UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Elèctrica

PREDICCIÓ DE LA DEMANDA ELÈCTRICA A L'EEBE

Memòria i Annexos

Autor: Joan Manuel Márquez Fernández
Director: Roberto Villafáfila Robles

Convocatòria: Gener 2021

Resum

Aquest projecte consisteix en el desenvolupament d'un model predictiu de la demanda d'energia elèctrica del campus EEBE, mitjançant eines d'intel·ligència artificial i emprant un històric de dades registrades pels mateixos comptadors d'energia elèctrica de l'escola.

Com a punt de partida, es realitza una exposició del marc teòric que enquadra el treball, sent els temes principals el monitoratge del consum elèctric i els mitjans per realitzar-lo, l'anàlisi de dades per l'optimització de la gestió energètica, i la implementació d'eines d'intel·ligència artificial en la predicció de la demanda elèctrica.

Posteriorment, es passa a l'anàlisi del cas concret, la demanda elèctrica del campus EEBE, pel qual es fa una primera introducció a les dades amb les quals es treballarà, informant del tractament previ necessari i les primeres conclusions que se n'extreuen en referència al seu comportament. I a continuació, s'exposa la metodologia emprada per poder obtenir els models que permetin assolir l'objectiu cercat. Aquests models d'intel·ligència artificial s'obtenen mitjançant les llibreries de SciKit Learn, corresponents al llenguatge Python, i es treballa en l'entorn conegut com Jupyter Notebook.

Per finalitzar el projecte, es presenten les conclusions extretes i els models que permeten obtenir una millor aproximació a la variable incògnita, la demanda d'energia elèctrica.

Resumen

Este proyecto consiste en el desarrollo de un modelo de predicción de la demanda de energia eléctrica del campus EEBE, mediante herramientas de inteligencia artificial y un histórico de los datos registrados por los contadores de consumo eléctrico del centro.

Inicialmente, se realiza una exposición del marco teórico que enmarca el trabajo, siendo los temas principales la monitorización del consumo eléctrico y los medios para su realización, el análisis de datos para la optimización de la gestión energética, y la implementación de la inteligencia artificial en la predicción de la demanda eléctrica.

Tras esto, se pasa al análisis del caso particular, la demanda eléctrica del campus EEBE, para el que se realiza una primera introducción a los datos sobre los que se trabajara, informando del tratamiento previo necesario y las primeras conclusiones que se extraen respecto a su comportamiento. A continuación, se expone la metodologia usada para obtener los modelos que permiten alcanzar el objetivo principal. Los algoritmos de inteligencia artificial se obtienen de las librerias de SciKit Learn, correspondientes al lenguaje Python, y se trabaja en el entorno conocido como Jupyter Notebook.

Para finalizar el proyecto, se presentan las conclusiones obtenidas y los modelos que permiten conseguir una mejor aproximación a la variable incógnita, la demanda de energia eléctrica.

Abstract

This projects consists on the development of a model capable of predicting the electricity demand of the EEBE campus, using artificial intelligence tools and a history of the data recorded by the meters of the center.

Initially, an exposition of the theoretical framework that frames the work is made, being its main topics the monitoring of electricity consumption and the means for its realization, the data analysis for the optimitzation of the energy management, and the implementation of artificial intelligence in predicting electricity demand.

After this, we proceed to the analysis of the particular case, the electrical demand of the EEBE campus, for which a first introduction to the available data is made, informing about its previous treatment and the first conclusions drawn regarding its behaviour. Next, the methodology used to obtain the main objective is exposed. The artificial intelligence algorithms are obtained from the SciKit Learn libraries, which belong to Python, and the work is developed in the environment known as Jupyter Notebook.

To finalize the project, the conclusions obtained and the models that allow a better approximation to the unknown variable, the electrical energy demand, are presented.

Agraïments

Voldria expressar els meus agraïments a Roberto Villafáfila, tutor d'aquest projecte, qui no sols ha facilitat en gran mesura l'accés a la informació necessària i el traçat d'un camí de treball correcte, sinó que des d'un principi va confiar en aquest treball.

També agrair a la meva família el seu suport al llarg d'aquesta tesi i de tot el grau, que tot i la distància, han estat al meu costat en tot moment.

Finalment, gràcies a tots els companys i amics amb qui he compartit tantes hores, a classe i a la biblioteca.

Índex

1. Prefaci

1.1. Motivació

L'ús de l'energia elèctrica es troba en auge en la societat actual innegablement. Aquest fet, alhora que la necessitat d'una transició energètica a formes de generació menys nocives pel medi ambient, com són les fonts d'energia renovables, comporta la necessitat d'optimitzar la gestió dels recursos energètics disponibles.

A priori, es pot considerar que la gestió de l'energia es pot contemplar des de dues perspectives: des de la generació d'aquesta, amb la cerca de noves fonts o millora de les existents, o bé des de l'optimització del consum elèctric. Però també hi ha una tercera opció, que seria l'enllaç de les dues esmentades: la predicció del consum per adaptar la generació a les conclusions extretes. Aquest fet és principalment el que ha marcat la tria del tema de la tesi aquí present, ja que està en creixent importància amb el desenvolupament i implementació de les xarxes elèctriques intel·ligents, també conegudes com smart grids, i la generació aïllada per abastiment d'instal·lacions individuals.

D'altra banda, l'ús dels mètodes d'intel·ligència artificial es troba en augment en tots els àmbits de la vida quotidiana, fet que es demostrarà més endavant en aquest text. Això, també genera cert interès en el coneixement d'aquestes tecnologies, cada dia més properes i establertes en tot allò que ens envolta.

1.2. Estudis previs

Aquest treball guarda similitud amb altres treballs de fi de grau de diferents campus de la Universitat Politècnica de Catalunya, però cap planteja exactament el mateix objectiu ni metodologia, alhora que les dades seran sempre diferents, en tant que són d'altres campus. Alguns d'aquests treballs realitzen estudis de consum d'una determinada part de la instal·lació per la gestió d'una instal·lació solar fotovoltaica que la subministra [4] o bé anàlisi del consum d'un edifici del conjunt d'un campus [9] i per això serviran com a referència, tot i que les conclusions obtingudes seran òbviament diferents.

2. Introducció

2.1. Objectius del treball

El treball cerca realitzar una modelització de les dades de consum d'energia elèctrica del campus EEBE, analitzant les seves corbes de demanda agregada i posteriorment utilitzant les dades històriques disponibles per poder desenvolupar models de predicció mitjançant intel·ligència artificial.

Per assolir aquest objectiu, es segueix el procés marcat pels següents punts:

- Estudi de l'estat de la predicció mitjançant intel·ligència artificial, per establir uns fonaments teòrics sobre la matèria que tracta aquest treball.
- Anàlisi inicial de les dades disponibles de consum energètic de l'Escola d'Enginyeria de Barcelona Est, per entendre el seu comportament i conèixer les variables que poden intervenir en l'augment o disminució del consum.
- Càrrega de les dades analitzades en l'entorn de programació i tractament d'aquestes, alhora que addició de les variables pertinents per la predicció de la demanda elèctrica, la incògnita.
- Aplicació dels models d'intel·ligència artificial escollits.
- Comparació dels resultats obtinguts per establir el model més vàlid pel cas d'estudi.
- Optimització del model triat.

2.2. Abast del treball

En el present estudi s'han utilitzat les dades de consum energètic del campus EEBE, corresponents al registre de l'any 2019, pel desenvolupament del model estimador de la mitjana del consum horari del conjunt del campus.

La primera aproximació a la comprensió del funcionament de les dades es realitza mitjançant el software Excel del paquet Office, en el qual es realitzen tota una sèrie de representacions de les dades que, més endavant en aquesta tesi, es mostren i analitzen. Això es deu al fet que les dades s'obtenen en el format corresponent a aquest programa, des de la web SIRENA UPC.

El model esmentat s'obtindrà mitjançant els diferents algoritmes presents dintre de la biblioteca Scikit, de Python, tot en l'entorn de programació "Jupyter Notebook", que pel seu format facilita la divisió i comprensió del codi, alhora que l'addició de comentaris sobre el mateix.

La resta de dades emprades, relacionades amb les dates dels dies analitzats, s'obtenen del calendari acadèmic present en la pàgina web del campus EEBE.

3. Estat de l'art

3.1. Monitoratge de dades energètiques

Avui en dia, l'emergència climàtica ens porta inevitablement a enfocar els nostres esforços en l'optimització de l'eficiència energètica. Amb aquest propòsit, ha proliferat un model de negoci enfocat exclusivament en l'estudi de l'ús dels recursos energètics en la indústria, per així reduir el malbaratament que es pugui realitzar. Això, també reporta beneficis en les mateixes indústries consumidores d'aquest servei, en tant que suposa un estalvi econòmic per a elles. Aquestes empreses d'auditoria energètica desenvolupen un software orientat al monitoratge i planificació energètica que posteriorment implementen en les indústries client. Com es pot observar en les seves pàgines web, algunes d'elles, com WATTABIT [29] o GEMWEB [6], ja han executat projectes conjunts amb grans empreses privades i entitats públiques com ajuntaments.

Aquest àmbit ha arribat també a la Universitat Politècnica de Catalunya, on l'empresa DEXMA ha implementat el seu software a través del web app Sirena. [3]

3.1.1. SIRENA UPC

La UPC té el seu propi sistema de monitoratge i registre de dades de consum energètic, no sol elèctric, sinó que també de gas i aigua. Aquest sistema, anomenat Sirena, registra les dades dels campus de la universitat mitjançant una xarxa formada per un total de 215 analitzadors, alhora que ofereix un històric de dades per la seva consulta en qualsevol moment.

Respecte al campus EEBE, la web esmentada ofereix les dades de consum energètic en funció del registre de dos comptadors, d'una banda el principal, i d'altra banda, el de socors. El còmput del registre de tots dos ofereix la corba de demanda agregada, que seria l'objecte principal d'estudi d'aquest treball.

Aquesta aplicació és una eina vàlida per molts propòsits, sent així la base de dades per treballs com aquest, però la seva funció principal es pot considerar la de permetre el progrés cap al compliment dels propòsits establerts en el Pla UPC Energia 2020. [25]

3.1.1.1. Pla UPC ENERGIA 2020

Aquest pla, hereu del Pla d'Estalvi Energètic 2011-2014, gira envers la reducció del consum energètic realitzat en els diferents campus de la UPC, i s'emmarca dintre del propòsit de la Unió Europea recollit en l'estratègia UE 202020, que cerca complir amb els tres punts següents:

- Reduir un 20% el consum d'energia primària de la UE
- Reduir un 20% les emissions de gasos d'efecte hivernacle respecte 1990
- Elevar la contribució de les energies renovables al 20% de consum

Els principis d'aquest pla estan basats en els propis de la UPC, i són exposats en l'acord núm. 165/2015 del Consell de Govern. Tots ells, condueixen al compliment dels dos objectius principals d'aquest pla d'actuació:

- Assolir una universitat de baixa intensitat energètica i baixa emissió de carboni, sostenible a mitjà i llarg termini.
- Experimentar la innovació als campus, per potenciar el rol de la universitat com a recurs de coneixement i aprenentatge vers una societat sostenible energèticament.

