente, se ha podido apreciar el potencial que tiene la IA para contribuir al crecimiento económico, visto desde una perspectiva macroeconómica.

Sin embargo, desde una perspectiva de impacto negativo se ha observado una característica compartida por las distintas aplicaciones analizadas. El caracter *black-box* que suelen presentar este tipo de modelos de *machine learning*, en el sentido de que las decisiones internas que toman no son comprensibles de una manera sencilla, hace que todas los sistemas derivados de la utilización de técnicas de IA y *machine learning* tengan un carácter poco transparente. El problema de la falta de transparencia a causa de la utilización de estos modelos, que deriva del carácter poco comprensible de los mismos, se presentará en un futuro como el mayor de los problemas que van a tener que enfrentar las compañías, tanto privadas como entidades reguladoras, que están utilizando o se plantean utilizar sistemas de inteligencia artificial para complementar, mejorar o sustituir sus actividades. En este sentido se aprecia también un creciente interés por parte del sector público o regulatorio por la aplicación de este tipo de sistemas, siguiendo como es habitual el sector privado, que puede hacer cambiar en un futuro cómo se llevan a cabo tareas tales como la detección de fraude. A raíz de esta falta de transparencia presente en los actuales sistemas de IA en finanzas, se ha podido apreciar una deriva creciente, tanto por parte de los consumidores de servicios financieros basados en IA como de las mismas compañías, hacia sistemas de IA que sean fácilmente explicables, comprensibles y mucho más transparentes.

Otra de las potenciales consecuencias negativas que se observa se deriva de la gran implantación que están teniendo este tipo de servicios dentro del sector más clásico de los servicios financieros. La presencia cada vez mayor de sistemas de IA totalmente automatizados genera un potencial riesgo ya que puede contribuir a un efecto en cadena de carácter extremadamente rápido si se produce en *crash* en el sector concreto en el que operen. Si muchos agentes confían en este tipo de aplicaciones o modelos para llevar a cabo su actividad, los efectos desencadenantes que esto puede tener frente a una situación de *shock* son extremadamente rápidos e incontrolables. Esto unido al hecho de que los sistemas de IA en finanzas son mayoritariamente poco transparentes puede dificultar en análisis a tiempo real de los sistemas que están propagando un determinado *crash*, e incluso puede provocar que la comprensión de las causas de un determinado *crash* pueda extenderse durante semanas.

En general se observa que la situación actual de la inteligencia artificial en el sector financiero es muy incierta. Actualmente estamos viviendo una primera etapa de aplicación masiva, donde muchos agentes empiezan a hacer el cambio hacia sistemas basados íntegra o parcialmente en modelos de IA y *machine learning*. La previsión que se puede elaborar en base al análisis elaborado en esta tesis es que la etapa de implantación aun puede durar entre 5 y 15 años, de manera que la predicción a largo plazo es una entrada en una etapa de madurez, donde la IA se usará en la gran mayoría de sub-sectores dentro del sector financiero

y habrá permitido mejorar, automatizar, crear y destruir muchos de los empleos dentro de este sector que existen hoy en día.

Por lo que respecta a las conclusiones que se extraen de los experimentos realizados, se proceden a detallar las conclusiones extraídas a partir de los distintos objetivos definidos. En primer lugar el trabajo tenía por objetivo el definir un modelo de predicción de la dirección de movimiento de un precio sobre 4 empresas concretas, utilizando un modelo Random Forest con los parámetros optimizados. Inicialmente se ha constatado que cada empresa presenta una serie temporal de precios de cierre única, con patrones totalmente distintos y con una alta volatilidad. Esto ha sido útil para poder tomar conciencia de la dificultad de obtener rendimientos elevados de los modelos predictivos aplicados a este tipo de datos. También se observa que el hecho de alisar las series temporales reduce el ruido presente en los datos y ayuda a predecir la dirección de movimiento del precio. Por lo que respecta a la optimización de parámetros realizada se ha podido ver que optimizar los parámetros sobre una muestra de validación que consiste en un sólo año de datos, que además puede presentar un patrón distinto a la muestra de prueba, provoca que los modelos generados con dichos parámetros optimizados no tengan un buen rendimiento sobre muestra de prueba. Esto se debe al hecho de que las series temporales consideradas presentan numerosos cambios estructurales, que pueden tener lugar entre la muestra de validación y la de prueba. En general, una vez realizado el análisis se concluye que *los indicadores técnicos considerados en el presente trabajo son útiles cuando se trata de predecir la dirección de movimiento del precio de cierre.* Aunque el rendimiento obtenido de estos modelos no es el mismo para todas las empresas, y en algunos casos no es útil para una situación real de inversión, en general se observa que el hecho de añadir al modelo los indicadores técnicos considerados mejora la predicción ingenua de 50% de probabilidad asignada a cada una de los dos niveles de la variable respuesta (sube o baja).

En segundo lugar este trabajo tenía como objetivo el aplicar este modelo conceptual de predicción de la dirección de movimiento de una manera masiva, generalizada y utilizando computacion en paralelo, sobre un gran conjunto de empresas. El interés de este objetivo es el de poder aplicar este análisis en la realidad y así poder escoger la empresa en la cual invertir en base al mejor rendimiento obtenido sobre muestra de prueba. Pese a la limitada capacidad computacional de la que se ha dispuesto durante la realización de la presente tesis se ha podido realizar el análisis masivo paralelizando la computación para ganar velocidad de cálculo. Gracias a esta aplicación general se ha podido confirmar lo observado en el experimento con 4 empresas concretas. Los indicadores técnicos aportan información útil que permite mejorar la predicción ingenua a la hora de predecir la dirección de movimiento del precio de cierre. Paralelamente se observa gracias a esta aplicación generalizada que los modelos SMD parecen obtener generalmente mejores resultados cuanto más cerca en el tiempo se hace la predicción. Es decir, en general, se obtienen mejores rendimientos cuando se predice la dirección que tomará el precio de cierre al cabo de un mes en vez de al cabo de tres meses. Finalmente también se destaca que esta aplicación masiva ha permitido confirmar una intuición que se desprende del ejemplo concreto aplicado a 4 empresas: el rendimiento

obtenido con este tipo de modelos SMD depende de la empresa a la que se aplique, o lo que es lo mismo, no todas las empresas obtienen un buen rendimiento usando este tipo de modelos. De esto se deduce que parece existir algún tipo de característica que hace que los modelos SMD aplicados a una determinada empresa tengan mejor rendimiento que los aplicados en otra. Esta investigación sobre esta característica podría ser el objeto de estudio de posteriores tesis o artículos académicos.

En tercer lugar esta tesis pretendía indagar en el campo de creciente implantación del *machine learning* automatizado. Este campo de estudio hace referencia a automatizar el proceso de creación de modelos de ML tales como modelos tipo *ensemble*, de manera que el usuario no tenga que construirlos "a mano". Gracias a la aplicación de los modelos SMD construidos de una manera automática usando todos los modelos disponibles en el paquete de R *h2o* se ha podido comprobar que se obtiene un rendimiento generalmente parecido al obtenido utilizando los modelos Random Forest. Sin embargo, la conclusión a la que se llega es que, en realidad, la utilización de funciones automatizadas para la creación de modelos de ML parece generalmente más interesante ya que permite obtener más o menos el mismo rendimiento que los modelos Random Forest creados manualmente pero con un esfuerzo o carga de trabajo muy inferiores. Resultados generalmente parecidos con muchas menos horas de trabajo en su realización. Este hecho ha permitido tomar conciencia de que la automatización en la construcción de modelos de ML es una de las tendencias que tiene mayor fuerza en el desarrollo teórico actual del campo de la inteligencia artificial. El hecho de automatizar una tarea, como puede ser la construcción de modelos de IA, puede permitir a la humanidad seguir desarrollando ideas en este campo, de manera que su continua evolución parece estar garantizada.

En cuarto lugar, otro de los grandes objetivos del presente trabajo era el de estudiar, en paralelo a la aplicación de los modelos SMD, la construcción de modelos predictivos entrenados directamente sobre el precio de cierre y la rentabilidad. En este sentido, también se pretendía analizar el rendimiento que pueden ofrecer los modelos llamados LSTM, al ser éstos capaces de recordar dependencias temporales grandes. Este tipo de modelos están en auge hoy en día a causa de su gran rendimiento en un variado abanico de aplicaciones, y es por eso que se consideraron ideales para ser probados sobre los datos concretos de una empresa. Aunque los precios de cierre son difíciles de predecir con modelos estadísticos clásicos, el tipo de redes neuronales recurrentes LSTM que se han utilizado en el presente trabajo han probado ser, tras su aplicación satisfactoria, capaces de captar relaciones que no son capaces de captar los modelos estadísticos clásicos. Sin embargo este tipo de modelos se ha aplicado sobre una empresa concreta por simplicidad, por lo que el rendimiento sobre otras empresas es por el momento inexplorado. A su vez, también se constata que sólo se ha realizado el primer paso de todo un seguido de procedimientos que deberían realizarse para aumentar la capacidad predictiva de estos modelos. Durante la redacción de esta tesis no se ha podido disponer de una alta elevada capcidad computacional por lo que se han tenido que mantener estáticos los hyper-parámetros de estos modelos. Por esta razón, se llega a la conclusión de que aun existe un gran potencial latente en este tipo de modelos en cuanto al rendimiento

predictivo, que sólo se puede obtener disponiendo de una gran capacidad computacional. Este aumento en la capacidad predictiva vendrá dado por una optimización intensiva de los hyper parámetros del modelo y por un proceso de entrenamiento suficientemente extenso, y ésto sólo se puede conseguir disponiendo de una alta capacidad computacional. De no ser así, el usuario intentando optimizar y entrenar este tipo de modelos se verá a si mismo enfrentando una gran cantidad de horas de espera requeridas hasta la finalización de dicho proceso de optimización y entrenamiento.

CAPÍTULO VII

BIBLIOGRAFÍA

Allaire, J., & Chollet, F. (2019). *Keras: R interface to 'keras'*. Retrieved from https://CRAN.R-project.org/package=keras

Andrew K. Rose, M. M. S. (2011). Cross-country causes and consequences of the 2008 crisis: Early warning. *Japan and the World Economy*.

An-Sing Chena, H. D., Mark T. Leungb. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index. *Computers & Operations Research, 30 (6) (2003)*, 901–923. Retrieved from https://ac.els-cdn.com/S0305054802000370/1-s2.0-S0305054802000370-main.pdf?_tid=a9ff1500-f141-436c-9298f0bf19d9c5c0&acdnat=1545913123_1b0ff6cdd7d647a8c5e9891ad3f6ea65

Board, F. S. (2017). *Artificial intelligence and machine learning in financial services. Market developments and financial stability implications*. 4–34.

Chen, L. (1989). *Protrader: An expert system for strock trading*.

C.L. Huang, C. T. (2009). A hybrid sofm-svr with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications, 36 (2) (2009)*, 1529–1539. Retrieved from https://ac.els-cdn.com/S0957417407006069/1-s2.0-S0957417407006069-main.pdf?_tid=ffd2e07d-4100-4d76bc64-653d8ae68de7&acdnat=1545913369_ce2c3b4a4eb42518498e95713c23e5c3

Day, S. (2017). *Quants turn to machine learning to model market impact*. RISK Magazine.

DeLong, J. B. (2009). The financial crisis of 2007–2009: Understanding its causes, consequences—and its possible cures. *MTI-CSC Economics Speaker Series Lecture*.

Dietterich, T. G. (n.d.). *Ensemble methods in machine learning*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.4718rep=rep1type=pdf

Dominic Masters, C. L. (2018). *Revisiting small batch training for deep neural networks*.

Funcas, K. y. (2017). *Fintech: Innovación al servicio del cliente*. 7.

Gilliland, M. (2009). The coefficient of variation for assessing forecastability. *The Business Forecasting Deal*.

Golberg, e. a. (1995). *The fincen artificial intelligence systems: Identifying potential money laundering from reports of large cash transactions*.

Han, Z. (2012). *Data and text mining of financial markets using news and social media*. 1–

102. Retrieved from https://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc12/FullText/Han-Zhichao-fulltext.pdf

Jonathan Brogaard, T. H., & Riordan, R. (2013). High frequency trading and price discovery. *Working Paper Series*, 2–57.

Jordan, M., & Mitchell, T. (2015). *Machine learning: Trends, perspectives, and prospects*.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing 55 (2003)*, 307–319. Retrieved from https://ac.els-cdn.com/S0925231203003722/1-s2.0-S0925231203003722-main.pdf?_tid=ec7d432b-e65c-434eb695-6d2035ff9829&acdnat=1545911976_ef14d01ea7992160caa0d8b4aa7c13aa

Knight, W. (2017). *The dark secret at the heart of ai*.