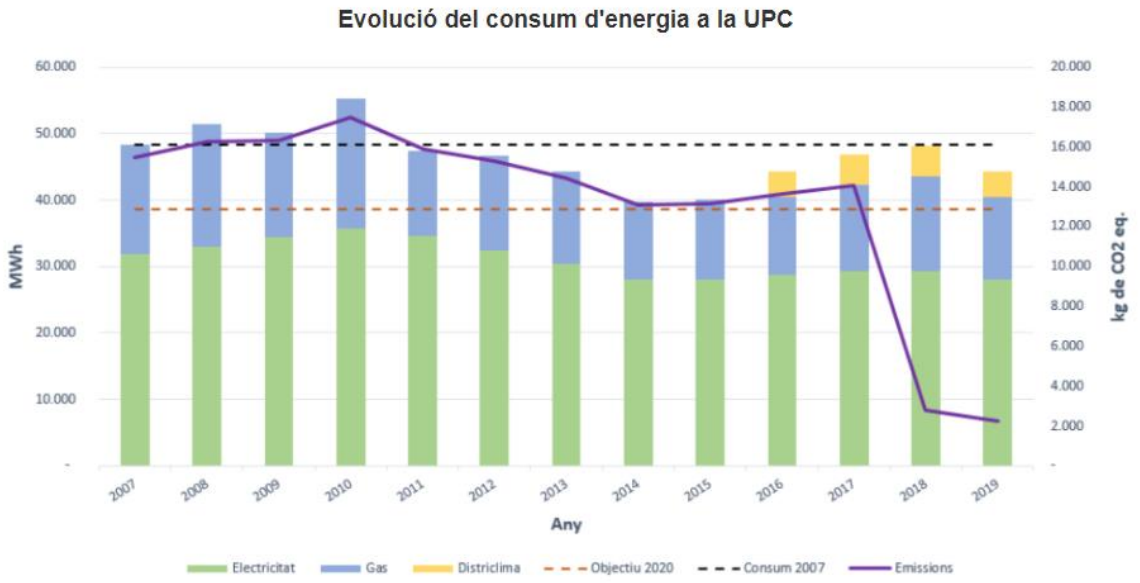


Figura 3-1.- Representació del consum energètic del conjunts de campus de la UPC des del 2007 fins al 2019. Font: Informe Sirena 2019 [28]

De forma anual, es publica l'informe SIRENA de la UPC, on es recullen totes les fites aconseguides de cara al compliment del pla esmentat, i totes elles prenent com a base les dades obtingudes mitjançant el software Sirena.

3.2. Predicció mitjançant intel·ligència artificial

Tal com s'ha exposat, l'eficiència energètica és una temàtica cada dia més important en la indústria elèctrica. Un dels mètodes seguits per l'optimització de l'ús dels recursos energètics és la predicció d'aquesta demanda elèctrica mitjançant eines d'aprenentatge computacional o intel·ligència artificial.

Aquest camp de la informàtica ha resultat útil no solament a nivell tecnològic, sinó en molts altres camps de la vida quotidiana, fins al punt d'arribar a àmbits com la política [13] o la publicitat [31]. És per tant, una eina molt versàtil i que pot aportar en gairebé qualsevol camp d'estudi.

Pel que fa al camp de l'enginyeria elèctrica, l'operador del sistema elèctric ibèric, REE, fa ús d'aquests mètodes per garantir l'equilibri entre producció i consum elèctrics en el sistema esmentat.

3.2.1. Predicció realitzada per REE

Per realitzar aquesta predicció, REE compta amb un model pel mercat diari, orientat a la predicció dels talls de subministrament de la xarxa, i vint-i-quatre models horaris pel mercat intradiari, un per cada hora, tots amb l'objectiu de realitzar prediccions que permetin mantenir l'equilibri entre generació i demanda [2]. Posteriorment , l'operador del mercat OMIE, utilitzarà aquestes prediccions per la gestió dels mercats elèctrics diari i intradiari esmentats.

Tots els models segueixen una mateixa estructura bàsica, presentada en l'equació 3.1:

$$lnC_t = p_t + s_t + CSD_t + CWEA_t + U_t \quad \textbf{(Eq. 3.1.)}$$

On:

- Ct és el consum elèctric del dia t
- Pt és la tendència de la càrrega
- St és la influència de l'estacionalitat des d'una perspectiva setmanal
- CSDt és la contribució dels dies especials, com puguin ser els festius
- CWEAt és la resta de la contribució dels factors meteorològics, fet lligat amb la variable St
- Ut representa les pertorbacions del sistema que puguin ocórrer en el curt termini

Juntament amb una àmplia disposició de mitjans, REE gaudeix d'un avantatge tècnic respecte altres entitats que realitzin prediccions d'aquest tipus, i és que REE treballa sobre la corba de demanda agregada del mercat espanyol. Això comporta que el comportament d'un dia per l'altre, que és el termini en què REE realitza les prediccions, no variarà excessivament, i per tant, es facilita la predicció.

Per realitzar les prediccions que s'esmenten, hi ha una gran quantitat de models, que es divideixen en dues grans famílies, models d'aprenentatge supervisat i no supervisat.

3.2.2. Models d'aprenentatge supervisat

Els models d'aprenentatge supervisat funcionen amb dades d'entrada i sortida conegudes per tal de desenvolupar models capaços de realitzar prediccions. És per això, que s'utilitza quan es tenen dades de sortida del model reals.

Dintre d'aquest tipus d'aprenentatge, es poden trobar dos tipus de tècniques que s'aplicaran en funció del cas a analitzar, i són la classificació i la regressió. [11]

3.2.2.1. Classificació

Els models de classificació s'empren quan les variables de sortida són qualitatives. Per ficar un exemple, es podria desenvolupar un model que, tenint en compte el fenotip d'un bolet, pugui classificar-lo com a verinós o comestible, basant-se en un registre d'altres bolets també classificats segons les seves característiques físiques, i dels quals coneixem si són verinosos o no.

Extrapolant des de l'exemple exposat, es pot veure com funcionen realment aquest tipus de models, que seria analitzant el que es coneix com a *training set*, corresponent a la part de la base de dades que empra per entrenar-se, per així les característiques que li permetran classificar en un grup o altre el resultat. Amb la resta de la base de dades, es realitzarà la validació del model, d'aquí que es conegui com a *test set*.

Alguns dels models de classificació més típics són les màquines de vectors de suport, els arbres de decisió, els k-veïns més propers i els classificadors bayessians. Cap d'ells serà aplicable al cas d'estudi present, ja que el que es pretén és predir un valor, no classificar-lo.

3.2.2.2. Regressió

Els models de regressió permeten predir variables de sortida quantitatives. Un exemple seria el d'aquest cas d'estudi, la predicció del valor de la demanda elèctrica del campus EEBE, en funció de tota una sèrie de variables d'entrada, que més endavant en la tesi es comentaran.

El funcionament consisteix en l'obtenció d'una equació que relaciona les variables d'entrada amb la variable de sortida, que serà la que el model predirà. Igual que en la classificació, s'utilitza una part de la base de dades per, en aquest cas, obtenir l'equació objectiu, i una altra part per avaluar el model obtingut.

Alguns dels models de regressió més comuns són els lineals, els no lineals i els arbres de decisió (que com podem veure, s'apliquen per models de classificació també).

3.2.3. Models d'aprenentatge no supervisat

Els models d'aprenentatge no supervisat funcionen buscant la relació que hi ha entre unes variables d'entrada i sortida que a priori és desconeguda. Com a exemple, es trobaria la segmentació de clients potencials, per així encarar noves estratègies de màrqueting [7]. Hi hauria moltes variables que podrien afectar el model, però no es sap en quina mesura ho farien.

Els algoritmes més coneguts d'aprenentatge no supervisat són els de clustering, que consisteixen en la creació de grups basats en determinats patrons o tendències similars en les dades. Per crear aquests grups, es segueixen criteris de distància.

Alguns dels models més comuns dintre del clustering són el k-means, el clustering jeràrquic, el DBScan clustering o el Principal Component Analysis. [8]

3.2.4. Funcionament dels models triats per l'estudi

Per aquest cas, s'han triat una sèrie de mètodes de regressió per veure com s'adaptaven a la resolució del problema. La tria dels models s'ha realitzat de forma arbitrària. En cas que cap dels seleccionats inicialment presenti resultats satisfactoris, es realitzaran proves amb altres regressors.

3.2.4.1. Regressió lineal

Aquesta tècnica calcula els coeficients que acompanyen les variables independents en l'equació 3.2, per així aconseguir un model lineal que permeti caracteritzar el comportament de les variables i la relació que guarden.

$$Y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \cdots + b_n \cdot x_n + error \qquad \textbf{(Eq. 3.2.)}$$

On:

- Y seria la variable incògnita, en el nostre cas, el consum d'energia elèctrica
- B0 és el terme constant
- Bn són els coeficients
- Xn són les variables independents, en el nostre cas, hora, dia, qualitat de la mesura...

El principi de funcionament seria el mètode dels mínims quadrats, el qual el model empra per millorar.

3.2.4.2. K Nearest Neighbors

Aquest algoritme es basa en la cerca de punts de dades similars apresos en l'etapa d'entrenament, i a partir d'ells, realitza la predicció. La "k" serà el nombre de punts "veïns" que tindrà en compte per realitzar aquesta predicció. [20]

3.2.4.3. Decision Trees

Els models de Decision Trees creen un arbre de decisions a partir de regles binàries, de si o no, que estableixen tenint en compte les variables d'entrada més significatives. Seran després les variables d'entrada les que marquin quina rama es segueix de l'arbre i així quin és el resultat final. [16]

3.2.4.4. Random Forests

Aquest algoritme pren com a base els Decision Trees, amb la diferència que genera un conjunt d'arbres de decisió individuals, que després es tindran en compte simultàniament per obtenir la predicció desitjada.

3.2.4.5. Adaboost

Pertanyent als grups de tècniques de Boosting, l'algoritme Adaboost (Adaptive Boosting) crea arbres de decisió, que generen una sèrie de prediccions les quals classifica com a correctes o incorrectes (tot dintre del set d'entrenament), i dóna més pes a les incorrectes, per així crear un nou arbre millorat respecte a l'anterior. [14]

3.2.4.6. Gradient Boosting

Aquest mètode també pertany al grup dels Boosting com el cas anterior. La diferència principal es troba en el fet que, en el cas anterior, s'avaluava cada error independentment assignant-li un pes, però, en aquest cas, es cerca minimitzar el gradient d'una funció error que avalua tots els punts erronis alhora. [19]

3.2.4.7. SVR

Els algoritmes coneguts com a Support Vector Machines troben els hiperplans que millor permeten separar les dades en grups, maximitzant el marge entre elles, és a dir, creant l'espai més gran possible per cada grup. En el cas del SVR, sigles corresponents a Support Vector Regressor, apliquen aquest mètode a models d'anàlisi per regressió. Tenint en compte que el punt que es desitja predir podria generar-se dintre d'un marge infinit de punts (ja que els SVM sol separen grups de dades), és essencial aplicar un valor epsilon que genera un nou marge dintre del qual es considera que es trobarà aquest valor que es cerca, i així es minimitza l'error al tolerat. [10]

4. Cas d'estudi

4.1. Condicions i dades de partida

Les dades de consum energètic utilitzades pel desenvolupament del model s'obtenen del comptador principal del campus EEBE, concretament, del registre que realitza aquest equip en l'any 2019.

El software emprat pel monitoratge del comptador és la web "Sirena UPC", tal com s'ha comentat anteriorment. Per obtenir les dades que s'han utilitzat en el primer anàlisi, previ a la programació, s'han considerat les dades enregistrades del comptador principal exclusivament, per falta de disponibilitat de les dades del comptador de socors en el moment de realització de la feina.

4.2. Tractament inicial de la informació

Previ a la introducció de les dades en l'entorn de programació, s'ha conduït un estudi per tenir una primera idea de com és el comportament de les dades i, així, conèixer les possibles causes d'aquesta conducta, que posteriorment es traduiran en les variables a tindre en compte per desenvolupar el model de predicció.

Les dades descarregades de la web Sirena són el registre horari d'energia activa (pel període de 2019) i el registre de potència activa quart-horària (corresponent a la mitjana calculada del registre de maxímetre) del mateix període. En el primer registre esmentat, les variables presents són:

- Energia consumida: expressada en kWh, és una variable quantitativa corresponent al consum energètic que s'ha realitzat en una determinada hora. És el principal objecte d'estudi, ja que és la variable a predir.
- Hora: variable que expressa quina és l'hora per un determinat valor d'energia registrat. Aquesta variable pren sempre valors entre 0 i 24 (sent aquest darrer redundant amb el 0, fet que es comentarà més endavant)
- Dia: variable que indica a quin dia de l'any correspon el registre. Aquesta variable comprèn des de l'1 de gener fins al 31 de desembre.
- Qualitat: variable que indica com s'ha obtingut el registre, si s'ha extret directament de comptador o bé s'ha obtingut a partir de la corba de potència quart-horària. També hi cap la possibilitat que aquesta variable sigui "no disponible", fet que indica que es desconeix la naturalesa del registre.

La taula 4.1 correspon al consum horari registrat en les primeres dotze hores del dia 1 de gener de 2019.

Taula 4-1.- Consum energètic horari registrat l'1 de Gener de 2019 de les 00.00h fins a les 12.00h. Font pròpia.

Dia 01/01/2019					
Hora	0	Consum (kWh)	192	Qualitat	Real
Hora	1	Consum (kWh)	193	Qualitat	Real
Hora	2	Consum (kWh)	193	Qualitat	Real
Hora	3	Consum (kWh)	192	Qualitat	Real
Hora	4	Consum (kWh)	197	Qualitat	Real
Hora	5	Consum (kWh)	198	Qualitat	Real
Hora	6	Consum (kWh)	209	Qualitat	Real
Hora	7	Consum (kWh)	234	Qualitat	No disponible
Hora	8	Consum (kWh)	233	Qualitat	Real
Hora	9	Consum (kWh)	231	Qualitat	Real
Hora	10	Consum (kWh)	230	Qualitat	Real
Hora	11	Consum (kWh)	223	Qualitat	Real
Hora	12	Consum (kWh)	205	Qualitat	Real

Pel que fa al registre quart-horari per maxímetre de potència activa, les variables que hi intervenen són:

- Energia: variable quantitativa corresponent al registre realitzat en un determinat quart d'hora. Correspon a la integració del valor registrat per maxímetre en el quart d'hora, per així obtenir l'energia mitjana consumida.
- Hora: igual que en el cas anterior, és una variable que indica l'hora en què es realitza el registre analitzat.

- Quart d'hora: variable que indica el quart d'hora al que correspon el registre. Els valors que pren són 1, 2, 3 o 4. Ficant com a exemple les 12.00 h, el quart 1 seria de 12.00 h a 12.15 h, el quart 2 de 12.15 h a 12.30 h, el quart 3 de 12.30 h a 12.45 h i el quart 4 de 12.45 h a 13.00 h.
- Dia: igual que en el cas anterior, variable que indica a quin dia de l'any correspon el registre.
- Qualitat: com en el cas anterior, variable qualitativa que indica si el registre s'ha extret directament de comptador o bé s'ha obtingut de la corba d'energia horària.

La taula 4.2 correspon al consum quart-horari registrat en el dia 1 de gener de 2019, des de les 06.00 h fins a les 11.00 h.

Amb aquests dos registres, fusionats en un mateix arxiu, s'ha representat la mitjana diària de cada mes, que servirà com a primer estudi de les dades i el seu comportament, tot desenvolupat en l'apartat 4.3.1, a continuació.

Com a mètode per a fer una primera validació de les dades, s'ha realitzat un excel on s'ha comparat la mitjana horària resultant dels registres quart-horaris del maxímetre amb el registre horari. El fet que concordin implica que els registres realment tenen una correlació.
En aquest mateix excel, s'ha realitzat la representació de la mitjana del consum elèctric diari de cada mes del 2019, obtenint les gràfiques que s'exposen en l'apartat posterior, on s'analitzaran per així extreure'n les primeres conclusions, i en l'annex B.

Taula 4-2.- Registre del consum quart-horari del dia 1 de gener de 2019 de les 06.00h fins a les 11.00 h. Font pròpia.

Dia	Hores	Quart	Energia (kWh)	Qualitat
01/01/2019	6	1	200.000	Real
01/01/2019	6	2	212.000	Real
01/01/2019	6	3	212.000	Real
01/01/2019	6	4	212.000	Real
01/01/2019	7	1	228.000	Real
01/01/2019	7	2	232.000	Real
01/01/2019	7	3	236.000	Real
01/01/2019	7	4	240.000	Real
01/01/2019	8	1	236.000	Real
01/01/2019	8	2	236.000	Real
01/01/2019	8	3	232.000	Real
01/01/2019	8	4	228.000	Real
01/01/2019	9	1	232.000	Real
01/01/2019	9	2	232.000	Real
01/01/2019	9	3	232.000	Real
01/01/2019	9	4	228.000	Real
01/01/2019	10	1	232.000	Real
01/01/2019	10	2	228.000	Real
01/01/2019	10	3	232.000	Real
01/01/2019	10	4	228.000	Real

4.3. Aproximació a l'anàlisi de les dades

Per entendre les dades que es tracten abans d'introduir-les en els algoritmes de predicció, s'ha realitzat una representació de la mitjana del consum diari de cada mes. El procediment seguit per a cada mes és l'exposat a l'esquema 4.1.

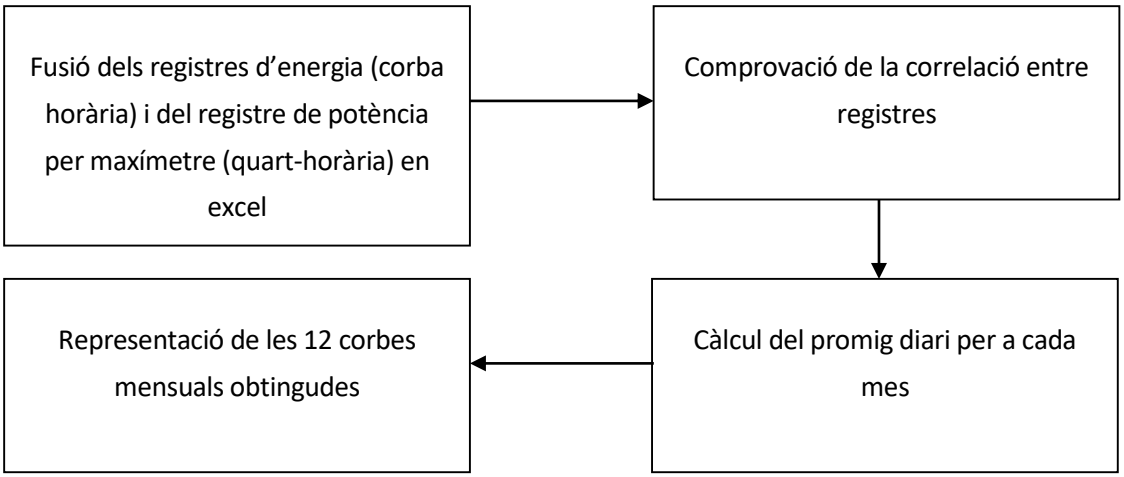


Figura 4-1.- Esquema del mètode per la obtenció de les corbes d'energia diaria de cada mes. Font pròpia.

Per obtenir el càlcul de la mitjana del consum diari, s'ha utilitzat la funció "mitjana" d'excel sobre el registre horari. S'ha realitzat així perquè, tenint en compte que la mitjana dels quatre quarts registrats en la corba quart-horària es correspon amb el registre horari, no tindria sentit fer-ho sobre el registre quart-horari, sol suposaria una sobrecarrega de dades per a l'eina de càlcul.

4.3.1. Anàlisi de les primeres gràfiques extretes

Abans d'analitzar les gràfiques, s'exposen les consideracions prèvies que s'havien realitzat, per veure si realment es compleixen o no:

- Els dies no lectius, no laborals i caps de setmana, el consum hauria de disminuir notablement.
- Els dies lectius/laborals, haurien de mantenir certa similitud en el consum, sense grans canvis. Com a molt, s'esperaria que hi hagi lleugers increments del consum en els mesos de clima més extrem, sigui fred o calor.
- El consum es concentrarà en l'horari de, aproximadament, 06.00 h del matí fins a 20.00 h de la tarda.

Un cop plantejades les hipòtesi, s'ha realitzat la representació de la mitjana del consum energètic diari de tots els mesos de l'any 2019. Totes les gràfiques extretes es poden observar en l'Annex B, i en aquest apartat, s'adjuntaran també aquelles considerades de més interès per l'anàlisi a desenvolupar.

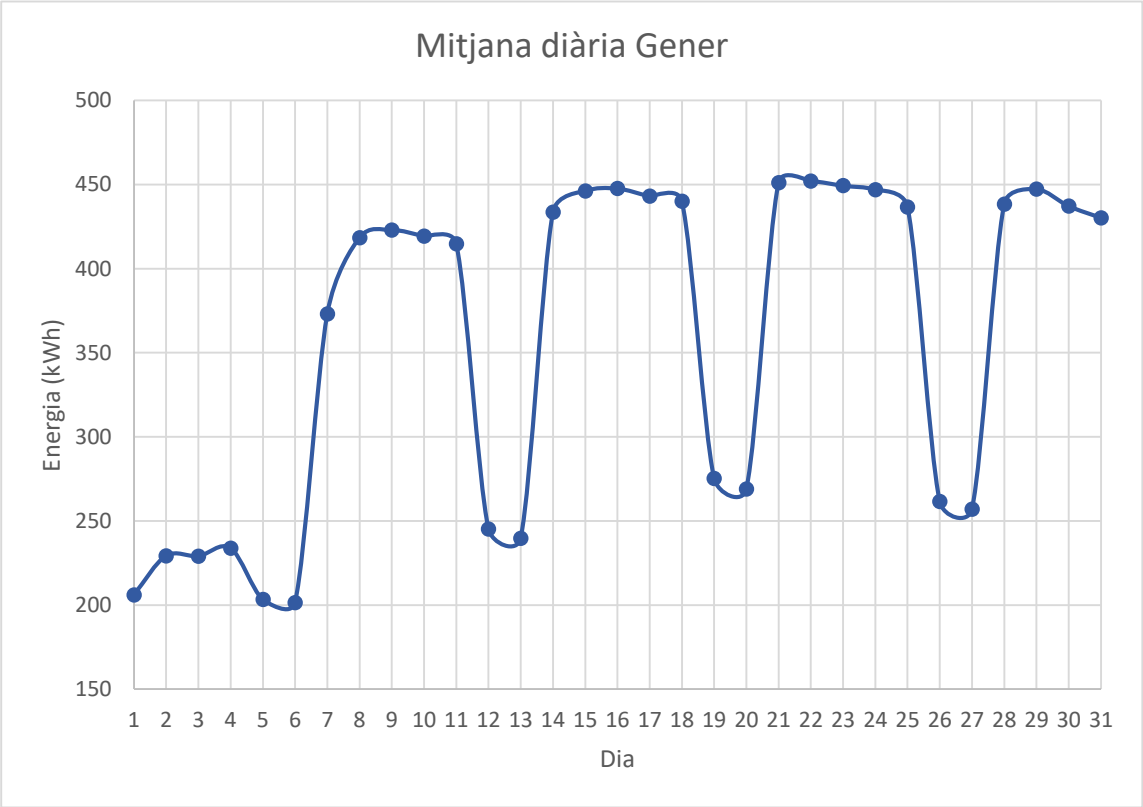


Figura 4-2.- Corba de la mitjana del consum energètic diari del mes de Gener de 2019. Font pròpia.