Kuroda, H. (2016). *Information technology and financial services: The central bank's perspective*.

Lambert, D. (1980). *Commodities(now called futures)*.

Leondes, C. T. (2002). *Expert systems: The technology of knowledge management and decision making for the 21st century* (pp. 1–22).

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. Retrieved from https://www.r-project.org/doc/Rnews/Rnews_2002-3.pdf

Luckyson Khaidem, S. R. D., Snehanshu Saha. (2016). Predicting the direction of stock market prices using random forest. *To Appear in Applied Mathematical Finance*, 21.

Manish Kumar, M. T. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. *Indian Institute of Capital Markets 9th Capital Markets Conference Paper*. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=876544

Masoud, N. (2014). Predicting direction of stock prices indexmovement using artificial neural networks:The case of libyan financial market. *British Journal of Economics, Management & Trade*, 597–619.

McCorduck, P. (2004). *Machines who think (2nd ed.)*. A. K. Peters, Ltd.

Mitchell, C. (2016). *Model validation: For elements of determining the accuracy of your model*. British Bankers Association.

Olah, C. (2015). *Understanding lstm networks*. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oliver Linton, S. M. (2018). Implications of high-frequency trading for security markets. *Cemmap*, 1–24.

O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. London: Allen Lane.

Philipp Probst, M. W., & Boulesteix, A.-L. (2018). *Hyperparameters and tuning strategies for random forest*. Retrieved from https://arxiv.org/pdf/1804.03515.pdf

P.R. Lane, G. M.-F. (2011). The cross-country incidence of the global crisis. *IMF Economic Review*, *59*, 77–110.

Ray Tsaih, C. C. L., Yenshan Hsu. (1998). Forecasting s&P 500 stock index futures with a hybrid ai system. *Decision Support Systems, 23 (1998)*, 161–174. Retrieved from https://ac.els-cdn.com/S0167923698000281/1-s2.0-S0167923698000281-main.pdf_tid=5c0465f0-1321-4ffd-920b-4ff50ab6fd3b&acdnat=1545912698_86722ea4bbc2cf26a7590b8bdfeab92d

Securities Commissions, I. O. of. (2017). *Research report on financial technologies (fintech).*

Stefan Lessmann, H.-V. S., Bart Baesens, & Thomas, L. (2015). Benchmarking state-of-the art classification algorithms for credit scoring: An update of research. In *European Journal of Operational Research 247* (pp. 124–136).

Stuart Russel, P. N. (1995). Artificial intelligence. Modern approach. *New Jersey: Prentice Hall, Englewood Cliffs*.

Ulrich, J. (2018). *TTR: Technical trading rules*. Retrieved from https://CRAN.R-project.org/package=TTR

Urban Jermann, V. Q. (2003). *Stock market boom and the productivity gains of the 1990s*.

Warren buffett : Latest portfolio. (n.d.). http://warrenbuffettstockportfolio.com/.

Wilder, J. (1978). New concepts in technical trading systems. *Trend Research Greensboro, North Carolina*.

Wilkins, C. (2017). *Blame it on the machines?*

workshop, W. D. (2019). *Dartmouth workshop — Wikipedia, the free encyclopedia*. http://en.wikipedia.org/w/index.php?title=Dartmouth%20workshop&oldid=878151960.

Yakup Kara, Ö. K. B., Melek Acar Boyacioglu. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Systems with Applications*, 5311–5319. Retrieved from https://ac.els-cdn.com/S0957417410011711/1-s2.0-S0957417410011711-main.pdf?_tid=e5e12593-4a56-4db0-9d5f ff9784196726&acdnat=1545074291_d05308f34977383caaff6ae009a6acf5

Y. Nakamori, S. W., W. Huang. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32 (10)*, 2513–2522.

CAPÍTULO VIII
ANEXO

En el siguiente anexo se añaden los scripts utilizados en el presente trabajo, ordenados por secciones siguiendo la estructura del mismo. Teniendo en cuenta que la propia tesis se ha escrito en R Markdown utilizando LaTeX como soporte, cabe destacar el hecho de que en el anexo se incluyen sólo los códigos para generar las partes técnicas del mismo, no el codigo de R Markdown para generar la tesis. Éste se puede encontrar en un proyecto en el siguiente enlace de GitHub: https://github.com/ArnauMunsOrenga/TFG. Sin embargo, sí que se incluye el código necesario para generar las tablas y los gráficos.

Apartado V.1

Obtención y descripción

```
library(CombMSC)
library(randomForest)
library(quantmod)
library(TTR)
library(tidyverse)
library(caret)
library(foreach)
library(kableExtra)
library(dplyr)
library(formatR)

getSymbols(Symbols = "KO", from = "2000-01-01", to = "2018-12-31")
getSymbols(Symbols = "WFC", from = "2000-01-01", to = "2018-12-31")
getSymbols(Symbols = "AAPL", from = "2000-01-01", to = "2018-12-31")
getSymbols(Symbols = "AXP", from = "2000-01-01", to = "2018-12-31")

prices.KO <- as.data.frame(KO)
prices.WFC <- as.data.frame(WFC)
prices.AAPL <- as.data.frame(AAPL)
prices.AXP <- as.data.frame(AXP)
str(prices.KO)


a <- format(data.frame(paste0(round(summary(prices.KO$KO.Open),
    2)) %>% rbind(a = paste0(round(summary(prices.KO$KO.High),
    2))) %>% rbind(b = paste0(round(summary(prices.KO$KO.Low),
    2))) %>% rbind(c = paste0(round(summary(prices.KO$KO.Close),
    2))) %>% rbind(paste0(round(summary(prices.KO$KO.Volume)/1e+06,
    3), "M"))), scientific = TRUE)

rownames(a) <- c("Open", "High", "Low", "Close", "Volume")
names(a) <- names(summary(prices.KO$KO.Open))

kable(a, "latex", digits = 2) %>% kable_styling(font_size = 10,
    latex_options = c("basic"))
# {Estadísticos descriptivos para los distintos precios de
# Coca-Cola Company}


a <- format(data.frame(paste0(round(summary(prices.AAPL$AAPL.Open),
    2)) %>% rbind(a = paste0(round(summary(prices.AAPL$AAPL.High),
    2))) %>% rbind(b = paste0(round(summary(prices.AAPL$AAPL.Low),
```

```
    2))) %>% rbind(c = paste0(round(summary(prices.AAPL$AAPL.Close),
    2))) %>% rbind(paste0(round(summary(prices.AAPL$AAPL.Volume)/1e+06,
    3), "M"))), scientific = TRUE)

rownames(a) <- c("Open", "High", "Low", "Close", "Volume")
names(a) <- names(summary(prices.KO$KO.Open))
kable(a, "latex") %>% kable_styling(font_size = 10, latex_options = c("basic"))

# {Estadísticos descriptivos para los distintos precios de
# Apple Inc.}


a <- format(data.frame(paste0(round(summary(prices.AXP$AXP.Open),
    2)) %>% rbind(a = paste0(round(summary(prices.AXP$AXP.High),
    2))) %>% rbind(b = paste0(round(summary(prices.AXP$AXP.Low),
    2))) %>% rbind(c = paste0(round(summary(prices.AXP$AXP.Close),
    2))) %>% rbind(paste0(round(summary(prices.AXP$AXP.Volume)/1e+06,
    3), "M"))), scientific = TRUE)

rownames(a) <- c("Open", "High", "Low", "Close", "Volume")
names(a) <- names(summary(prices.KO$KO.Open))
kable(a, "latex", digits = 2) %>% kable_styling(font_size = 10,
    latex_options = c("basic"))
# {Estadísticos descriptivos para los distintos precios de
# American Express CO.}


a <- format(data.frame(paste0(round(summary(prices.WFC$WFC.Open),
    2)) %>% rbind(a = paste0(round(summary(prices.WFC$WFC.High),
    2))) %>% rbind(b = paste0(round(summary(prices.WFC$WFC.Low),
    2))) %>% rbind(c = paste0(round(summary(prices.WFC$WFC.Close),
    2))) %>% rbind(paste0(round(summary(prices.WFC$WFC.Volume)/1e+06,
    3), "M"))), scientific = TRUE)

rownames(a) <- c("Open", "High", "Low", "Close", "Volume")
names(a) <- names(summary(prices.KO$KO.Open))
kable(a, "latex", digits = 2) %>% # {Estadísticos descriptivos para los distintos precios de
# Wells Fargo and CO.}


chartSeries(KO$KO.Close, name = "KO Close", color.vol = T)

chartSeries(AAPL$AAPL.Close, "candlesticks", name = "AAPL Close price",
    color.vol = T)

chartSeries(AXP$AXP.Close, "candlesticks", name = "AXP Close price",
    color.vol = T)

chartSeries(WFC$WFC.Close, "candlesticks", name = "WFC Close price",
    color.vol = T)



kable(data.frame(Nombre = c("Apple Inc.", "Wells Fargo & CO",
    "Coca-Cola Company", "American Express CO"), Media = c(mean(prices.AAPL$AAPL.Open),
    mean(prices.WFC$WFC.Open), mean(prices.KO$KO.Open), mean(prices.AXP$AXP.Open)),
    Desv_std = c(sd(prices.AAPL$AAPL.Open), sd(prices.WFC$WFC.Open),
        sd(prices.KO$KO.Open), sd(prices.AXP$AXP.Open)), CoV = c((sd(prices.AAPL$AAPL.Open)/mean(prices.AAPL$AAPL.Open)),
        (sd(prices.WFC$WFC.Open)/mean(prices.WFC$WFC.Open)),
        (sd(prices.KO$KO.Open/mean(prices.KO$KO.Open))), (sd(prices.AXP$AXP.Open)/mean(prices.AXP$AXP.Open)))),
    "latex") %>% kable_styling(font_size = 10, latex_options = c("basic"))

kable(data.frame(Nombre = c("Apple Inc.", "Wells Fargo & CO",
    "Coca-Cola Company", "American Express CO"), Media = c(mean(prices.AAPL$AAPL.High),
    mean(prices.WFC$WFC.High), mean(prices.KO$KO.High), mean(prices.AXP$AXP.High)),
    Desv_std = c(sd(prices.AAPL$AAPL.High), sd(prices.WFC$WFC.High),
        sd(prices.KO$KO.High), sd(prices.AXP$AXP.High)), CoV = c((sd(prices.AAPL$AAPL.High)/mean(prices.AAPL$AAPL.High)),
```

```
        (sd(prices.WFC$WFC.High)/mean(prices.WFC$WFC.High)),
        (sd(prices.KO$KO.High/mean(prices.KO$KO.High))), (sd(prices.AXP$AXP.High)/mean(prices.AXP$AXP.High)))),
    "latex") %>% kable_styling(font_size = 10, latex_options = c("basic"))

kable(data.frame(Nombre = c("Apple Inc.", "Wells Fargo & CO",
    "Coca-Cola Company", "American Express CO"), Media = c(mean(prices.AAPL$AAPL.Low),
    mean(prices.WFC$WFC.Low), mean(prices.KO$KO.Low), mean(prices.AXP$AXP.Low)),
    Desv_std = c(sd(prices.AAPL$AAPL.Low), sd(prices.WFC$WFC.Low),
        sd(prices.KO$KO.Low), sd(prices.AXP$AXP.Low)), CoV = c((sd(prices.AAPL$AAPL.Low)/mean(prices.AAPL$AAPL.Low)),
        (sd(prices.WFC$WFC.Low)/mean(prices.WFC$WFC.Low)), (sd(prices.KO$KO.Low/mean(prices.KO$KO.Low))),
        (sd(prices.AXP$AXP.Low)/mean(prices.AXP$AXP.Low)))),
    "latex") %>% kable_styling(font_size = 10, latex_options = c("basic"))

kable(data.frame(Nombre = c("Apple Inc.", "Wells Fargo & CO",
    "Coca-Cola Company", "American Express CO"), Media = c(mean(prices.AAPL$AAPL.Close),
    mean(prices.WFC$WFC.Close), mean(prices.KO$KO.Close), mean(prices.AXP$AXP.Close)),
    Desv_std = c(sd(prices.AAPL$AAPL.Close), sd(prices.WFC$WFC.Close),
        sd(prices.KO$KO.Close), sd(prices.AXP$AXP.Close)), CoV = c((sd(prices.AAPL$AAPL.Close)/mean(prices.AAPL$AAPL.Close)),
        (sd(prices.WFC$WFC.Close)/mean(prices.WFC$WFC.Close)),
        (sd(prices.KO$KO.Close/mean(prices.KO$KO.Close))), (sd(prices.AXP$AXP.Close)/mean(prices.AXP$AXP.Close)))),
    "latex") %>% kable_styling(font_size = 10, latex_options = c("basic"))
```