En la figura 4.2, corresponent al període de Gener de 2019, es pot veure com el consum davalla notablement en les primeres dates del mes, en què no hi ha activitat docent en l'escola. Es pot veure com les dues primeres hipòtesis plantejades es compleixen, havent-hi un petit increment de consum en la setmana del 14 al 20 respecte a l'anterior, atribuïble a la reincorporació de l'alumnat que no havia tornat en la setmana anterior.

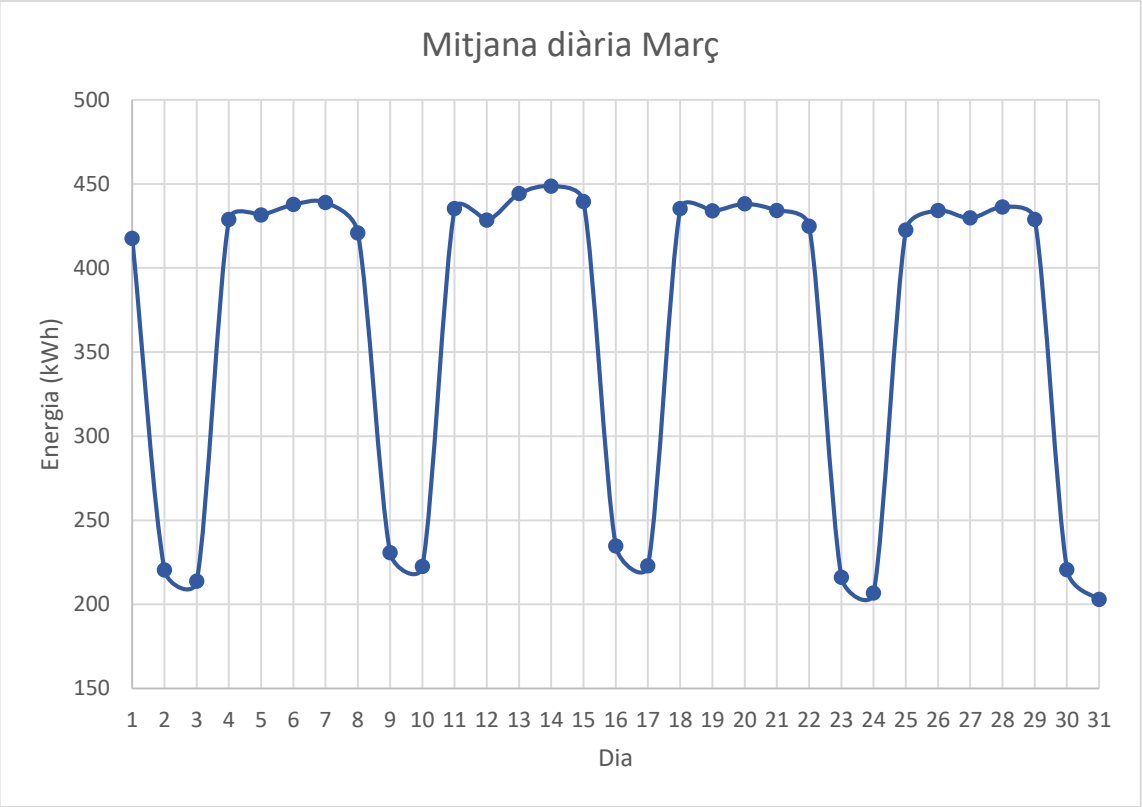


Figura 4-3.- Corba de la mitjana del consum energètic diari del mes de Març de 2019. Font pròpia.

Al mes de Març, representat a la figura 4.3, la corba respecta la constància al llarg de cada setmana. En ella no s'hi aprecia cap anomalia, per això també resulta bon exemple dels resultats esperats per un mes on tots els dies han estat laborals (excepte caps de setmana).

En el mes d'Abril, representat en la figura 4.4, s'observa una forta davallada de la corba en la setmana dels dies 15 al 21, corresponent al període de setmana santa. També es compleixen les hipòtesis plantejades inicialment, encara que es podria qüestionar la segona hipòtesi, ja que, es pot diferenciar clarament els dies de la setmana esmentada que són laborals dels que són festius, tot i que cap d'ells és lectiu. Es podria començar a sospitar que realment no és la presència d'alumnes a l'escola el que més influeix, sinó el desenvolupament d'activitat professional (i en conseqüència, la presència de professors). Si això fos cert, es podria atribuir la disminució del consum al fet que aquesta setmana correspon a un període en què una gran part de la població realitza vacances, i molts professors podrien haver aprofitat per fer-ne.

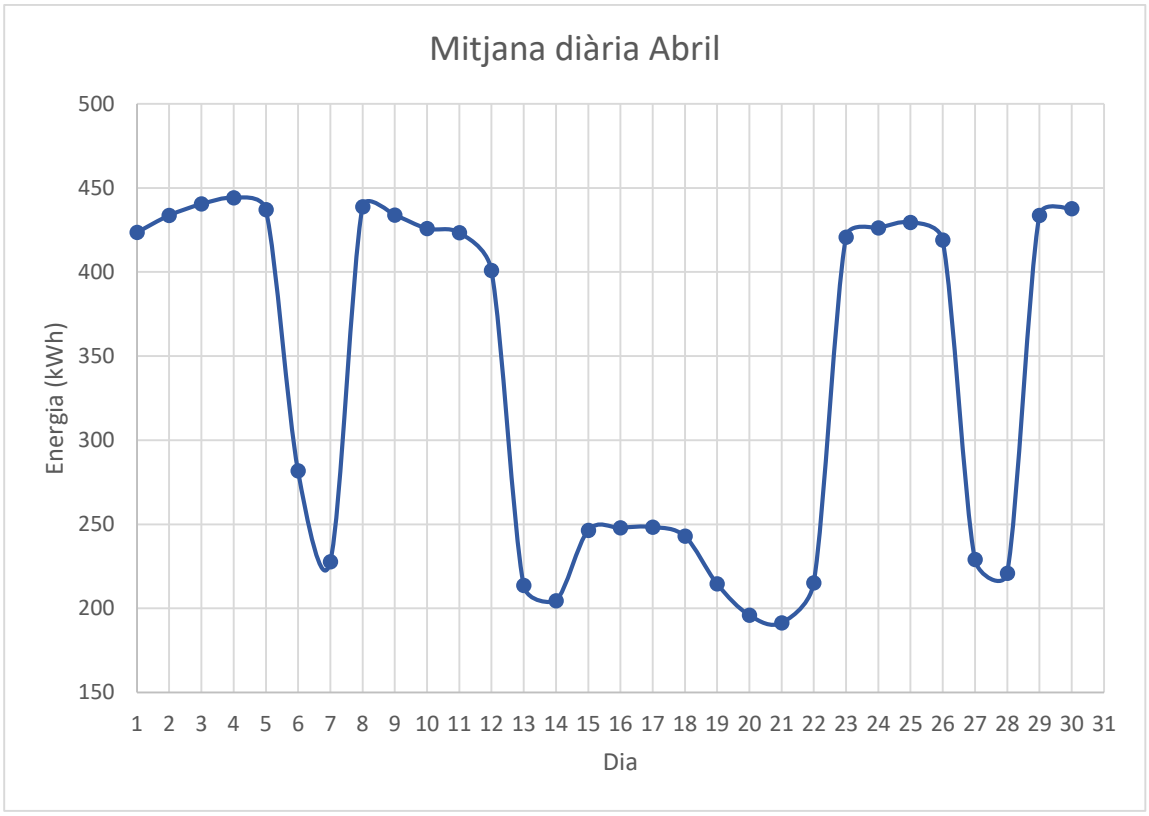


Figura 4-4.- Corba de la mitjana del consum energètic diari del mes d'Abril de 2019. Font pròpia.

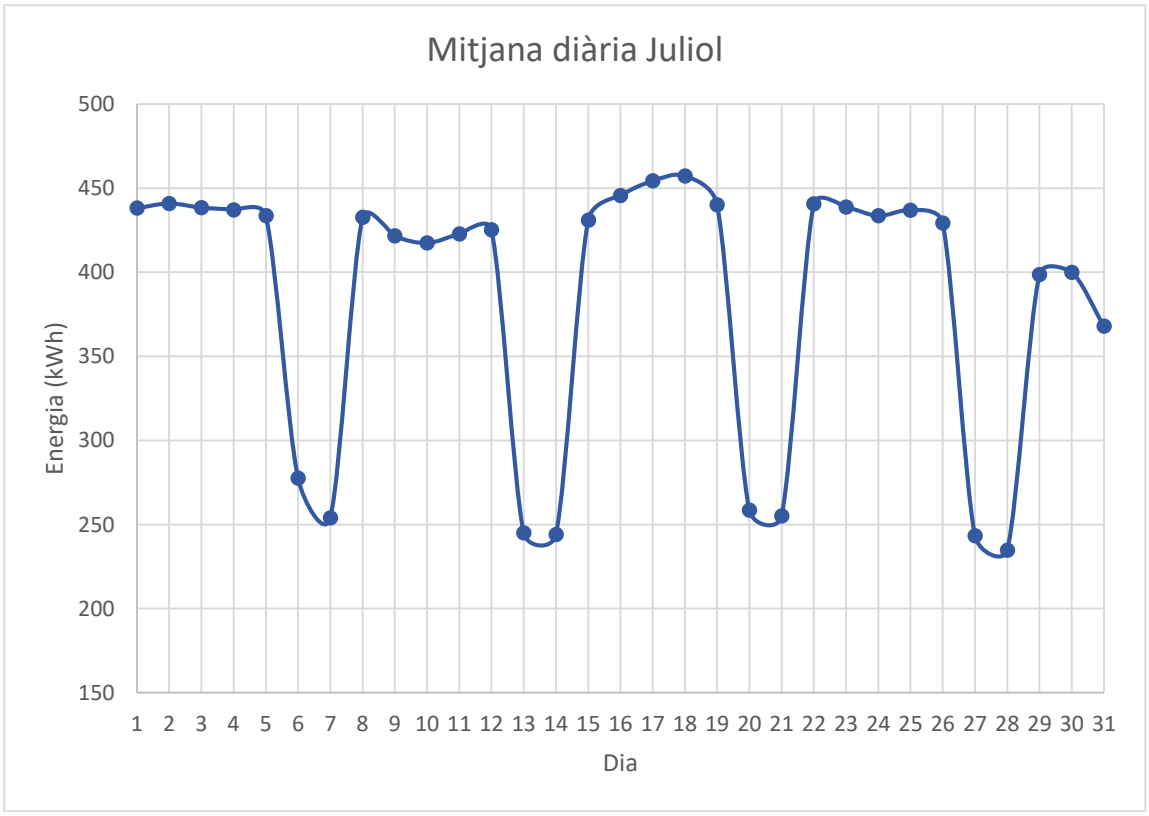


Figura 4-5.- Corba de la mitjana del consum energètic diari del mes de Juliol de 2019. Font pròpia.

El mes de Juliol, presentat a la figura 4.5, és interessant perquè s'hi observa com, tot i no realitzar-se classes, el consum es manté com en un mes lectiu qualsevol. Això porta a confirmar la hipòtesi que es comentava segons la gràfica anterior, i és que la presència d'alumnat en l'escola realment no influeix. Això fa descartar parcialment la segona de les hipòtesis inicials. Aquest fet és determinant més endavant en la tesi, a l'hora de determinar les variables d'entrada que condicionaran el model predictiu. De fet, permetrà simplificar-lo classificant els dies en laborables o no laborables.

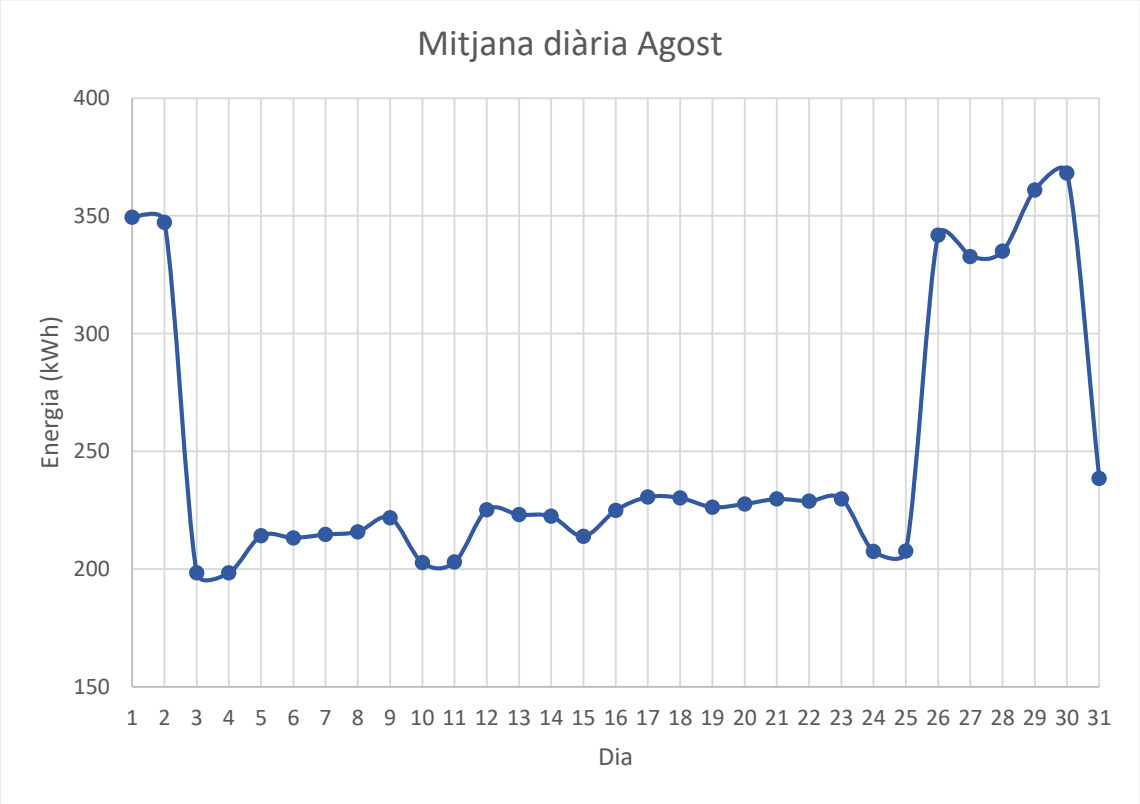


Figura 4-6.- Corba de la mitjana del consum energètic diari del mes d'Agost de 2019. Font pròpia.

En la gràfica 4.6, corresponent a l'Agost, es visualitza el mateix fenomen que en la setmana santa, una gran davallada del consum, però encara s'hi diferencia els dies laborals dels caps de setmana. Aquest fet reforça la hipòtesi plantejada sobre el període vacacional, ja que l'agost és també un dels mesos de menys activitat professional. A més, s'observa com l'activitat retorna a l'escola en la darrera setmana d'agost, fet lògic tenint en compte que poques setmanes després s'inicia de nou el quadrimestre lectiu.

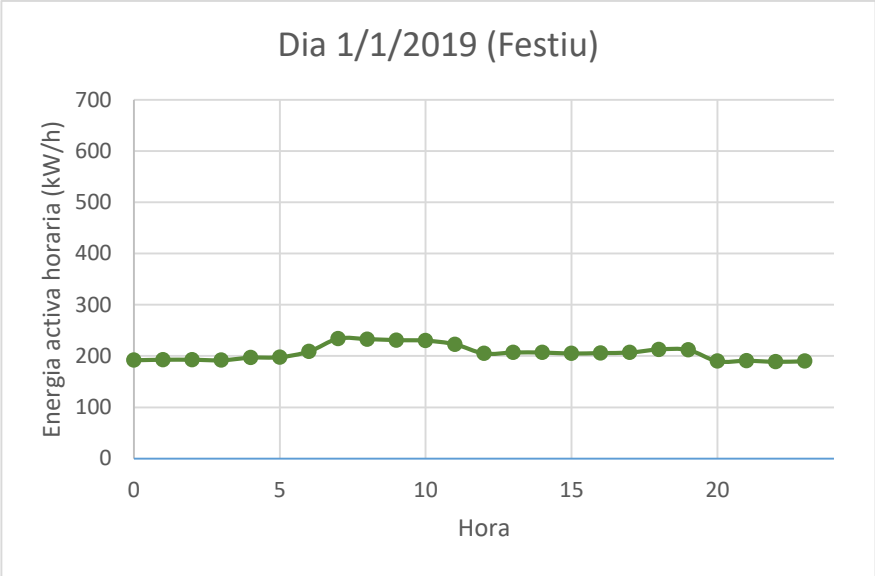


Figura 4-7.- Registre de l'energia activa horaria en el dia 1 de gener de 2019. Font pròpia.

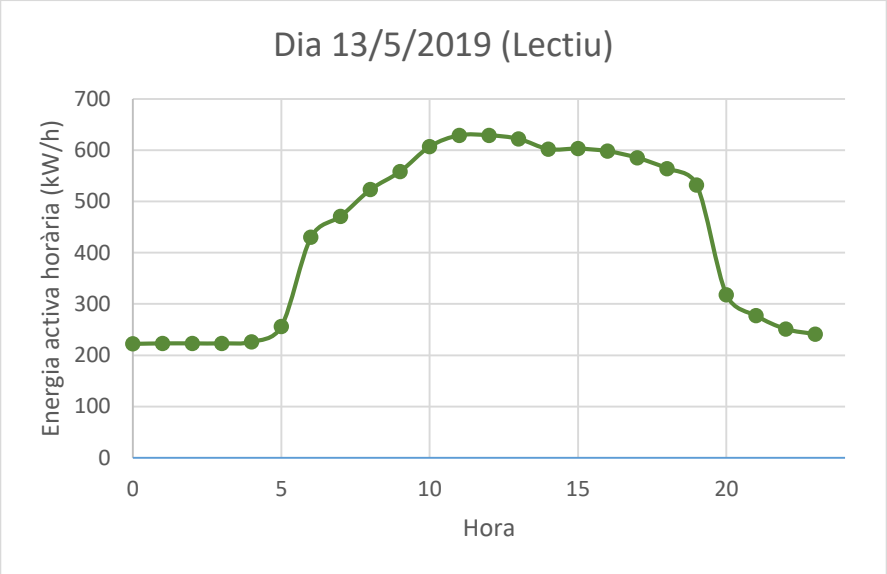


Figura 4-8.- Registre d'energia horaria activa en el dia 13 de maig de 2019. Font pròpia.

En les gràfiques 4.7 i 4.8 es visualitzen dos registres horaris, corresponents a un dia festiu i un dia laborable. En la gràfica corresponent al dia laborable, es confirma la hipòtesi que el consum es concentra aproximadament entre les 06.00 h i les 20.00 h, fet condicionat per l'horari de l'escola.

4.3.2. Conclusions del primer anàlisi

Sintetitzant el que s'ha exposat d'acord amb l'estudi de les diferents corbes obtingudes, es pot afirmar el següent:

- Les variables que principalment influeixen en el consum elèctric són el dia, hora i el desenvolupament d'activitat professional o de recerca en el campus.
- No s'ha fet cap apreciació sobre la variable "Qualitat de la mesura" en tant que no s'hi ha observat cap afectació en les dades derivada d'aquesta.
- Es descarta la hipòtesi que la realització d'activitat docent influeix en el consum de l'escola, és a dir, la presència d'alumnat no implica una diferència significativa en el consum energètic de l'escola.

Tenint en compte tota aquesta informació, es considera que es pot procedir amb la programació del model de predicció, l'obtenció del qual és el propòsit principal d'aquest treball.

5. Predicció mitjançant intel·ligència artificial

5.1. Eines emprades

Com s'ha comentat anteriorment en la tesi, la programació es realitzarà en llenguatge Python i en l'entorn conegut com Jupyter Notebook. S'ha optat per aquestes opcions perquè es corresponen amb les treballades en altres assignatures del grau, alhora que són més senzilles a nivell d'interpretació per usuaris no especialitzats en programació, tot i mantenir el potencial que altres llenguatges més complexos tenen.

Es carregaran els mòduls csv i sys, però s'utilitzaran per a la càrrega inicial de les dades i per la realització de proves i comprovacions sobre els dataframes, respectivament. Alhora, s'empraran les biblioteques Pandas, Matplotlib i Scikit Learn. A continuació, es fa una breu explicació de la utilitat de cadascuna.