Feature extraction

En este apartado se simplifica el código y **se muestra sólo para el caso de una empresa concreta (KO)**. Hay que tener en cuenta que el código original de este apartado incluye las 4 empresas utilizadas como ejemplo.

```
# funcion que calcula la variable respuesta tal como está
# definida en el presente trabajo
target_calc <- function(data, freq) {
    target <- c()
    for (i in 1:(nrow(data) - freq)) {
        if (as.numeric(data[i + freq, 4]) > as.numeric(data[i,
            4])) {
            target[i] <- "Up"
        } else {
            target[i] <- "Down"
        }
    }
    return(target)
}

# función de feature extraction: calcula los indicadores
# técnicos utilizados en el trabajo
feature_extraction_finance <- function(data) {
    data <- data %>%
    # Aroon
    cbind(aroon(data[, c("High", "Low")], 20)) %>%
    #----------------------------------------

    # Simple moving average 10 day
    cbind(sma10 = SMA(data[, "Close"], 10)) %>%
    #----------------------------------------

    # exponential moving average 10 day
    # cbind(ema10=EMA(data[,'Close'],10)) %>%

    #----------------------------------------

    # momentum 1 day
    cbind(mom1 = momentum(data[, "Close"], 1)) %>% cbind(mom2 = momentum(data[,
        "Close"], 2)) %>% cbind(mom3 = momentum(data[, "Close"],
```

```
    3)) %>% cbind(mom4 = momentum(data[, "Close"], 4)) %>%
    cbind(mom5 = momentum(data[, "Close"], 5)) %>%
# momentum 9 day
cbind(mom9 = momentum(data[, "Close"], 9)) %>%
cbind(mom15 = momentum(data[, "Close"], 15)) %>%
#---------------------------------------

# Rate of change 1 day
cbind(ROC1 = ROC(data[, "Close"], 1)) %>%
# Rate of change 9 day
cbind(ROC9 = ROC(data[, "Close"], 9)) %>%
#---------------------------------------

# fastK%, fastD% and slowD% nFastK = 14, nFastD = 3, nSlowD =
# 3
cbind(data.frame(stoch(as.xts(data[, c("High", "Low", "Close")])))) %>%

# SMI n = 13, nFast = 2, nSlow = 25
cbind(SMI = data.frame(SMI(as.xts(data[, c("High", "Low",
    "Close")])))[, 1]) %>%
#---------------------------------------

# RSI
cbind(RSI = RSI(data[, "Close"])) %>%
#---------------------------------------

# Williams Accumulation/Distribution
cbind(data.frame(wAD = williamsAD(as.xts(data[, c("High",
    "Low", "Close")])))) %>%

# Williams Percentage range
cbind(data.frame(WPR(as.xts(data[, c("High", "Low", "Close")]))))

# names(data)[23]<-'WPR'
names(data)[4] <- "Close_"
names(data)[which(names(data) == "Close")] <- "WPR"
names(data)[4] <- "Close"
#---------------------------------------

data <- data %>% # Moving Average convergence divergence
cbind(macd = MACD(data[, "Close"], 12, 26, 9, maType = "EMA")[,
    1]) %>%
#---------------------------------------

# Comodity Channel Index
cbind(data.frame(CCI = CCI(as.xts(data[, c("High", "Low",
    "Close")])))) %>%
#---------------------------------------

# On Balance Volume
cbind(OBV = OBV(data[, "Close"], data[, "Volume"])) %>%
#---------------------------------------

# Average true range, true range1 and true range2
cbind(data.frame(ATR(as.xts(data[, c("High", "Low", "Close")]))))

b3 <- data.frame(tr2 = ((data[, 4] - data$trueLow)/(data$trueHigh -
    data$trueLow)))

data <- data %>% cbind(b3) %>%
select(-c(trueHigh, trueLow)) %>%
#---------------------------------------

# Trend detection index, using both TDI and DI
cbind(TDI(data[, "Close"])) %>%
# ADX and DX and DIp/DIn
cbind(data.frame(ADX(as.xts(data[, c("High", "Low", "Close")])),
```

```
        20)))

    b2 <- data.frame(PNratio = data$DIp/data$DIn)

    data <- data %>% cbind(b2) %>%
    select(-c(DIp, DIn)) %>%
    # Bollinger band width
    cbind(data.frame(BBands(as.xts(data[, c("High", "Low", "Close")]))))

    b <- data.frame(BBwidth = ((data$up - data$dn)/data$mavg))

    data <- data %>% cbind(b) %>%
    select(-c(dn, up, mavg)) %>%
    #-----------------------------------
    select(-c(oscillator))

    data <- data[-c(1:39), ]  #deleting NA generated by average calculations

    data <- data[, -c(1:5)]  #borramos variables que no se utilizan

    return(data)
}

########################################################################################################## funciones de alisado

# alisa los datos usando una EMA y el periodo definido
EMA_finance <- function(data, smoothing_period) {
    dates <- rownames(as.data.frame(EMA(data[, 4], smoothing_period)))

    data <- as.data.frame(EMA(data[, 1], smoothing_period)) %>%
        bind_cols(as.data.frame(EMA(data[, 2], smoothing_period))) %>%
        bind_cols(as.data.frame(EMA(data[, 3], smoothing_period))) %>%
        bind_cols(as.data.frame(EMA(data[, 4], smoothing_period))) %>%
        bind_cols(as.data.frame(EMA(data[, 5], smoothing_period)))

    data <- data %>% slice(first(which(!is.na(.[, 4]))):nrow(.))

    colnames(data) <- c("Open", "High", "Low", "Close", "Volume")
    rownames(data) <- dates[-c(1:(length(dates) - nrow(data)))]

    return(data)
}

# función que alisa los datos usando una SES
smooth_formula <- function(data) {


    smooth.open <- c()
    smooth.low <- c()
    smooth.high <- c()
    smooth.close <- c()
    smooth.close <- c()
    smooth.volume <- c()


    data <- as.data.frame(data)

    smooth.open[1] <- data[1, 1]
    smooth.high[1] <- data[1, 2]
    smooth.low[1] <- data[1, 3]
    smooth.close[1] <- data[1, 4]
    smooth.volume[1] <- data[1, 5]

    alpha <- 0.05

    for (i in 2:nrow(data)) {
        smooth.open[i] <- (alpha * (data[i, 1])) + ((1 - alpha) *
```

```
            (smooth.open[i - 1]))
        smooth.high[i] <- alpha * data[i, 2] + ((1 - alpha) *
            smooth.high[i - 1])
        smooth.low[i] <- alpha * data[i, 3] + ((1 - alpha) *
            smooth.low[i - 1])
        smooth.close[i] <- alpha * data[i, 4] + ((1 - alpha) *
            smooth.close[i - 1])
        smooth.volume[i] <- alpha * data[i, 5] + ((1 - alpha) *
            smooth.volume[i - 1])
    }

    dataa <- data.frame(Open = smooth.open, High = smooth.high,
        Low = smooth.low, Close = smooth.close, Volume = smooth.volume)

    return(dataa)
}

############################################################################################## 3
```

Modelling

```
KO_30<-EMA_finance(KO,30)
KO_60<-EMA_finance(KO,60)
KO_90<-EMA_finance(KO,90)
KO_formula<-smooth_formula(KO)

#############################################################################################
#########COCA COLA COMPANY
  ####### 30 day EMA
    KO_30_1m<-KO_30 %>%
      slice(1:(nrow(.)-20)) %>%
      bind_cols(target=target_calc(KO_30,20))

rownames(KO_30_1m)<-rownames(KO_30)[c(1:(nrow(KO_30)-20))]
KO_30_1m$target<-factor(KO_30_1m$target)

    KO_30_2m<-KO_30 %>%
      slice(1:(nrow(.)-40)) %>%
      bind_cols(target=target_calc(KO_30,40))


    KO_30_3m<-KO_30 %>%
      slice(1:(nrow(.)-60)) %>%
      bind_cols(target=target_calc(KO_30,60))

        rownames(KO_30_2m)<-rownames(KO_30)[c(1:(nrow(KO_30)-40))]
KO_30_2m$target<-factor(KO_30_2m$target)

rownames(KO_30_3m)<-rownames(KO_30)[c(1:(nrow(KO_30)-60))]
KO_30_3m$target<-factor(KO_30_3m$target)

    # KO_30_target<-list(KO_30_1m = KO_30_1m,
    #                    KO_30_2m = KO_30_2m,
    #                    KO_30_3m = KO_30_3m)
    #rm(KO_30_1m,KO_30_2m,KO_30_3m)
########################
  ####### 60 day EMA
    KO_60_1m<-KO_60 %>%
      slice(1:(nrow(.)-20)) %>%
      bind_cols(target=target_calc(KO_60,20))


    KO_60_2m<-KO_60 %>%
      slice(1:(nrow(.)-40)) %>%
      bind_cols(target=target_calc(KO_60,40))
```

```
    KO_60_3m<-KO_60 %>%
      slice(1:(nrow(.)-60)) %>%
      bind_cols(target=target_calc(KO_60,60))

        rownames(KO_60_1m)<-rownames(KO_60)[c(1:(nrow(KO_60)-20))]
KO_60_1m$target<-factor(KO_60_1m$target)

rownames(KO_60_2m)<-rownames(KO_60)[c(1:(nrow(KO_60)-40))]
KO_60_2m$target<-factor(KO_60_2m$target)

rownames(KO_60_3m)<-rownames(KO_60)[c(1:(nrow(KO_60)-60))]
KO_60_3m$target<-factor(KO_60_3m$target)


    # KO_60_target<-list(KO_60_1m = KO_60_1m,
    #                    KO_60_2m = KO_60_2m,
    #                    KO_60_3m = KO_60_3m)
    #rm(KO_60_1m,KO_60_2m,KO_60_3m)
##############################################3
  ####### 90 day EMA
    KO_90_1m<-KO_90 %>%
      slice(1:(nrow(.)-20)) %>%
      bind_cols(target=target_calc(KO_90,20))

    KO_90_2m<-KO_90 %>%
      slice(1:(nrow(.)-40)) %>%
      bind_cols(target=target_calc(KO_90,40))




    KO_90_3m<-KO_90 %>%
      slice(1:(nrow(.)-60)) %>%
      bind_cols(target=target_calc(KO_90,60))

    rownames(KO_90_1m)<-rownames(KO_90)[c(1:(nrow(KO_90)-20))]
KO_90_1m$target<-factor(KO_90_1m$target)
rownames(KO_90_2m)<-rownames(KO_90)[c(1:(nrow(KO_90)-40))]
KO_90_2m$target<-factor(KO_90_2m$target)

rownames(KO_90_3m)<-rownames(KO_90)[c(1:(nrow(KO_90)-60))]
KO_90_3m$target<-factor(KO_90_3m$target)

    # KO_90_target<-list(KO_90_1m = KO_90_1m,
    #                    KO_90_2m = KO_90_2m,
    #                    KO_90_3m = KO_90_3m)
    #rm(KO_90_1m,KO_90_2m,KO_90_3m)
####################################
    #target on formula smoothed data
    KO_fun_1m<-KO_formula %>%
      slice(1:(nrow(.)-20)) %>%
      bind_cols(target=target_calc(KO_formula,20))

    KO_fun_2m<-KO_formula %>%
      slice(1:(nrow(.)-40)) %>%
      bind_cols(target=target_calc(KO_formula,40))

    KO_fun_3m<-KO_formula %>%
      slice(1:(nrow(.)-60)) %>%
      bind_cols(target=target_calc(KO_formula,60))

  rownames(KO_fun_1m)<-index(KO)[c(1:(nrow(KO)-20))]
KO_fun_1m$target<-factor(KO_fun_1m$target)

rownames(KO_fun_2m)<-index(KO)[c(1:(nrow(KO)-40))]
KO_fun_2m$target<-factor(KO_fun_2m$target)
```

```
rownames(KO_fun_3m)<-index(KO)[c(1:(nrow(KO)-60))]
KO_fun_3m$target<-factor(KO_fun_3m$target)


a<-data.frame(EMA30=c(r1=(table(KO_30_1m$target)/sum(table(KO_30_1m$target))),
                      r2=(table(KO_30_2m$target)/sum(table(KO_30_2m$target))),
                      r3=(table(KO_30_3m$target)/sum(table(KO_30_3m$target)))),
              EMA60=c(r1=(table(KO_60_1m$target)/sum(table(KO_60_1m$target))),
                      r2=(table(KO_60_2m$target)/sum(table(KO_60_2m$target))),
                      r3=(table(KO_60_3m$target)/sum(table(KO_60_3m$target)))),
              EMA90=c(r1=(table(KO_90_1m$target)/sum(table(KO_90_1m$target))),
                      r2=(table(KO_90_2m$target)/sum(table(KO_90_2m$target))),
                      r3=(table(KO_90_3m$target)/sum(table(KO_90_3m$target)))),
              exp.smooth=c(r1=(table(KO_fun_1m$target)/sum(table(KO_fun_1m$target))),
                           r2=(table(KO_fun_2m$target)/sum(table(KO_fun_2m$target))),
                           r3=(table(KO_fun_3m$target)/sum(table(KO_fun_3m$target))))) %>%
  t(.) %>% as_tibble() %>%
  mutate_all(.funs = function(x) x*100)

rownames(a)<-c("EMA30","EMA60","EMA90","Alisado exponencial")

kable(a, "latex") %>%
add_header_above(c(" ", "Predicción 1 mes" = 2, "Predicción 2 meses" = 2,"Predicción 3 meses"=2)) %>%
  kable_styling(font_size = 10,latex_options = c("basic"))
#{Proporción de la variable respuesta Coca-Cola CO.}

#feature extraction
KO_30_1m_target_feature<-feature_extraction_finance(KO_30_1m)
KO_30_2m_target_feature<-feature_extraction_finance(KO_30_2m)
KO_30_3m_target_feature<-feature_extraction_finance(KO_30_3m)

KO_60_1m_target_feature<-feature_extraction_finance(KO_60_1m)
KO_60_2m_target_feature<-feature_extraction_finance(KO_60_2m)
KO_60_3m_target_feature<-feature_extraction_finance(KO_60_3m)

KO_90_1m_target_feature<-feature_extraction_finance(KO_90_1m)
KO_90_2m_target_feature<-feature_extraction_finance(KO_90_2m)
KO_90_3m_target_feature<-feature_extraction_finance(KO_90_3m)

KO_fun_1m_target_feature<-feature_extraction_finance(KO_fun_1m)
KO_fun_2m_target_feature<-feature_extraction_finance(KO_fun_2m)
KO_fun_3m_target_feature<-feature_extraction_finance(KO_fun_3m)

#Partición train-validation-test
train_KO_30_1m_target_feature<-
  KO_30_1m_target_feature[year(rownames(KO_30_1m_target_feature))%in%c(2000:2015),]

validation_KO_30_1m_target_feature<-
  KO_30_1m_target_feature[year(rownames(KO_30_1m_target_feature))%in%c( 2016) ,]

test_KO_30_1m_target_feature<-
  KO_30_1m_target_feature[year(rownames(KO_30_1m_target_feature))%in%c(2017:2018) ,]



train_KO_60_1m_target_feature<-
  KO_60_1m_target_feature[year(rownames(KO_60_1m_target_feature))%in%c(2000:2015),]

validation_KO_60_1m_target_feature<-
  KO_60_1m_target_feature[year(rownames(KO_60_1m_target_feature))%in%c( 2016) ,]

test_KO_60_1m_target_feature<-
  KO_60_1m_target_feature[year(rownames(KO_60_1m_target_feature))%in%c(2017:2018) ,]
```

```
train_KO_90_1m_target_feature<-
  KO_90_1m_target_feature[year(rownames(KO_90_1m_target_feature))%in%c(2000:2015),]

validation_KO_90_1m_target_feature<-
  KO_90_1m_target_feature[year(rownames(KO_90_1m_target_feature))%in%c( 2016) ,]

test_KO_90_1m_target_feature<-
  KO_90_1m_target_feature[year(rownames(KO_90_1m_target_feature))%in%c(2017:2018) ,]



train_KO_30_2m_target_feature<-
  KO_30_2m_target_feature[year(rownames(KO_30_2m_target_feature))%in%c(2000:2015),]

validation_KO_30_2m_target_feature<-
  KO_30_2m_target_feature[year(rownames(KO_30_2m_target_feature))%in%c( 2016),]

test_KO_30_2m_target_feature<-
  KO_30_2m_target_feature[year(rownames(KO_30_2m_target_feature))%in%c(2017:2018),]



train_KO_60_2m_target_feature<-
  KO_60_2m_target_feature[year(rownames(KO_60_2m_target_feature))%in%c(2000:2015),]

validation_KO_60_2m_target_feature<-
  KO_60_2m_target_feature[year(rownames(KO_60_2m_target_feature))%in%c( 2016),]

test_KO_60_2m_target_feature<-
  KO_60_2m_target_feature[year(rownames(KO_60_2m_target_feature))%in%c(2017:2018),]


train_KO_90_2m_target_feature<-
  KO_90_2m_target_feature[year(rownames(KO_90_2m_target_feature))%in%c(2000:2015),]

validation_KO_90_2m_target_feature<-
  KO_90_2m_target_feature[year(rownames(KO_90_2m_target_feature))%in%c( 2016) ,]

test_KO_90_2m_target_feature<-
  KO_90_2m_target_feature[year(rownames(KO_90_2m_target_feature))%in%c(2017:2018) ,]



train_KO_30_3m_target_feature<-
  KO_30_3m_target_feature[year(rownames(KO_30_3m_target_feature))%in%c(2000:2015),]

validation_KO_30_3m_target_feature<-
  KO_30_3m_target_feature[year(rownames(KO_30_3m_target_feature))%in%c( 2016) ,]

test_KO_30_3m_target_feature<-
  KO_30_3m_target_feature[year(rownames(KO_30_3m_target_feature))%in%c(2017:2018) ,]



train_KO_60_3m_target_feature<-
  KO_60_3m_target_feature[year(rownames(KO_60_3m_target_feature))%in%c(2000:2015),]

validation_KO_60_3m_target_feature<-
  KO_60_3m_target_feature[year(rownames(KO_60_3m_target_feature))%in%c( 2016) ,]

test_KO_60_3m_target_feature<-
  KO_60_3m_target_feature[year(rownames(KO_60_3m_target_feature))%in%c(2017:2018) ,]


train_KO_90_3m_target_feature<-
  KO_90_3m_target_feature[year(rownames(KO_90_3m_target_feature))%in%c(2000:2015),]
```

```
validation_KO_90_3m_target_feature<-
  KO_90_3m_target_feature[year(rownames(KO_90_3m_target_feature))%in%c( 2016) ,]

test_KO_90_3m_target_feature<-
  KO_90_3m_target_feature[year(rownames(KO_90_3m_target_feature))%in%c(2017:2018) ,]


#fun
train_KO_fun_1m_target_feature<-
  KO_fun_1m_target_feature[year(rownames(KO_fun_1m_target_feature))%in%c(2000:2015),]

validation_KO_fun_1m_target_feature<-
  KO_fun_1m_target_feature[year(rownames(KO_fun_1m_target_feature))%in%c( 2016),]

test_KO_fun_1m_target_feature<-
  KO_fun_1m_target_feature[year(rownames(KO_fun_1m_target_feature))%in%c(2017:2018),]

train_KO_fun_2m_target_feature<-
  KO_fun_2m_target_feature[year(rownames(KO_fun_2m_target_feature))%in%c(2000:2015),]

validation_KO_fun_2m_target_feature<-
  KO_fun_2m_target_feature[year(rownames(KO_fun_2m_target_feature))%in%c( 2016),]

test_KO_fun_2m_target_feature<-
  KO_fun_2m_target_feature[year(rownames(KO_fun_2m_target_feature))%in%c(2017:2018),]

train_KO_fun_3m_target_feature<-
  KO_fun_3m_target_feature[year(rownames(KO_fun_3m_target_feature))%in%c(2000:2015),]

validation_KO_fun_3m_target_feature<-
  KO_fun_3m_target_feature[year(rownames(KO_fun_3m_target_feature))%in%c( 2016),]

test_KO_fun_3m_target_feature<-
  KO_fun_3m_target_feature[year(rownames(KO_fun_3m_target_feature))%in%c(2017:2018),]


#mtry fine tuning for RF.
errors_KO_30_1m_target_feature<-c()
errors_KO_30_2m_target_feature<-c()
errors_KO_30_3m_target_feature<-c()

errors_KO_60_1m_target_feature<-c()
errors_KO_60_2m_target_feature<-c()
errors_KO_60_3m_target_feature<-c()

errors_KO_90_1m_target_feature<-c()
errors_KO_90_2m_target_feature<-c()
errors_KO_90_3m_target_feature<-c()



for(i in 1:32){
   model_KO_30_1m_target_feature<-randomForest(target~.,data=train_KO_30_1m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_30_1m_target_feature,validation_KO_30_1m_target_feature),
                  validation_KO_30_1m_target_feature$target)->a_KO_30_1m_target_feature
  errors_KO_30_1m_target_feature<-c(errors_KO_30_1m_target_feature,1-a_KO_30_1m_target_feature$overall[1])

  model_KO_30_2m_target_feature<-randomForest(target~.,data=train_KO_30_2m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_30_2m_target_feature,validation_KO_30_2m_target_feature),
                  validation_KO_30_2m_target_feature$target)->a_KO_30_2m_target_feature
  errors_KO_30_2m_target_feature<-c(errors_KO_30_2m_target_feature,1-a_KO_30_2m_target_feature$overall[1])


  model_KO_30_3m_target_feature<-randomForest(target~.,data=train_KO_30_3m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_30_3m_target_feature,validation_KO_30_3m_target_feature),
                  validation_KO_30_3m_target_feature$target)->a_KO_30_3m_target_feature
  errors_KO_30_3m_target_feature<-c(errors_KO_30_3m_target_feature,1-a_KO_30_3m_target_feature$overall[1])
```

```
  #~~

  model_KO_60_1m_target_feature<-randomForest(target~.,data=train_KO_60_1m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_60_1m_target_feature,validation_KO_60_1m_target_feature),
                  validation_KO_60_1m_target_feature$target)->a_KO_60_1m_target_feature
  errors_KO_60_1m_target_feature<-c(errors_KO_60_1m_target_feature,1-a_KO_60_1m_target_feature$overall[1])

  model_KO_60_2m_target_feature<-randomForest(target~.,data=train_KO_60_2m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_60_2m_target_feature,validation_KO_60_2m_target_feature),
                  validation_KO_60_2m_target_feature$target)->a_KO_60_2m_target_feature
  errors_KO_60_2m_target_feature<-c(errors_KO_60_2m_target_feature,1-a_KO_60_2m_target_feature$overall[1])


  model_KO_60_3m_target_feature<-randomForest(target~.,data=train_KO_60_3m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_60_3m_target_feature,validation_KO_60_3m_target_feature),
                  validation_KO_60_3m_target_feature$target)->a_KO_60_3m_target_feature
  errors_KO_60_3m_target_feature<-c(errors_KO_60_3m_target_feature,1-a_KO_60_3m_target_feature$overall[1])

  #~~
  model_KO_90_1m_target_feature<-randomForest(target~.,data=train_KO_90_1m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_90_1m_target_feature,validation_KO_90_1m_target_feature),
                  validation_KO_90_1m_target_feature$target)->a_KO_90_1m_target_feature
  errors_KO_90_1m_target_feature<-c(errors_KO_90_1m_target_feature,1-a_KO_90_1m_target_feature$overall[1])

  model_KO_90_2m_target_feature<-randomForest(target~.,data=train_KO_90_2m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_90_2m_target_feature,validation_KO_90_2m_target_feature),
                  validation_KO_90_2m_target_feature$target)->a_KO_90_2m_target_feature
  errors_KO_90_2m_target_feature<-c(errors_KO_90_2m_target_feature,1-a_KO_90_2m_target_feature$overall[1])


  model_KO_90_3m_target_feature<-randomForest(target~.,data=train_KO_90_3m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_90_3m_target_feature,validation_KO_90_3m_target_feature),
                  validation_KO_90_3m_target_feature$target)->a_KO_90_3m_target_feature
  errors_KO_90_3m_target_feature<-c(errors_KO_90_3m_target_feature,1-a_KO_90_3m_target_feature$overall[1])

  ##
  print("KO OK")


  model_KO_fun_1m_target_feature<-randomForest(target~.,data=train_KO_fun_1m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_fun_1m_target_feature,validation_KO_fun_1m_target_feature),
                  validation_KO_fun_1m_target_feature$target)->a_KO_fun_1m_target_feature
  errors_KO_fun_1m_target_feature<-c(errors_KO_fun_1m_target_feature,1-a_KO_fun_1m_target_feature$overall[1])

  model_KO_fun_2m_target_feature<-randomForest(target~.,data=train_KO_fun_2m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_fun_1m_target_feature,validation_KO_fun_2m_target_feature),
                  validation_KO_fun_2m_target_feature$target)->a_KO_fun_2m_target_feature
  errors_KO_fun_2m_target_feature<-c(errors_KO_fun_2m_target_feature,1-a_KO_fun_2m_target_feature$overall[1])

  model_KO_fun_3m_target_feature<-randomForest(target~.,data=train_KO_fun_3m_target_feature,mtry=i)
  confusionMatrix(predict(model_KO_fun_1m_target_feature,validation_KO_fun_3m_target_feature),
                  validation_KO_fun_3m_target_feature$target)->a_KO_fun_3m_target_feature
  errors_KO_fun_3m_target_feature<-c(errors_KO_fun_3m_target_feature,1-a_KO_fun_3m_target_feature$overall[1])

  print("fun OK")


  print(paste0(i," finished"))
}


#KO
#{Coca Cola CO.: valores optimizados para mtry y accuracy obtenida}
a<-data.frame(mtry1m=as.numeric(c(which(errors_KO_30_1m_target_feature==min(errors_KO_30_1m_target_feature))[1],
                                  which(errors_KO_60_1m_target_feature==min(errors_KO_60_1m_target_feature[-1]))[1],
                                  which(errors_KO_90_1m_target_feature==min(errors_KO_90_1m_target_feature[-1]))[1],
                                  which(errors_KO_fun_1m_target_feature==min(errors_KO_fun_1m_target_feature[-1]))[1])),
```

```
              accuracy1m=as.numeric(c((1-min(errors_KO_30_1m_target_feature))*100,
                                      (1-min(errors_KO_60_1m_target_feature[-1]))*100,
                                      (1-min(errors_KO_90_1m_target_feature[-1]))*100,
                                      (1-min(errors_KO_fun_1m_target_feature[-1]))*100)),


              mtry2m=as.numeric(c(which(errors_KO_30_2m_target_feature==min(errors_KO_30_2m_target_feature[-1]))[1],
                                  which(errors_KO_60_2m_target_feature==min(errors_KO_60_2m_target_feature[-1]))[1],
                                 which(errors_KO_90_2m_target_feature==min(errors_KO_90_2m_target_feature))[1],
                                 which(errors_KO_fun_2m_target_feature==min(errors_KO_fun_2m_target_feature))[1])),

                 accuracy2m=as.numeric(c((1-min(errors_KO_30_2m_target_feature[-1]))*100,
                                      (1-min(errors_KO_60_2m_target_feature[-1]))*100,
                                      (1-min(errors_KO_90_2m_target_feature))*100,
                                      (1-min(errors_KO_fun_2m_target_feature))*100)),


              mtry3m=as.numeric(c(which(errors_KO_30_3m_target_feature==min(errors_KO_30_3m_target_feature[-1]))[1],
                                  which(errors_KO_60_3m_target_feature==min(errors_KO_60_3m_target_feature[-1]))[1],
                                 which(errors_KO_90_3m_target_feature==min(errors_KO_90_3m_target_feature[-1]))[1],
                                 which(errors_KO_fun_3m_target_feature==min(errors_KO_fun_3m_target_feature))[1])),

              accuracy3m=as.numeric(c((1-min(errors_KO_30_3m_target_feature[-1]))*100,
                                      (1-min(errors_KO_60_3m_target_feature[-1]))*100,
                                      (1-min(errors_KO_90_3m_target_feature[-1]))*100,
                                      (1-min(errors_KO_fun_3m_target_feature))*100))

              )

rownames(a)<-c("EMA30","EMA60","EMA90","Alisado exponencial")

kable(a, "latex") %>%
  add_header_above(c(" ", "Predicción 1 mes" = 2, "Predicción 2 meses" = 2,"Predicción 3 meses"=2)) %>%
  kable_styling(font_size = 10,latex_options = c("basic"))


#KO models sobre muestra test
train_vali_KO_30_1m<-
  train_KO_30_1m_target_feature %>% bind_rows(validation_KO_30_1m_target_feature)
      rownames(train_vali_KO_30_1m)<-c(rownames(train_KO_30_1m_target_feature),rownames(validation_KO_30_1m_target_feature))

train_vali_KO_60_1m<-
  train_KO_60_1m_target_feature %>% bind_rows(validation_KO_60_1m_target_feature)
      rownames(train_vali_KO_60_1m)<-c(rownames(train_KO_60_1m_target_feature),rownames(validation_KO_60_1m_target_feature))

train_vali_KO_90_1m<-
  train_KO_90_1m_target_feature %>% bind_rows(validation_KO_90_1m_target_feature)
       rownames(train_vali_KO_90_1m)<-c(rownames(train_KO_90_1m_target_feature),rownames(validation_KO_90_1m_target_feature))

train_vali_KO_fun_1m<-
  train_KO_fun_1m_target_feature %>% bind_rows(validation_KO_fun_1m_target_feature)
       rownames(train_vali_KO_fun_1m)<-c(rownames(train_KO_fun_1m_target_feature),rownames(validation_KO_fun_1m_target_feature)

       train_vali_KO_30_2m<-
  train_KO_30_2m_target_feature %>% bind_rows(validation_KO_30_2m_target_feature)
      rownames(train_vali_KO_30_2m)<-c(rownames(train_KO_30_2m_target_feature),rownames(validation_KO_30_2m_target_feature))

train_vali_KO_60_2m<-
  train_KO_60_2m_target_feature %>% bind_rows(validation_KO_60_2m_target_feature)
      rownames(train_vali_KO_60_2m)<-c(rownames(train_KO_60_2m_target_feature),rownames(validation_KO_60_2m_target_feature))

train_vali_KO_90_2m<-
  train_KO_90_2m_target_feature %>% bind_rows(validation_KO_90_2m_target_feature)
       rownames(train_vali_KO_90_2m)<-c(rownames(train_KO_90_2m_target_feature),rownames(validation_KO_90_2m_target_feature))

       train_vali_KO_fun_2m<-
  train_KO_fun_2m_target_feature %>% bind_rows(validation_KO_fun_2m_target_feature)
```

```
       rownames(train_vali_KO_fun_2m)<-c(rownames(train_KO_fun_2m_target_feature),rownames(validation_KO_fun_2m_target_feature))

       train_vali_KO_30_3m<-
  train_KO_30_3m_target_feature %>% bind_rows(validation_KO_30_3m_target_feature)
      rownames(train_vali_KO_30_3m)<-c(rownames(train_KO_30_3m_target_feature),rownames(validation_KO_30_3m_target_feature))

train_vali_KO_60_3m<-
  train_KO_60_3m_target_feature %>% bind_rows(validation_KO_60_3m_target_feature)
      rownames(train_vali_KO_60_3m)<-c(rownames(train_KO_60_3m_target_feature),rownames(validation_KO_60_3m_target_feature))

train_vali_KO_90_3m<-
  train_KO_90_3m_target_feature %>% bind_rows(validation_KO_90_3m_target_feature)
       rownames(train_vali_KO_90_3m)<-c(rownames(train_KO_90_3m_target_feature),rownames(validation_KO_90_3m_target_feature))

       train_vali_KO_fun_3m<-
  train_KO_fun_3m_target_feature %>% bind_rows(validation_KO_fun_3m_target_feature)
       rownames(train_vali_KO_fun_3m)<-c(rownames(train_KO_fun_3m_target_feature),rownames(validation_KO_fun_3m_target_feature))

model_KO_30_1m_target_feature<-randomForest(target~.,data=train_vali_KO_30_1m,mtry=13)
model_KO_60_1m_target_feature<-randomForest(target~.,data=train_vali_KO_60_1m,mtry=2)
model_KO_90_1m_target_feature<-randomForest(target~.,data=train_vali_KO_90_1m,mtry=2)
model_KO_fun_1m_target_feature<-randomForest(target~.,data=train_vali_KO_fun_1m,mtry=2)

model_KO_30_2m_target_feature<-randomForest(target~.,data=train_vali_KO_30_2m,mtry=2)
model_KO_60_2m_target_feature<-randomForest(target~.,data=train_vali_KO_60_2m,mtry=2)
model_KO_90_2m_target_feature<-randomForest(target~.,data=train_vali_KO_90_2m,mtry=4)
model_KO_fun_2m_target_feature<-randomForest(target~.,data=train_vali_KO_fun_2m,mtry=16)

model_KO_30_3m_target_feature<-randomForest(target~.,data=train_vali_KO_30_3m,mtry=2)
model_KO_60_3m_target_feature<-randomForest(target~.,data=train_vali_KO_60_3m,mtry=2)
model_KO_90_3m_target_feature<-randomForest(target~.,data=train_vali_KO_90_3m,mtry=2)
model_KO_fun_3m_target_feature<-randomForest(target~.,data=train_vali_KO_fun_3m,mtry=3)


#Confusion matrix muestra test
CM_KO_30_1m<-confusionMatrix(predict(model_KO_30_1m_target_feature,test_KO_30_1m_target_feature),
                             test_KO_30_1m_target_feature$target)
CM_KO_60_1m<-confusionMatrix(predict(model_KO_60_1m_target_feature,test_KO_60_1m_target_feature),
                             test_KO_60_1m_target_feature$target)
CM_KO_90_1m<-confusionMatrix(predict(model_KO_90_1m_target_feature,test_KO_90_1m_target_feature),
                             test_KO_90_1m_target_feature$target)
CM_KO_fun_1m<-confusionMatrix(predict(model_KO_fun_1m_target_feature,test_KO_fun_1m_target_feature),
                              test_KO_fun_1m_target_feature$target)


CM_KO_30_2m<-confusionMatrix(predict(model_KO_30_2m_target_feature,test_KO_30_2m_target_feature),
                             test_KO_30_2m_target_feature$target)
CM_KO_60_2m<-confusionMatrix(predict(model_KO_60_2m_target_feature,test_KO_60_2m_target_feature),
                             test_KO_60_2m_target_feature$target)
CM_KO_90_2m<-confusionMatrix(predict(model_KO_90_2m_target_feature,test_KO_90_2m_target_feature),
                             test_KO_90_2m_target_feature$target)
CM_KO_fun_2m<-confusionMatrix(predict(model_KO_fun_2m_target_feature,test_KO_fun_2m_target_feature),
                              test_KO_fun_2m_target_feature$target)

CM_KO_30_3m<-confusionMatrix(predict(model_KO_30_3m_target_feature,test_KO_30_3m_target_feature),
                             test_KO_30_3m_target_feature$target)
CM_KO_60_3m<-confusionMatrix(predict(model_KO_60_3m_target_feature,test_KO_60_3m_target_feature),
                             test_KO_60_3m_target_feature$target)
CM_KO_90_3m<-confusionMatrix(predict(model_KO_90_3m_target_feature,test_KO_90_3m_target_feature),
                             test_KO_90_3m_target_feature$target)
CM_KO_fun_3m<-confusionMatrix(predict(model_KO_fun_3m_target_feature,test_KO_fun_3m_target_feature),
                              test_KO_fun_3m_target_feature$target)


a<-data.frame(acc1m=as.numeric(c(round(CM_KO_30_1m$overall[1]*100,2),
                                 round(CM_KO_60_1m$overall[1]*100,2),
                                 round(CM_KO_90_1m$overall[1]*100,2),
```

```
                                round(CM_KO_fun_1m$overall[1]*100,2)),
              sensi1m=as.numeric(c(round(CM_KO_30_1m$byClass[1]*100,2),
                                   round(CM_KO_60_1m$byClass[1]*100,2),
                                   round(CM_KO_90_1m$byClass[1]*100,2),
                                   round(CM_KO_fun_1m$byClass[1]*100,2))),
              speci1m=as.numeric(c(round(CM_KO_30_1m$byClass[2]*100,2),
                                   round(CM_KO_60_1m$byClass[2]*100,2),
                                   round(CM_KO_90_1m$byClass[2]*100,2),
                                   round(CM_KO_fun_1m$byClass[2]*100,2))),

              acc2m=as.numeric(c(round(CM_KO_30_2m$overall[1]*100,2),
                                   round(CM_KO_60_2m$overall[1]*100,2),
                                  round(CM_KO_90_2m$overall[1]*100,2),
                                  round(CM_KO_fun_2m$overall[1]*100,2))),
              sensi2m=as.numeric(c(round(CM_KO_30_2m$byClass[1]*100,2),
                                   round(CM_KO_60_2m$byClass[1]*100,2),
                                   round(CM_KO_90_2m$byClass[1]*100,2),
                                   round(CM_KO_fun_2m$byClass[1]*100,2))),
              speci2m=as.numeric(c(round(CM_KO_30_2m$byClass[2]*100,2),
                                   round(CM_KO_60_2m$byClass[2]*100,2),
                                   round(CM_KO_90_2m$byClass[2]*100,2),
                                   round(CM_KO_fun_2m$byClass[2]*100,2))),

              acc3m=as.numeric(c(round(CM_KO_30_3m$overall[1]*100,2),
                                   round(CM_KO_60_3m$overall[1]*100,2),
                                  round(CM_KO_90_3m$overall[1]*100,2),
                                  round(CM_KO_fun_3m$overall[1]*100,2))),
              sensi3m=as.numeric(c(round(CM_KO_30_3m$byClass[1]*100,2),
                                   round(CM_KO_60_3m$byClass[1]*100,2),
                                   round(CM_KO_90_3m$byClass[1]*100,2),
                                   round(CM_KO_fun_3m$byClass[1]*100,2))),
              speci3m=as.numeric(c(round(CM_KO_30_3m$byClass[2]*100,2),
                                   round(CM_KO_60_3m$byClass[2]*100,2),
                                   round(CM_KO_90_3m$byClass[2]*100,2),
                                   round(CM_KO_fun_3m$byClass[2]*100,2)))



              )

rownames(a)<-c("EMA30","EMA60","EMA90","Alisado exponencial")

ConfusionKO<-a
SensiKO<-ConfusionKO[,grep(x=names(ConfusionKO), pattern="sensi")]
SpeciKO<-ConfusionKO[,grep(x=names(ConfusionKO), pattern="speci")]
ConfusionKO<-ConfusionKO[,grep(x=names(ConfusionKO), pattern="acc")]

kable(a, "latex") %>%
  add_header_above(c(" ", "Predicción 1 mes" = 3, "Predicción 2 meses" = 3,"Predicción 3 meses"=3)) %>%
  kable_styling(font_size = 10,latex_options = c("basic"))


#{Heatmap de la accuracy obtenida con los modelos Random Forest sobre muestra test}
acc.heatmap<-ConfusionKO %>% rbind(ConfusionAAPL)%>% rbind(ConfusionAXP)%>% rbind(ConfusionWFC)

rownames(acc.heatmap)<-apply(expand.grid(c("EMA30","EMA60","EMA90","smooth f"),
                                         c("KO", "AAPL", "AXP", "WFC")), 1, paste, collapse=".")

acc.heatmap$comb<-rownames(acc.heatmap);rownames(acc.heatmap)<-NULL;
acc.heatmap$comb<-rep(c("EMA30","EMA60","EMA90","smooth f"),4)

stock.heatmap <- acc.heatmap %>%
  select(comb,acc1m,acc2m,acc3m)%>%
  gather("forecast.window","test.acc", 2:4) %>%
  cbind(company=rep(c("KO","AAPL","AXP","WFC"),each=4),
        test.acc.label=as.character(.$test.acc)) %>%
```

```
  ggplot(mapping = aes(x = comb, y = forecast.window,fill = test.acc)) +
  geom_tile() +
  geom_text(aes(label=test.acc.label),size=2)+
  xlab(label = "Smooth type")+
  facet_grid(~ company, switch = "x", scales = "free_x", space = "free_x")+
scale_fill_gradient2('test.acc', low = "blue", mid = "white", high = "red", midpoint = 50)+
theme_bw()+
  theme(
    axis.text.x = element_text(angle=90, hjust = 0.5 )
  )

stock.heatmap

#{Heatmap de la sensitividad obtenida con los modelos Random Forest sobre muestra test}
sensi.heatmap<-SensiKO %>% rbind(SensiAAPL)%>% rbind(SensiAAPL)%>% rbind(SensiWFC)

rownames(sensi.heatmap)<-apply(expand.grid(c("EMA30","EMA60","EMA90","smooth f"),
                                           c("KO", "AAPL", "AXP", "WFC")), 1, paste, collapse=".")

sensi.heatmap$comb<-rownames(sensi.heatmap);rownames(sensi.heatmap)<-NULL;
sensi.heatmap$comb<-rep(c("EMA30","EMA60","EMA90","smooth f"),4)

stock.heatmap.sensi <- sensi.heatmap %>%
  select(comb,sensi1m,sensi2m,sensi3m)%>%
  gather("forecast.window","test.sensi", 2:4) %>%
  cbind(company=rep(c("KO","AAPL","AXP","WFC"),each=4),
        test.sensi.label=as.character(.$test.sensi)) %>%

  ggplot(mapping = aes(x = comb, y = forecast.window,fill = test.sensi)) +
  geom_tile() +
  geom_text(aes(label=test.sensi.label),size=2)+
  xlab(label = "Smooth type")+
  facet_grid(~ company, switch = "x", scales = "free_x", space = "free_x")+
scale_fill_gradient2('test.sensi', low = "blue", mid = "white", high = "red", midpoint = 50)+
theme_bw()+
  theme(
    axis.text.x = element_text(angle=90, hjust = 0.5 )
  )

stock.heatmap.sensi

#heatmap specificity
speci.heatmap<-SpeciKO %>% rbind(SpeciAAPL)%>% rbind(SpeciAAPL)%>% rbind(SpeciWFC)

rownames(speci.heatmap)<-apply(expand.grid(c("EMA30","EMA60","EMA90","smooth f"),
                                           c("KO", "AAPL", "AXP", "WFC")), 1, paste, collapse=".")

speci.heatmap$comb<-rownames(speci.heatmap);rownames(speci.heatmap)<-NULL;
speci.heatmap$comb<-rep(c("EMA30","EMA60","EMA90","smooth f"),4)

stock.heatmap.speci <- speci.heatmap %>%
  select(comb,speci1m,speci2m,speci3m)%>%
  gather("forecast.window","test.speci", 2:4) %>%
  cbind(company=rep(c("KO","AAPL","AXP","WFC"),each=4),
        test.speci.label=as.character(.$test.speci)) %>%

  ggplot(mapping = aes(x = comb, y = forecast.window,fill = test.speci)) +
  geom_tile() +
  geom_text(aes(label=test.speci.label),size=2)+
  xlab(label = "Smooth type")+
  facet_grid(~ company, switch = "x", scales = "free_x", space = "free_x")+
scale_fill_gradient2('test.speci', low = "blue", mid = "white", high = "red", midpoint = 50)+
theme_bw()+
  theme(
    axis.text.x = element_text(angle=90, hjust = 0.5 )
  )
```

```
stock.heatmap.speci
```