5.1.1. Pandas

Pandas és una biblioteca orientada a la manipulació de bases de dades, construïda sobre el paquet NumPy. La utilitat principal és la de permetre treballar sobre taules conegudes com a DataFrames, que faciliten la classificació de les dades carregades en el programa, en el nostre cas, l'arxiu de la corba de potència de maxímetre quart-horària. [17]

5.1.2. Matplotlib

Aquesta biblioteca s'empra per realitzar representacions de les dades que en facilitin l'anàlisi. Permet crear diferents tipus de gràfiques, i en el cas d'estudi, permet visualitzar la distribució de les dades d'entrenament, les de test, i més opcions que més endavant en la tesi s'exposaran. [12]

5.1.3. Scikit Learn

Scikit Learn és la biblioteca que conté els algoritmes d'aprenentatge computacional que es poden aplicar als conjunts de dades per obtenir-ne models predictius. Està construïda sobre NumPy, SciPy i Matplotlib. Aquesta biblioteca és la que permetrà aplicar els diferents models predictius sobre la base de dades present. [21]

5.2. Càrrega de dades

Un cop s'ha carregat l'entorn Jupyter Notebook, s'han d'introduir les dades a tractar. Per a fer-ho, es passa l'arxiu corresponent al registre horari d'energia activa al format csv (comma separated values), per permetre'n la comprensió per part de Python i la biblioteca Pandas.

5.3. Processament de les dades

5.3.1. Representació gràfica de la mostra inicial

Mitjançant les eines que les llibreries esmentades ens faciliten, s'han extret algunes gràfiques per analitzar el conjunt de dades a treballar, més enllà de les ja exposades a l'apartat 4.3.1 d'aquest treball.

En la Figura 5.1., on s'observa la corba d'energia horària de tot l'any, s'hi poden diferenciar clarament els períodes vacacionals de Setmana Santa, Agost i Nadal. En la gràfica, s'observa com la corba de consum es manté en valors similars al llarg de l'any, davallant en els períodes d'aproximació a l'inici i fi de les vacances d'Agost. Amb aquest primer anàlisi general, es procedeix a analitzar la figura 5.2.

En la figura 5.2. es representa l'histograma corresponent al registre d'energia horària activa, graficant així els diferents intervals de dades i la seva freqüència d'aparició en el conjunt de dades. Es pot veure com la majoria del registre són valors entre 200 i 270 kWh, corresponents principalment a les hores de tancament del campus, els caps de setmana i els dies no laborables, i alhora, també destaca el rang entre 550 i 620 kWh, on encaixarien les dades del consum en les hores de màxima activitat docent i de recerca al campus, que al ser un registre més restringit, presenta una freqüència molt inferior a la del primer esmentat. Aquesta gràfica ens permet intuir que el model tendirà a agrupar les dades en un d'aquests dos conjunts. La resta de barres es corresponen amb hores de transició, com puguin ser el tancament i obertura de l'escola, o valors donat per condicions poc comunes, que es donen especialment en períodes de baixa presència al campus.

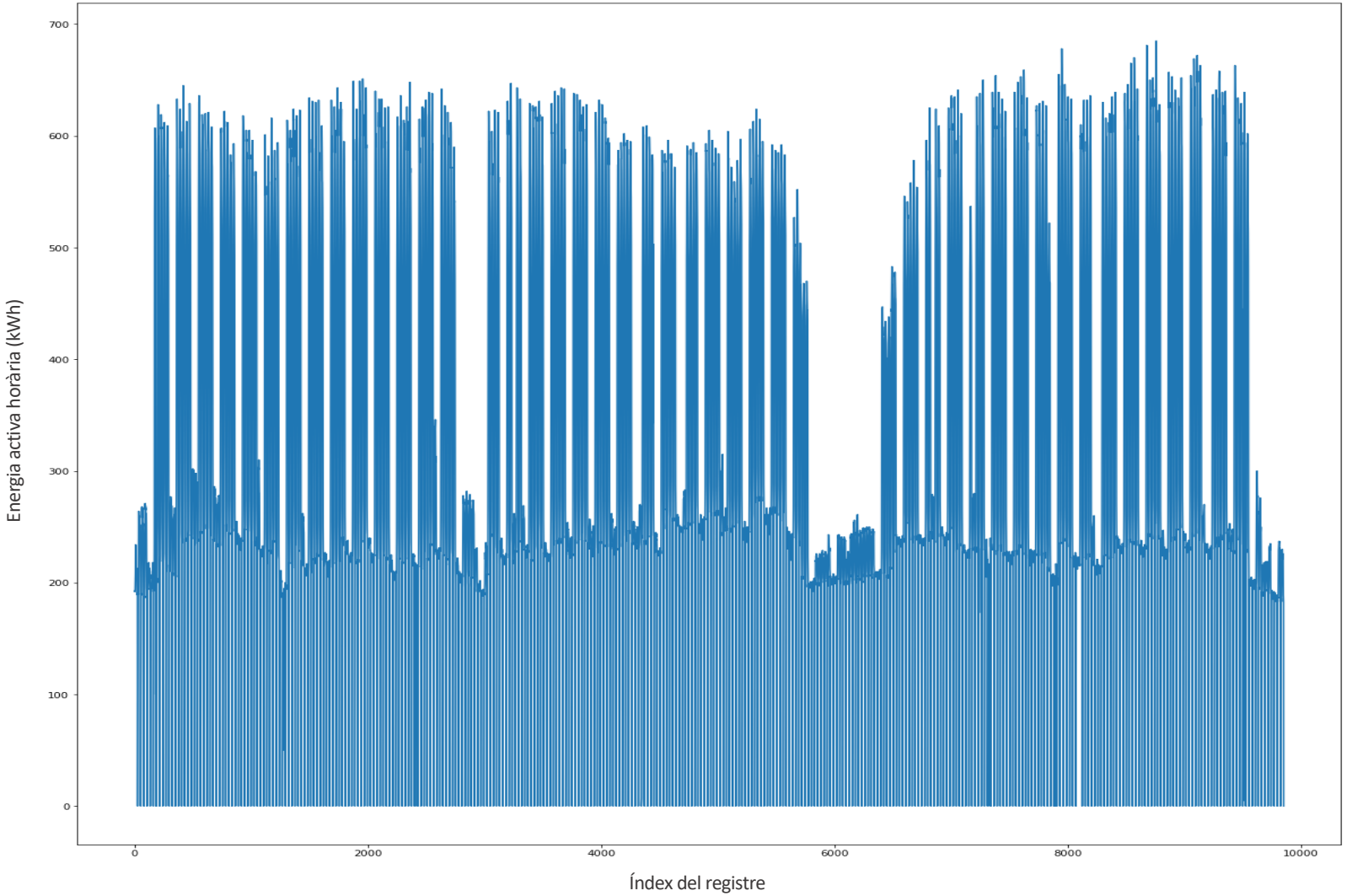


Figura 5-1.- Representació de la corba d'energia horària de tot el conjunt de dades. Font pròpia.

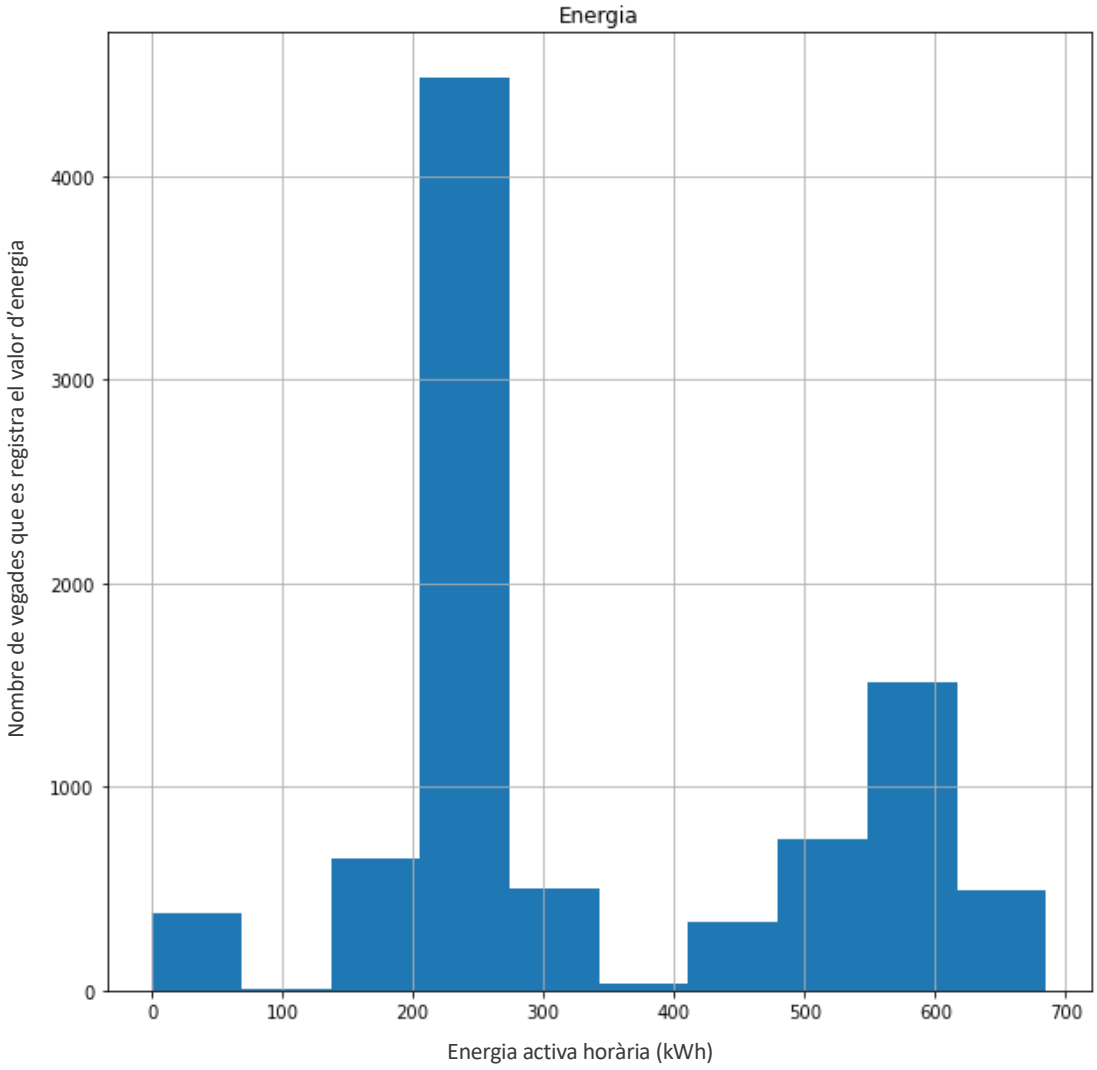


Figura 5-2.- Histograma de la corba d'energia horària de tot el conjunt de dades. En l'eix d'abscises, valor d'energia registrat en kWh, i en l'eix d'ordenades, número de vegades que es registra el valor. Font pròpia.

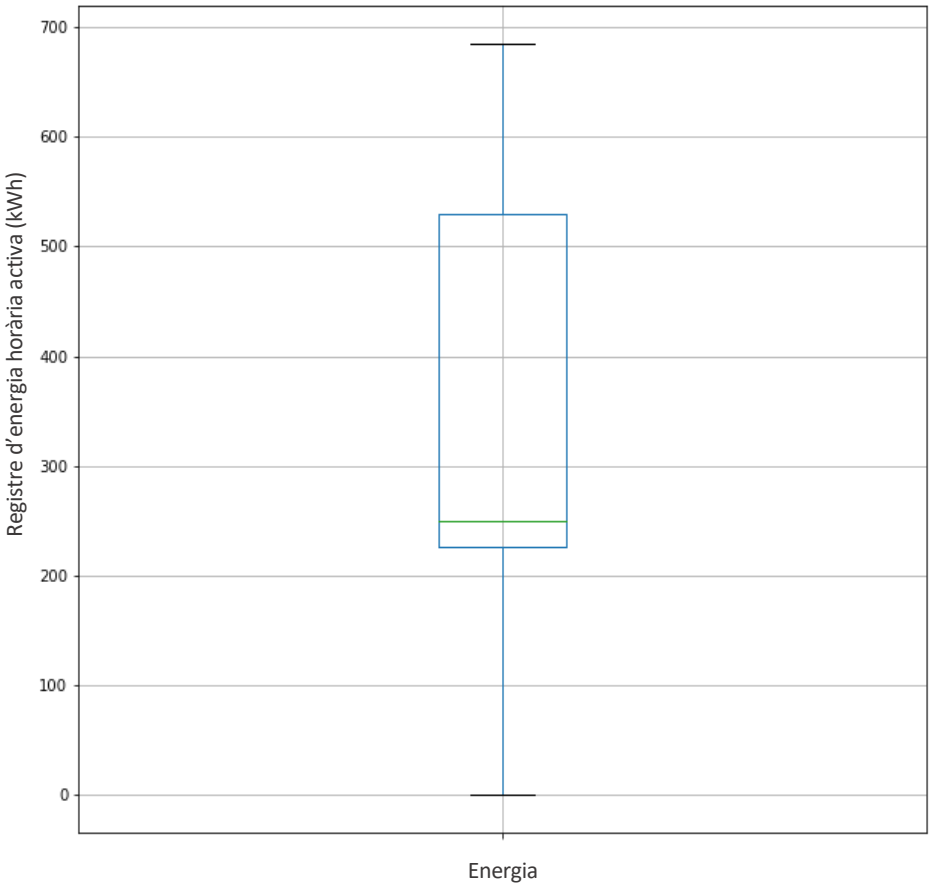


Figura 5-3.- Diagrama de caixa de la distribució de les dades d'energia horària en kWh. Font pròpia.

Taula 5-1.- Quartils del diagrama de caixa de la figura 4.3. Font pròpia.

Quartil	Valor d'energia (kWh)
Límit inferior	0
Q1 (=25)	226
Q2 (=50)	249
Q3 (=75)	529
Límit superior	642

Les dades representades en el diagrama de caixa de la figura 5.3, reafirmen el que s'ha exposat en l'anàlisi de la figura 5.2. Observem com la mediana de les dades és molt propera al valor del primer quartil, 249 kWh i 226 kWh respectivament, fet que indica que la meitat dels registres es troba en el

rang entre 0 i 249 kWh, rang molt més petit que el que compren la resta de dades, que seria entre 249 i 642 kWh.

5.3.2. Tractament de les dades

Aquesta part del procés consisteix en la neteja de les dades presents, per evitar problemes derivats d'errors en el registre, com puguin ser hores duplicades, files buides, registres erronis, etc. En l'annex A, on s'adjunta tot el codi, es pot observar quins han sigut els passos concrets seguits pel preprocessament de les dades. De forma sintetitzada, els errors solucionats són els següents:

- En la base de dades en format csv, hi apareixien dues files buides entre un dia i un altre. Aquestes files s'han eliminat.
- En el registre, hi apareixen les 00.00 h i les 24.00 h , redundants. S'ha optat per eliminar el registre de l'hora 24, ja que sempre era buit, al contrari de l'hora 0, que sí que aporta dades d'energia. Per facilitar els càlculs a realitzar per la implementació de variables, aquest pas s'ha dut a terme just abans de la normalització de les dades, quedant així intercalat entre les dues passes esmentades tot el que seria l'addició de noves variables per classificar cada registre.

5.3.3. Variables pel desenvolupament del model

Alhora que es realitza una neteja de dades, també s'han afegit variables que permeten classificar cada registre en funció de les observacions que s'han realitzat respecte al comportament de les dades. Les variables afegides són:

- Dia: corresponent al número de dia del qual es tracta. Per exemple, l'1 de gener seria el dia 1, l'1 de febrer és el dia 32, etc. Aquesta variable resulta útil per aplicar les dues següents, i podria ser interessant de cara a la predicció del consum d'altres anys, realitzant una ampliació de la base de dades que utilitza l'algoritme regressor. Aquesta variable s'ha afegit tot just després d'eliminar les dues files buides que constaven entre cada registre diari, i tenint en compte el fet que hi ha un registre redundant (hora 0 i 24 d'un mateix dia).

- Cap de setmana: variable que classifica un determinat registre en funció de si el dia al qual pertany és cap de setmana, cas en el qual prendrà el valor 1, o bé si és un dia entre setmana, cas en el qual prendrà valor 0. Per aplicar-lo, es fa ús de la variable "Dia", tal com s'exposava abans. Per aplicar aquesta variable, es té en compte que el dia 1 de gener de 2019 era dimarts, per tant, el primer dissabte i diumenge són 5 i 6 de gener respectivament. En conseqüència, l'índex del proper dissabte serà 12 (fruit de sumar set dies al primer dissabte dia 5), el següent

serà el dia 19... Fet que porta a establir que, la classificació en caps de setmana, es pot realitzar segons les equacions 5.1 i 5.2.

$$Índex_{diss} = 5 + 7 \cdot i \quad ; \qquad \forall\, i \in [0{,}53) \qquad \textbf{(Eq. 5.1.)}$$

On:

- $Índex_{disc}$: Número de dia de l'any del dissabte en qüestió. Per exemple, el primer dissabte de 2019, dia 5 de gener, tenia índex 5.
- i : valor que incrementa per poder classificar els dissabtes de cada setmana. Per la setmana 1, serà 0, per la setmana 2 serà 1... fins a 52, ja que la setmana 53 de l'any acaba en dimarts, i si no, sortiria de rang.

$$Índex_{dium} = 6 + 7 \cdot i \quad ; \qquad \forall\, i \in [0{,}53) \qquad \textbf{(Eq. 5.2.)}$$

On:

- $Índex_{disc}$: Número de dia de l'any del diumenge en qüestió. Per exemple, el primer diumenge de 2019, dia 6 de gener, tenia índex 6.
- i : valor que incrementa per poder classificar els diumenges de cada setmana. Igual que en el cas anterior, per la setmana 1, serà 0, per la setmana 2 serà 1... fins a 52, ja que la setmana 53 de l'any acaba en dimarts, i si no, sortiria de rang.

- Festiu: variable que classifica un determinat registre segons si el dia al qual pertany és festiu, cas en el qual prendrà valor 1, o bé si és laborable, cas en el qual prendrà valor 0. També es fa ús de la variable "Dia" per classificar els dies en festius o no, com en el cas anterior. Això es pot realitzar ja que l'índex de tots els festius anuals és el mateix (per exemple, l'1 de gener sempre és el dia 1), amb l'única excepció de la Setmana Santa. Pel càlcul de l'índex de setmana santa, s'ha de tindre en compte el calendari lunar, en tant que s'estableix que el diumenge de rams serà el diumenge immediatament següent a la primera lluna plena de la primavera. Això implica que per a calcular l'índex d'aquesta festivitat s'hauria d'utilitzar el calendari lunar, però s'ha desestimat en tant que en el cas d'anàlisi les dades sol apliquen per la predicció de part del 2019. En cas de realitzar prediccions per nous anys, s'hauria de tindre en compte, ja sigui automatitzant el càlcul de l'índex d'aquestes dates, o bé introduint-les directament manualment al vector on consta l'índex al qual correspon cada dia festiu de l'any, que pel model del 2019, s'ha anomenat "festius_aux".

En la taula 5.2, s'observa com queda per ara part del conjunt de dades.

Taula 5-2.- Conjunt de dades amb les variables afegides. Font pròpia, extreta directament com a imatge de l'entorn de programació.

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana	Festius
0	0.0	192.0	Real	1.0	0.0	1.0
1	1.0	193.0	Real	1.0	0.0	1.0
2	2.0	193.0	Real	1.0	0.0	1.0
3	3.0	192.0	Real	1.0	0.0	1.0
4	4.0	197.0	Real	1.0	0.0	1.0
...	...	...	...	...	...	...
9847	19.0	226.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9848	20.0	198.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9849	21.0	189.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9850	22.0	183.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9851	23.0	183.0	No disponible	365.0	0.0	0.0

8760 rows × 6 columns

5.3.4. Normalització de les dades

El primer pas a dur a terme, previ a la normalització de les dades, és la divisió del conjunt de dades disponibles en dades d'entrenament per l'algoritme i dades per la posada a prova del model proposat. Per realitzar aquesta separació, es decideix de forma arbitrària que el valor llindar serà 2/3 del total de dades, i també es fa una barreja del conjunt de dades. En conseqüència, 6083 registres del total seran destinats a l'entrenament del model, i la resta, a l'avaluació d'aquest.

Un cop s'ha establert la variable "llindar", també s'ha optat per esborrar la variable "Qualitat de la mesura" de la taula, en tant que no s'hi aprecia que aporti cap diferència en els registres energètics que el model realitza.

Una altra de les mesures a tindre en compte abans d'aplicar els diferents algoritmes regressors és la normalització del conjunt de dades. La normalització és un mètode matemàtic aplicable quan les variables que intervenen tenen magnituds i rangs diferents entre elles. En aquest cas, tenint en compte que la variable hores va de 0 a 23, la variable dies d'1 a 365, i les variables cap de setmana i festius són binàries, sembla convenient normalitzar el conjunt de dades. Tot i això, s'ha optat per realitzar dues opcions de codi, una on es normalitzen les variables de forma prèvia a l'aplicació de models regressors, i una altra on no es fa aquesta normalització. Posteriorment, es compararan els resultats obtinguts

amb tots dos models per valorar si realment és necessari aquest procediment de normalització de les dades.

Dintre el codi on s'ha optat per normalitzar les dades, s'han seguit els següents passos:

- S'elimina de la taula de dades la columna "Energia", ja que la variable a predir no es normalitza, i es guarda en una nova variable.
- Càlcul de la mitjana de cada variable, sent els resultats els de la taula 5.3:

Taula 5-3.- Mitjana de cada variable. Font pròpia.

Variable	Mitjana del conjunt
Hora	11,477
Dia	183,667
Cap de setmana	0,285
Festius	0,043

- Càlcul de la desviació estàndard de cada variable, sent els resultats els de la taula 5.4:

Taula 5-4.- Desviació estàndard de cada variable. Font pròpia.

Variable	Desviació estàndard del conjunt
Hora	6,899
Dia	105,586
Cap de setmana	0,451
Festius	0,202

- Creació d'una nova taula de dades, on s'afegeixen les dades normalitzades, obtingudes de la primera taula 5.2, sense la columna "Energia" ni "Qualitat de la mesura", i aplicant l'equació 5.3. a tots els valors.

$$Valor_{normalitzat} = \frac{Valor_{inicial} - Mean_{var}}{Std_{var}} \quad \textbf{(Eq. 5.3.)}$$

On:

- $Valor_{normalitzat}$: Nou valor que prendrà una determinada variable per un registre concret.
- $Valor_{inicial}$: Valor de la variable en la taula 5.2.
- $Mean_{var}$: Mitjana de la variable, segons taula 5.3.
- Std_{var}: Desviació estàndard de la variable, segons taula 5.4.