Apartado V.2

```
source("C:/Users/i0386388/Desktop/TFG-master/functions/smooth_target.R")

source("C:/Users/i0386388/Desktop/tesis/Tesis/target_feature_extraction.R")

# Fetch all Symbols & store only the tickers to retrieve the data
symbols <- stockSymbols(exchange = "NYSE")
symbols <- symbols[,1]
#
symbols<-symbols[1000:1300]
n <- length(symbols)
pb <- txtProgressBar(min = 0, max = n, style=3)


library(doParallel)

#setup parallel backend to use many processors
cores=detectCores()
cl <- makeCluster(cores[1]-1) #not to overload your computer
registerDoParallel(cl)

# Actual loop:
#for(i in 1:length(symbols)) {
  time<-Sys.time()
results= foreach::foreach(i=1:length(symbols),
                          .export = c("symbols","n"),#ls()
                          .combine = rbind,
                          .packages = c("quantmod","TTR","tidyverse","caret","randomForest","CombMSC"),
                          .verbose = TRUE)%dopar%{

#   cat(paste0("Doing ",symbols[i],"\n",length(symbols)-i, "remaining"))

  symbols[i]-> symbol
  # specify the "from" date to desired start date
  tryit <- try(getSymbols(symbol,from="2000-01-01", src='yahoo'))
  if(inherits(tryit, "try-error")){
    i <- i+1
  } else {
    # specify the "from" date to desired start date
    #data <-
      getSymbols(symbol, from="2000-01-01", src='yahoo')
    #dataset <- merge(dataset, Ad(get(symbols[i])))
    rm(symbol)

  # setTxtProgressBar(pb, i)


  if(nrow(get(symbols[i]))>2000){
  data<-try(EMA_finance(get(symbols[i]),smoothing_period = 90),silent = T)

  if(class(data)!="try-error"){


  data_1m<-data %>%
    slice(1:(nrow(.)-20)) %>%
    bind_cols(target=target_calc(data,20))

  rownames(data_1m)<-rownames(data)[c(1:(nrow(data)-20))]
  data_1m$target<-factor(data_1m$target)

  data_2m<-data %>%
    slice(1:(nrow(.)-40)) %>%
```

```
   bind_cols(target=target_calc(data,40))

rownames(data_2m)<-rownames(data)[c(1:(nrow(data)-40))]
data_2m$target<-factor(data_2m$target)

data_3m<-data %>%
  slice(1:(nrow(.)-60)) %>%
  bind_cols(target=target_calc(data,60))

rownames(data_3m)<-rownames(data)[c(1:(nrow(data)-60))]
data_3m$target<-factor(data_3m$target)

rm(data)

 data_1m_target_feature<-feature_extraction_finance(data_1m)
 data_2m_target_feature<-feature_extraction_finance(data_2m)
 data_3m_target_feature<-feature_extraction_finance(data_3m)

 rm(data_1m,data_2m,data_3m)

 train_data_1m_target_feature<-
   data_1m_target_feature[1:floor(nrow(data_1m_target_feature)*0.85),]
 test_data_1m_target_feature<-
   data_1m_target_feature[(floor(nrow(data_1m_target_feature)*0.85)+1):nrow(data_1m_target_feature),]

 train_data_2m_target_feature<-
   data_2m_target_feature[1:floor(nrow(data_2m_target_feature)*0.85),]
 test_data_2m_target_feature<-
   data_2m_target_feature[(floor(nrow(data_2m_target_feature)*0.85)+1):nrow(data_2m_target_feature),]

 train_data_3m_target_feature<-
   data_3m_target_feature[1:floor(nrow(data_3m_target_feature)*0.85),]
 test_data_3m_target_feature<-
   data_3m_target_feature[(floor(nrow(data_3m_target_feature)*0.85)+1):nrow(data_3m_target_feature),]

 #remover las filas donde el pn ratio sea infinito5
 if(any(train_data_1m_target_feature$PNratio=="Inf")){
   RF_1m<-randomForest(target~.,data=train_data_1m_target_feature %>% select(-PNratio),ntree=1500)
 }else{
   RF_1m<-randomForest(target~.,data=train_data_1m_target_feature,ntree=1500)
 }

 if(any(train_data_2m_target_feature$PNratio=="Inf")){
 RF_2m<-randomForest(target~.,data=train_data_2m_target_feature%>% select(-PNratio),ntree=1500)
 }else{
   RF_2m<-randomForest(target~.,data=train_data_2m_target_feature,ntree=1500)

 }

 if(any(train_data_2m_target_feature$PNratio=="Inf")){
   RF_3m<-randomForest(target~.,data=train_data_3m_target_feature%>% select(-PNratio),ntree=1500)

 }else{
   RF_3m<-randomForest(target~.,data=train_data_3m_target_feature,ntree=1500)

 }

 if(symbols[i]=="AAPL"){summary(RF_3m)}
 #poner aqui todas las empresas
 CM_data_1m<-confusionMatrix(predict(RF_1m,test_data_1m_target_feature),test_data_1m_target_feature$target)
 CM_data_2m<-confusionMatrix(predict(RF_2m,test_data_2m_target_feature),test_data_2m_target_feature$target)
 CM_data_3m<-confusionMatrix(predict(RF_3m,test_data_3m_target_feature),test_data_3m_target_feature$target)

 rm(RF_1m,RF_2m,RF_3m)

 a<-data.frame(acc1m=as.numeric(c(round(CM_data_1m$overall[1]*100,2))),
               sensi1m=as.numeric(c(round(CM_data_1m$byClass[1]*100,2))),
```

```
                 speci1m=as.numeric(c(round(CM_data_1m$byClass[2]*100,2))),

                 acc2m=as.numeric(c(round(CM_data_2m$overall[1]*100,2))),
                 sensi2m=as.numeric(c(round(CM_data_2m$byClass[1]*100,2))),
                 speci2m=as.numeric(c(round(CM_data_2m$byClass[2]*100,2))),

                 acc3m=as.numeric(c(round(CM_data_3m$overall[1]*100,2))),
                 sensi3m=as.numeric(c(round(CM_data_3m$byClass[1]*100,2))),
                 speci3m=as.numeric(c(round(CM_data_3m$byClass[2]*100,2)))
                )


   b<-a[,grep(x=names(a), pattern="acc")] %>% mutate(stock=symbols[i])
   rm(a)
   b
   # if(i==1){
   #   heat.map.data<-a[,grep(x=names(a), pattern="acc")] %>% mutate(stock=symbols[i])
   #   cat("First done\n")
   #
   # }else{
   #   temp<-a[,grep(x=names(a), pattern="acc")] %>% mutate(stock=symbols[i])
   #   heat.map.data<-heat.map.data %>% rbind(temp)
   # }

    }

    }else{  rm(list=symbols[i])}#end of error in EMA
  }#end of second (try) if
} #end of loop

stopCluster(cl)
time



##heatmap
#{Heatmap de la accuracy obtenida sobre muestra test SMD masivo para 165 empresas. Fuente: elaboración propia}

results %>%
  gather("forecast.window","test.acc", 1:3) %>%
  arrange(stock) %>%
  cbind(test.acc.label=as.character(.$test.acc)) %>%
  ggplot(mapping = aes(x = stock, y = forecast.window,fill = test.acc)) +
  geom_tile() +
  #geom_text(aes(label=test.acc.label),size=2)+
  xlab(label = "Companies")+
  #facet_grid(~ company, switch = "x", scales = "free_x", space = "free_x")+
  scale_fill_gradient2('test.acc', low = "blue", mid = "white", high = "red", midpoint = 50)+
  theme_bw()+
  theme(
    axis.text.x = element_text(angle=90,vjust = 1,hjust = 0.5,size=7),
    axis.ticks.length = unit(10, "pt"),
    legend.position="bottom")+
  scale_y_discrete(expand=c(0,0))


#{Precio de cierre empresa Ferrellgas (FGP). Fuente: elaboración propia}

print(results[results$stock=="FGP",])

library(quantmod)
getSymbols("FGP",from="2000-01-01")

data.FGP<-try(EMA_finance(FGP,smoothing_period = 90),silent = T)
add_rownames(as.data.frame(data.FGP),"rownames") %>%
  mutate(rownames=as.Date(rownames,"%Y-%m-%d")) %>%
  ggplot(aes(x=rownames,y=Close))+
```

```
  geom_line() +
  theme_bw()+
  geom_vline(xintercept = as.Date("2016-06-21"),col="red",size=1.5)+
  scale_x_date(expand=c(0,0))+
  ggtitle("Stock FGP partición de datos entrenamiento - prueba. Datos alisados con una EMA 90 días")+
  xlab("")


#{Distribución de los resultados de accuracy sobre muestra test
#en las 3 ventanas temporales consideradas. Fuente: elaboración propia}
library(reshape2)
data<- melt(results[,-4])


ggplot(data,aes(x=value, fill=variable)) +
  geom_density(alpha=0.25)+
  theme_bw()+
  ggtitle("Accuracy sobre muestra test")+
  xlab("Accuracy")+
  theme(legend.position="bottom")+
  scale_x_continuous(expand=c(0,0))


library(fBasics)
basicStats(results[,-4])[c("Mean","Variance","Skewness","Kurtosis"),]

results %>% select(-stock) %>% summary

#{Top 10 empresas con mejor rendimiento sobre muestra test para cada ventana de predicción.}
library(kableExtra)
a<-dplyr::top_n(results,n = 10,wt = acc1m) %>% arrange(desc(acc1m)) %>% select(acc1m,stock) %>%
  cbind(dplyr::top_n(results,n = 10,wt = acc2m) %>% arrange(desc(acc2m))%>% select(acc2m,stock),
        dplyr::top_n(results,n = 10,wt = acc3m) %>% arrange(desc(acc3m))%>% select(acc3m,stock))

kable(a, "latex") %>%
  add_header_above(c("Predicción 1 mes" = 2, "Predicción 2 meses" = 2,"Predicción 3 meses"=2)) %>%
 kable_styling(font_size = 10,latex_options = c("basic"))
```

Apartado V.3

```
#
library(h2o)
h2o.init()
h2o.no_progress()  # Turn off progress bars for notebook readability
#


h2o.SMD<-function(train,test,company){
  library(tidyverse)
  train<-as.h2o(train %>% select(-c(aroonUp,aroonDn)))
  test<-as.h2o(test%>% select(-c(aroonUp,aroonDn)))

  y <- "target"
  x <- setdiff(names(train), c(y,"aroonUp","aroonDn"))

  aml <<- h2o.automl(y = y, x = x,
                     training_frame = train,
                     max_models =10,
                     seed = 420,
                     nfolds = 0, #no cross validation
                     balance_classes = F,
                     leaderboard_frame = test)

  se <<- h2o.getModel(as.data.frame(aml@leaderboard$model_id)[1,1])
```

```
  a<-data.frame(Company=company,
                AUC=h2o.auc(h2o.performance(se, newdata = test)),
                Accuracy=h2o.accuracy(h2o.performance(se, newdata = test),thresholds = 0.5)[[1]],
                Model=as.data.frame(aml@leaderboard$model_id)[1,1])
  return(a)
  #accuracy

}


h2o_AXP_30_1m<-h2o.SMD(train = train_vali_AXP_30_1m,test = test_AXP_30_1m_target_feature,company = "AXP")
h2o_KO_30_1m<-h2o.SMD(train = train_vali_KO_30_1m,test = test_KO_30_1m_target_feature,company = "KO_30_1m")
h2o_KO_30_2m<-h2o.SMD(train = train_vali_KO_30_2m,test = test_KO_30_2m_target_feature,company = "KO_30_2m")
h2o_KO_30_3m<-h2o.SMD(train = train_vali_KO_30_3m,test = test_KO_30_3m_target_feature,company = "KO_30_3m")

h2o_KO_60_1m<-h2o.SMD(train = train_vali_KO_60_1m,test = test_KO_60_1m_target_feature,company = "KO_60_1m")
h2o_KO_60_2m<-h2o.SMD(train = train_vali_KO_60_2m,test = test_KO_60_2m_target_feature,company = "KO_60_2m")
h2o_KO_60_3m<-h2o.SMD(train = train_vali_KO_60_3m,test = test_KO_60_3m_target_feature,company = "KO_60_3m")

h2o_KO_90_1m<-h2o.SMD(train = train_vali_KO_90_1m,test = test_KO_90_1m_target_feature,company = "KO_90_1m")
h2o_KO_90_2m<-h2o.SMD(train = train_vali_KO_90_2m,test = test_KO_90_2m_target_feature,company = "KO_90_2m")
h2o_KO_90_3m<-h2o.SMD(train = train_vali_KO_90_3m,test = test_KO_90_3m_target_feature,company = "KO_90_3m")

h2o.SMD.results<-h2o_KO_30_1m %>%
  rbind(h2o_KO_30_2m,h2o_KO_30_3m,h2o_KO_60_1m,h2o_KO_60_2m,h2o_KO_60_3m,h2o_KO_90_1m,h2o_KO_90_2m,h2o_KO_90_3m)
```

Apartado V.4

```
libraries <- c("lubridate","tidyverse","tidyr","forecast","ggplot2","seasonal","tidyverse",
               "manipulate","compiler","scales","lmtest","MASS","glmnet","forecTheta",
               "neuralnet","tsDyn","changepoint","RSNNS","xgboost","foreach","doSNOW",
               "tcltk","DescTools","rnn","glue","forcats","timetk","tidyquant","tibbletime",
               "cowplot","recipes","rsample","yardstick","keras","quantmod","tidyverse"
)
# check.libraries <- is.element(libraries, installed.packages()[, 1])==FALSE
# libraries.to.install <- libraries[check.libraries]
# if (length(libraries.to.install!=0)) {
#   install.packages(libraries.to.install)
# }

lapply(libraries, require, character.only=TRUE)
library(tidyverse)
library(quantmod)
getSymbols(Symbols = "KO", from ="2000-01-01",to="2018-12-31")
#install_keras()

load("C:/Users/i0386388/Desktop/tesis/lstm_prices.RData")
sample_predictions_lstm_tbl_prices<-sample_predictions_lstm_tbl
load("C:/Users/i0386388/Desktop/tesis/lstm_return.RData")
sample_predictions_lstm_tbl_return<-sample_predictions_lstm_tbl
rm(sample_predictions_lstm_tbl)



dates<-rownames(as.data.frame(KO))

Exerci.0<-data.frame(MSales=as.vector(KO$KO.Close),StartDate=dates)#Close prices

rownames(Exerci.0)<-Exerci.0$StartDate
input<-Exerci.0 %>% dplyr::select(MSales)


input.0 <- input %>%
  tk_tbl() %>%
  mutate(index = as_date(index)) %>%
```

```
  as_tbl_time(index = index) %>%
  rename("value"=MSales)


periods_train <- 996 #train length in each resample
periods_test  <- 83
skip_span     <- 650

rolling_origin_resamples <- rolling_origin(
  input.0,
  initial    = periods_train,
  assess     = periods_test,
  cumulative = FALSE,
  skip       = skip_span
)

rolling_origin_resamples


# Plotting function for a single split
plot_split <- function(split, expand_y_axis = TRUE, alpha = 1, size = 1, base_size = 14) {

  # Manipulate data
  train_tbl <- training(split) %>%
    add_column(key = "training")

  test_tbl  <- testing(split) %>%
    add_column(key = "testing")

  data_manipulated <- bind_rows(train_tbl, test_tbl) %>%
    as_tbl_time(index = index) %>%
    mutate(key = fct_relevel(key, "training", "testing"))

  # Collect attributes
  train_time_summary <- train_tbl %>%
    tk_index() %>%
    tk_get_timeseries_summary()

  test_time_summary <- test_tbl %>%
    tk_index() %>%
    tk_get_timeseries_summary()

  # Visualize
  g <- data_manipulated %>%
    ggplot(aes(x = index, y = value, color = key, group = 1)) +
    geom_line(size = size, alpha = alpha) +
    theme_tq(base_size = base_size) +
    scale_color_tq() +
    labs(
      title    = glue("Split: {split$id}"),
      subtitle = glue("{train_time_summary$start} to {test_time_summary$end}"),
      y = "", x = ""
    ) +
    theme(legend.position = "none")

  if (expand_y_axis) {

    input.0_time_summary <- input.0 %>%
      tk_index() %>%
      tk_get_timeseries_summary()

    g <- g +
      scale_x_date(limits = c(input.0_time_summary$start,
                              input.0_time_summary$end))
  }

  return(g)
```

```
}


plot_sampling_plan <- function(sampling_tbl, expand_y_axis = TRUE,
                               ncol = 3, alpha = 1, size = 1, base_size = 14,
                               title = "Sampling Plan") {

  # Map plot_split() to sampling_tbl
  sampling_tbl_with_plots <- sampling_tbl %>%
    mutate(gg_plots = map(splits, plot_split,
                          expand_y_axis = expand_y_axis,
                          alpha = alpha, base_size = base_size))

  # Make plots with cowplot
  plot_list <- sampling_tbl_with_plots$gg_plots

  p_temp <- plot_list[[1]] + theme(legend.position = "bottom")
  legend <- get_legend(p_temp)

  p_body  <- plot_grid(plotlist = plot_list, ncol = ncol)

  p_title <- ggdraw() +
    draw_label(title, size = 18, fontface = "bold", colour = palette_light()[[1]])

  g <- plot_grid(p_title, p_body, legend, ncol = 1, rel_heights = c(0.05, 1, 0.05))

  return(g)

}

rolling_origin_resamples %>%
  plot_sampling_plan(expand_y_axis = T, ncol = 3, alpha = 1, size = 1, base_size = 10,
                     title = "Ventana móbil de muestras de entrenamiento y prueba")

rolling_origin_resamples %>%
  plot_sampling_plan(expand_y_axis = F, ncol = 3, alpha = 1, size = 1, base_size = 10,
                     title = "Ventana móbil de muestras de entrenamiento y prueba. Ampliado")


#Creating function to train lstm over each split

predict_keras_lstm <- function(split, epochs, ...) {

  lstm_prediction <- function(split, epochs, ...) {

    # 5.1.2 Data Setup
    df_trn <- training(split)
    df_tst <- testing(split)

    df <- bind_rows(
      df_trn %>% add_column(key = "training"),
      df_tst %>% add_column(key = "testing")
    ) %>%
      as_tbl_time(index = index)

    # 5.1.3 Preprocessing
    rec_obj <- recipe(value ~ ., df) %>%
      step_sqrt(value) %>%
      step_center(value) %>%
      step_scale(value) %>%
      prep()

    df_processed_tbl <- bake(rec_obj, df)

    center_history <- rec_obj$steps[[2]]$means["value"]
    scale_history  <- rec_obj$steps[[3]]$sds["value"]
```

```
# 5.1.4 LSTM Plan
# lag_setting  <- 955 # = nrow(df_test) = periods_test
# batch_size   <- 5
# train_length <- 3820
# tsteps       <- 1
# epochs       <- 150

#training length / testing length must be a whole number
#training length / batch size and testing length / batch size must both be whole numbers


lag_setting  <- 83 # = nrow(df_test) = periods_test
batch_size   <- 1 #2 gives us more or less good results with AZN
train_length <- 996
tsteps       <- 6  #because we are only using one lag
epochs       <-epochs



# 5.1.5 Train/Test Setup
lag_train_tbl <- df_processed_tbl %>%
  mutate(value_lag = lag(value, n = lag_setting)) %>%
  filter(!is.na(value_lag)) %>%
  filter(key == "training") %>%
  tail(train_length)

x_train_vec <- lag_train_tbl$value_lag
x_train_arr <- array(data = x_train_vec, dim = c(length(x_train_vec), tsteps, 1))

y_train_vec <- lag_train_tbl$value
y_train_arr <- array(data = y_train_vec, dim = c(length(y_train_vec), 1))

lag_test_tbl <- df_processed_tbl %>%
  mutate(
    value_lag = lag(value, n = lag_setting)
  ) %>%
  filter(!is.na(value_lag)) %>%
  filter(key == "testing")

x_test_vec <- lag_test_tbl$value_lag
x_test_arr <- array(data = x_test_vec, dim = c(length(x_test_vec), tsteps, 1))

y_test_vec <- lag_test_tbl$value
y_test_arr <- array(data = y_test_vec, dim = c(length(y_test_vec), 1))

# 5.1.6 LSTM Model
model <- keras_model_sequential()

model %>%   #with units=300 is working well
  layer_lstm(units            = 100,
             input_shape      = c(tsteps, 1),
             batch_size       = batch_size,
             return_sequences = TRUE,
             unit_forget_bias = 1,
             stateful         = TRUE) %>%
  layer_lstm(units            = 100,
             return_sequences = FALSE,
             unit_forget_bias = 1,
             stateful         = TRUE) %>%
  layer_dense(units = 1)

model %>%
  compile(loss = 'mae', optimizer = 'adam')

# 5.1.7 Fitting LSTM
for (i in 1:epochs) {
  model %>% fit(x          = x_train_arr,
```

```
                    y          = y_train_arr,
                    batch_size = batch_size,
                    epochs     = 1,
                    verbose    = 1,
                    shuffle    = FALSE)

      model %>% reset_states()
      cat("Epoch: ", i)

    }

    # 5.1.8 Predict and Return Tidy Data
    # Make Predictions
    pred_out <- model %>%
      predict(x_test_arr, batch_size = batch_size) %>%
      .[,1]

    # Retransform values
    pred_tbl <- tibble(
      index   = lag_test_tbl$index,
      value   = (pred_out * scale_history + center_history)^2
    )

    # Combine actual data with predictions
    tbl_1 <- df_trn %>%
      add_column(key = "actual")

    tbl_2 <- df_tst %>%
      add_column(key = "actual")

    tbl_3 <- pred_tbl %>%
      add_column(key = "predict")

    # Create time_bind_rows() to solve dplyr issue
    time_bind_rows <- function(data_1, data_2, index) {
      index_expr <- enquo(index)
      bind_rows(data_1, data_2) %>%
        as_tbl_time(index = !! index_expr)
    }

    ret <- list(tbl_1, tbl_2, tbl_3) %>%
      reduce(time_bind_rows, index = index) %>%
      arrange(key, index) %>%
      mutate(key = as_factor(key))

    return(ret)

  }

  safe_lstm <- possibly(lstm_prediction, otherwise = NA)

  safe_lstm(split, epochs, ...)

}

predictions_lstm<-predict_keras_lstm(split)

#training the model and calculating forecasts====
sample_predictions_lstm_tbl <- rolling_origin_resamples %>%
  mutate(predict = map(splits, predict_keras_lstm, epochs = 100))

sample_predictions_lstm_tbl


calc_mape <- function(prediction_tbl) {

  mape_calculation <- function(data) {
```

```
    data %>%
      spread(key = key, value = value) %>%
      dplyr::select(-index) %>%
      filter(!is.na(predict)) %>%
      rename(
        truth    = actual,
        estimate = predict
      ) %>%
      filter(truth!=0) %>%
      mutate(diff=abs(truth-estimate)) %>%
      mutate(coc=diff/truth) %>%
      summarise(mape=mean(coc))*100
  }

  safe_mape <- possibly(mape_calculation, otherwise = NA)

  safe_mape(prediction_tbl)

}

calc_mape_sfi <- function(prediction_tbl) {

    mape_calculation_sfi <- function(data) {
    data %>%
      spread(key = key, value = value) %>%
      dplyr::select(-index) %>%
      filter(!is.na(predict)) %>%
      rename(
        truth    = actual,
        estimate = predict
      ) %>%
      filter(truth!=0) %>%
      summarise(diff=sum(abs(truth-estimate)),
                total=sum(truth)) %>%
      mutate(mape=(diff/total)*100)
  }

  safe_mape1 <- possibly(mape_calculation_sfi, otherwise = NA)

  safe_mape1(prediction_tbl)
}



# Setup single plot function

plot_prediction <- function(data, id, alpha = 1, size = 2, base_size = 6) {

  #mape_val <- calc_mape(data)

  g <- data %>%
    ggplot(aes(index, value, color = key)) +
    geom_point(alpha = alpha, size = size) +
    #geom_line(size = size) +
    tidyquant::theme_tq(base_size = base_size) +
    tidyquant::scale_color_tq() +
    theme(legend.position = "none") +
    labs(
      #title = glue("{id}, mape: {round(mape_val, digits = 1)}"),
      x = "", y = ""
    )

  return(g)
}
```

```
plot_predictions <- function(sampling_tbl, predictions_col,
                             ncol = 2, alpha = 1, size = 2, base_size = 18,
                             title = "Backtested Predictions") {

  predictions_col_expr <- enquo(predictions_col)

  # Map plot_split() to sampling_tbl
  sampling_tbl_with_plots <- sampling_tbl %>%
    mutate(gg_plots = map2(!! predictions_col_expr, id,
                           .f        = plot_prediction,
                           alpha     = alpha,
                           size      = size,
                           base_size = base_size))

  # Make plots with cowplot
  plot_list <- sampling_tbl_with_plots$gg_plots

  p_temp <- plot_list[[1]] + theme(legend.position = "bottom")
  legend <- get_legend(p_temp)

  p_body  <- plot_grid(plotlist = plot_list, ncol = ncol)



  p_title <- ggdraw() +
    draw_label(title, size = 18, fontface = "bold", colour = palette_light()[[1]])

  g <- plot_grid(p_title, p_body, legend, ncol = 1, rel_heights = c(0.05, 1, 0.05))

  return(g)

}


sample_predictions_lstm_tbl_prices %>%
  filter(id%in%c("Slice1", "Slice2")) %>%
  plot_predictions(predictions_col = predict, alpha = 1, size = 1, base_size = 10,
                   title = "Predicción sobre las muestras de prueba. Particiones 1 y 2")

sample_predictions_lstm_tbl_prices %>%
  filter(id%in%c("Slice3","Slice4")) %>%
  plot_predictions(predictions_col = predict, alpha = 1, size = 1, base_size = 10,
                   title = "Predicción sobre las muestras de prueba. Particiones 3 y 4")

sample_predictions_lstm_tbl_prices %>%
  filter(id%in%c( "Slice5","Slice6")) %>%
  plot_predictions(predictions_col = predict, alpha = 1, size = 1, base_size = 10,
                   title = "Predicción sobre las muestras de prueba. Particiones 5 y 6")



sample_mape_tbl <- sample_predictions_lstm_tbl_prices %>%
  mutate(mape = map(predict, calc_mape)) %>%
  dplyr::select(id, mape)


#sample_predictions_lstm_tbl_prices$predict[[1]]
# for(i in 1:length(sample_predictions_lstm_tbl_prices$predict)){
#   temp<-calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[i]])
#   if(base::exists("results")){
#     results<-results %>% rbind(temp)
#   }else{results<-temp}
# }

results<-calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[1]]) %>%
  rbind(calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[2]]),
        calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[3]]),
```

```
        calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[4]]),
        calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[5]]),
        calc_mape_sfi(sample_predictions_lstm_tbl_prices$predict[[6]]))

do.call(rbind.data.frame, sample_mape_tbl$mape) %>% mutate(Split=paste("split",seq(1,6,1))) %>%
  left_join(
    results %>% mutate(key=paste("split",seq(1,6,1))),
    by=c("Split"="key")
  ) %>%
  dplyr::rename("MAPE"=mape.x,"MAPE2"=mape.y) %>%
  dplyr::select(Split,MAPE,MAPE2)


#MAPE 1 and 2 plots
do.call(rbind.data.frame, sample_mape_tbl$mape) %>% mutate(Split=paste("split",seq(1,6,1))) %>%
  left_join(
    results %>% mutate(key=paste("split",seq(1,6,1))),
    by=c("Split"="key")
  ) %>%
  dplyr::rename("MAPE"=mape.x,"MAPE2"=mape.y) %>%
  dplyr::select(Split,MAPE,MAPE2) %>%
  gather(metrica,valor,2:3) %>%
  ggplot(aes(x=Split,y=valor,fill=metrica),color="black")+
  geom_bar(stat = "identity",position=position_dodge())+
  theme_tq()+
  scale_fill_brewer(palette="Set1")+
  theme(legend.text  = element_text(size=15),
        legend.title = element_text(size=15))+
  ggtitle("Métricas sobre las distintas particiones obtenidas con la LSTM")


#MAE and RMSE calculation

metrics<-function(x){

a<-x %>%
    spread(key = key, value = value) %>%
    dplyr::select(-index) %>%
    filter(!is.na(predict)) %>%
    rename(
        truth    = actual,
        estimate = predict
    ) %>%
    filter(truth!=0)

return(data.frame(MAE=MAE(x = a$estimate,ref=a$truth),
                  RMSE=RMSE(x = a$estimate,ref=a$truth)
        ))

}




metrics(sample_predictions_lstm_tbl_prices$predict[[1]]) %>%
  rbind(metrics(sample_predictions_lstm_tbl_prices$predict[[2]]),
        metrics(sample_predictions_lstm_tbl_prices$predict[[3]]),
        metrics(sample_predictions_lstm_tbl_prices$predict[[4]]),
        metrics(sample_predictions_lstm_tbl_prices$predict[[5]]),
        metrics(sample_predictions_lstm_tbl_prices$predict[[6]])) %>%
  mutate(particiones=paste("split",seq(1,6,1))) %>%
  dplyr::select(particiones,everything()) %>%
  arrange(RMSE)
```

Apartado V.5

El código utilizado en el apartado V.5 es el mismo que el utilizado en el apartado V.5, con el cambio en los datos de entrada. En este apartado se aplica la LSTM sobre la rendibilidad en vez de sobre el precio de cierre. Para simplificar el anexo no se vuelve a copiar el mismo código que en el apartado V.4 ya que es exactamente el mismo que el utilizado en el apartado V.5

```
libraries <- c("lubridate", "tidyverse", "tidyr", "forecast",
    "ggplot2", "seasonal", "tidyverse", "manipulate", "compiler",
    "scales", "lmtest", "MASS", "glmnet", "forecTheta", "neuralnet",
    "tsDyn", "changepoint", "RSNNS", "xgboost", "foreach", "doSNOW",
    "tcltk", "DescTools", "rnn", "glue", "forcats", "timetk",
    "tidyquant", "tibbletime", "cowplot", "recipes", "rsample",
    "yardstick", "keras", "quantmod", "tidyverse")
# check.libraries <- is.element(libraries,
# installed.packages()[, 1])==FALSE libraries.to.install <-
# libraries[check.libraries] if
# (length(libraries.to.install!=0)) {
# install.packages(libraries.to.install) }

lapply(libraries, require, character.only = TRUE)
library(tidyverse)
library(quantmod)
getSymbols(Symbols = "KO", from = "2000-01-01", to = "2018-12-31")
# install_keras()

load("C:/Users/i0386388/Desktop/tesis/lstm_prices.RData")
sample_predictions_lstm_tbl_prices <- sample_predictions_lstm_tbl
load("C:/Users/i0386388/Desktop/tesis/lstm_return.RData")
sample_predictions_lstm_tbl_return <- sample_predictions_lstm_tbl
rm(sample_predictions_lstm_tbl)



dates <- rownames(as.data.frame(KO))
log_return <- diff(log(as.vector(KO$KO.Close)))

Exerci.0 <- data.frame(MSales = log_return, StartDate = dates[-length(dates)])  #log returns

rownames(Exerci.0) <- Exerci.0$StartDate
input <- Exerci.0 %>% dplyr::select(MSales)
```