- En la taula final obtinguda amb els valors normalitzats, s'hi afegeix la columna corresponent al registre d'energia, que s'havia esborrat en el primer pas del procés.

Finalment, s'obté com a resultat del procés la taula 5.5.

Taula 5-5.- Primers cinc registres del conjunt de dades amb les variables normalitzades. Font pròpia, extreta directament com a imatge de l'entorn de programació.

	Hora	Dia	Cap de setmana	Festius	Energia
2606	0.360703	-0.811139	1.587376	-0.209686	230.0
9618	-0.791731	1.661196	-0.629863	-0.209686	255.0
4009	0.216649	-0.316672	-0.629863	-0.209686	615.0
6188	-0.935785	0.453556	1.587376	-0.209686	201.0
6723	-1.656055	0.643735	1.587376	-0.209686	237.0

5.4. Aplicació de models regressors i selecció del guanyador

Per l'aplicació dels diferents models que s'han exposat en l'estat de l'art previ, s'ha creat la funció regressió. En ella s'ha implementat l'entrenament del model, la realització de prediccions, i l'obtenció dels criteris r^2 i error quadràtic mig, que serviran per seleccionar el model guanyador. Tot això s'observa en la figura 5.4, corresponent a les línies de codi on es defineix la funció.

```
# Primer de tot, s'ha de definir la funció regressió.
#Per evaluar els models, utilitzarem els criteris de r2 (a maximitzar) i error quadràtic mig (a minimitzar)
from sklearn.metrics import r2_score, mean_squared_error
def regressio(nom,rgs):
    print(nom)
    rgs.fit(P_horaria2.iloc[:llindar,:-1],P_horaria2.iloc[:llindar,-1])

    prediccions = pd.Series(rgs.predict(P_horaria2.iloc[llindar:,:-1]), name = 'Prediccions')
    reals = pd.Series(P_horaria2.iloc[llindar:,-1], name='Reals')
    reals.index = range(P_horaria2.shape[0]-llindar)

    resultatr2 = round(r2_score(reals,prediccions),3)
    print('Error r2:',resultatr2)
    print('E.Q.M. :',round(mean_squared_error(reals,prediccions), 3))
    EQM = round(mean_squared_error(reals, prediccions), 3)


    return resultatr2, EQM, prediccions, reals
```

Figura 5-4.- Codi corresponent a la definició de la funció regressió. Font pròpia.

Per la implementació dels 4 paràmetres esmentats en el paràgraf anterior, s'empren les següents funcions:

- Rgs.fit: per crear el model de predicció tenint en compte un determinat algoritme, que ve definit pel valor de la variable "rgs", establert en executar la funció.
- Rgs.predict: per realitzar les prediccions tenint en compte el model prèviament entrenat amb la funció fit.
- R2_score: permet obtenir el valor del paràmetre r^2, per avaluar el model entrenat. Aquest paràmetre es considera millor a més proper a 1 es troba, tenint en compte que s'ha de trobar en el rang entre 0 i 1, amb algunes excepcions en què pot presentar valors inferiors a 0, pels quals significa que la mitjana de la sèrie de dades és una millor aproximació que la predicció que l'algoritme regressor realitza. Aquest paràmetre és el que es té principalment en compte per la selecció del model.
- Mean_squared_error: permet obtenir el valor de l'error quadràtic mig, per avaluar el model entrenat. Aquest paràmetre es considera millor a més baix és el seu valor, i s'utilitza com a alternativa al r^2, per si es presenta algun problema amb aquest.

Tenint això en compte, s'aplica aquesta funció als diferents algoritmes de regressió considerats, en la versió del codi normalitzada i sense normalitzar, i els resultats són els presentats a les taules 5.6 i 5.7.

Taula 5-6.- Avaluació dels models en el cas amb variables normalitzades. Font pròpia, extreta directament com a imatge de l'entorn de programació.

	r2	mse
Regressió Lineal	0.293	18032.817
kNN (k=1)	0.937	1599.140
Arbres de decisió	0.986	364.162
Random Forests	0.989	278.609
Adaboost	0.679	8175.381
Gradient Boosting	0.871	3299.147
SVR	0.688	7945.986

En el cas de les variables normalitzades, l'algoritme que presenta millors resultats pels dos criteris de selecció (r^2 i error quadràtic mig) és Random Forests. Els valors de r^2 i e.q.m. varien amb cada execució del codi, però sempre és el mateix model el que funciona millor respecte els comparats. Tenint en compte que el resultat és 0,989, es descarta l'opció d'aplicar altres algoritmes, ja que és un valor difícilment optimitzable. Seguidament, es presenten els resultats en el cas de les variables sense normalitzar.

Taula 5-7.- Avaluació dels models en el cas amb variables sense normalitzar. Font pròpia, extreta directament com a imatge de l'entorn de programació.

	r2	mse
Regressió Lineal	0.307	17852.909
kNN (k=1)	0.836	4223.683
Arbres de decisió	0.982	458.817
Random Forests	0.992	201.166
Adaboost	0.729	6979.180
Gradient Boosting	0.870	3355.646
SVR	-0.233	31753.320

En la implementació dels algoritmes amb variables sense normalitzar, s'observa que el millor algoritme és Random Forests, igual que en el cas anterior, i amb un valor r2 molt similar a l'obtingut amb l'altra opció. També es dóna el fet que els valors de r^2 i e.q.m. varien amb cada execució del codi, però sempre

és el mateix model el que funciona millor respecte els comparats. En ser un valor de r2 tan alt, també es descarta la prova de nous algoritmes.

Respecte els altres regressors, no es pot extreure una conclusió, ja que alguns milloren amb la normalització i d'altres no. També destaca el fet que, per l'algoritme SVR es dóna un valor r^2 negatiu al no normalitzar, cas estrany però contemplat en l'explicació dels paràmetres a tindre en compte per l'avaluació dels models, en l'inici d'aquest mateix apartat.

Sintetitzant la informació d'aquest apartat, es conclou que el millor model per l'aplicació és Random Forests, amb uns alts valors de correlació entre prediccions i registres reals, i que no influeix la normalització de les variables.

5.5. Optimització del model guanyador

Per la cerca d'una millora del model, s'ha creat una funció, presentada en la figura 5.5., per valorar el funcionament de l'algoritme tenint en compte diferents valors d'estimadors, concretament, 5, 10, 15, 20 i 25. Aquesta funció no presenta cap variació entre les dues versions del codi.

```
%%time

# Per comparar resultats, farem el següent: 5 execucions per a cada nombre d'estimadors.
estimadors=[]
ResultatsRFS=[]
MSEs = []
Rcuads=[]
for i in [5,10,15,20,25]:
    Comparaciomitja= []
    Rcuadsmitja=[]
    for j in [1,2,3,4,5]:
        result = regressio('Random Forests (n='+str(i)+'), execució numero '+str(j)+':', RandomForestRegressor(n_estimators=i))
        ResultatsRFS.append(result)
        estimadors.append('Random Forests (n='+str(i)+'), execució numero '+str(j)+':')
        Comparaciomitja.append(result[1])
        Rcuadsmitja.append(result[0])
    MSEs.append(np.asarray(Comparaciomitja).mean())
    Rcuads.append(np.asarray(Rcuadsmitja).mean())

# Comparativa = pd.DataFrame(data=ResultatsRFS,index=[estimadors],columns=['r2','mse'])
ComparativaErr = pd.DataFrame(data=Rcuads,index=['Mitja per 5 estimadors','Mitja per 10 estimadors','Mitja per 15 estimadors','Mi
ComparativaErr['mse']=MSEs
ComparativaErr
```

Figura 5-5.- Codi per l'obtenció de la comparativa de l'execució de l'algoritme amb diferents valors d'estimadors. Font pròpia, extreta directament de l'entorn de programació.

Per a cada valor del paràmetre "nombre d'estimadors", s'ha executat el codi 5 cops, com es pot veure en la figura 5.5, i s'ha calculat la mitjana de les 5 execucions. Els valors obtinguts es presenten en la taula 5.8.

Taula 5-8.- Comparació dels valors r2 i error quadràtic mig obtinguts pels diferents paràmetres considerats, amb les variables normalitzades. Font pròpia, extreta directament de l'entorn de programació.

	r2	mse
Mitja per 5 estimadors	0.9896	266.2092
Mitja per 10 estimadors	0.9892	273.9410
Mitja per 15 estimadors	0.9890	279.5378
Mitja per 20 estimadors	0.9898	259.3208
Mitja per 25 estimadors	0.9896	265.3042

En el cas de la taula 5-8, s'observa que el millor nombre d'estimadors és 20, que presenta el valor més alt de r^2 i més baix d'error quadràtic mig, però tenint en compte que la diferència entre resultats és mínima, s'ha optat per executar el codi diversos cops, i s'observa que depenent de l'execució, el millor nombre és un o altre, sempre mantenint tots ells un alt valor de r2 (superior a 0,97).

Taula 5-9.- Comparació dels valors r2 i error quadràtic mig obtinguts pels diferents paràmetres considerats, amb les variables sense normalitzar. Font pròpia, extreta directament de l'entorn de programació.

	r2	mse
Mitja per 5 estimadors	0.9894	270.3898
Mitja per 10 estimadors	0.9908	233.1238
Mitja per 15 estimadors	0.9910	233.4874
Mitja per 20 estimadors	0.9912	226.4278
Mitja per 25 estimadors	0.9910	234.0334

En la taula 5.9, es presenten els resultats dels diferents paràmetres pel cas de les variables sense normalitzar. Sembla que el model presenta millors resultats respecte la taula 5.8, però la diferència segueix sent mínima, i més entre els diferents paràmetres. També succeeix que depenent de l'execució, el nombre d'estimadors òptim varia.

6. Resultats

Finalment, tenint en compte tot el que s'ha exposat en l'apartat anterior, es pot concloure que l'algoritme òptim dels aplicats és Random Forests, amb un valor de correlació r^2 sempre superior a 0,97, i amb opció de variació respecte al nombre d'estimadors, ja que tots ells mantenen un alt valor r^2 amb petites diferències entre ells. La variació entre les versions normalitzades i sense normalitzar és mínima, però tenint en compte que en les darreres execucions l'opció sense normalitzar presenta valors r^2 millors per mil·lèsimes, es tria aquest model com a guanyador.

En la taula 6.1 i la figura 6.1, es presenta un extracte del què és la comparació de tots els valors predits amb els valors reals. A simple vista, ja destaca la similitud entre valors.

Taula 6-1.- Comparació d'alguns dels valors reals amb els valors predits per l'algoritme. Font pròpia, extreta directament de l'entorn de programació.

	Prediccions	Reals
0	457.8	463.0
1	231.1	231.0
2	560.8	575.0
3	595.0	578.0
4	526.9	512.0
...	...	...
2915	212.6	212.0
2916	599.1	574.0
2917	230.6	228.0
2918	241.1	239.0
2919	229.4	228.0

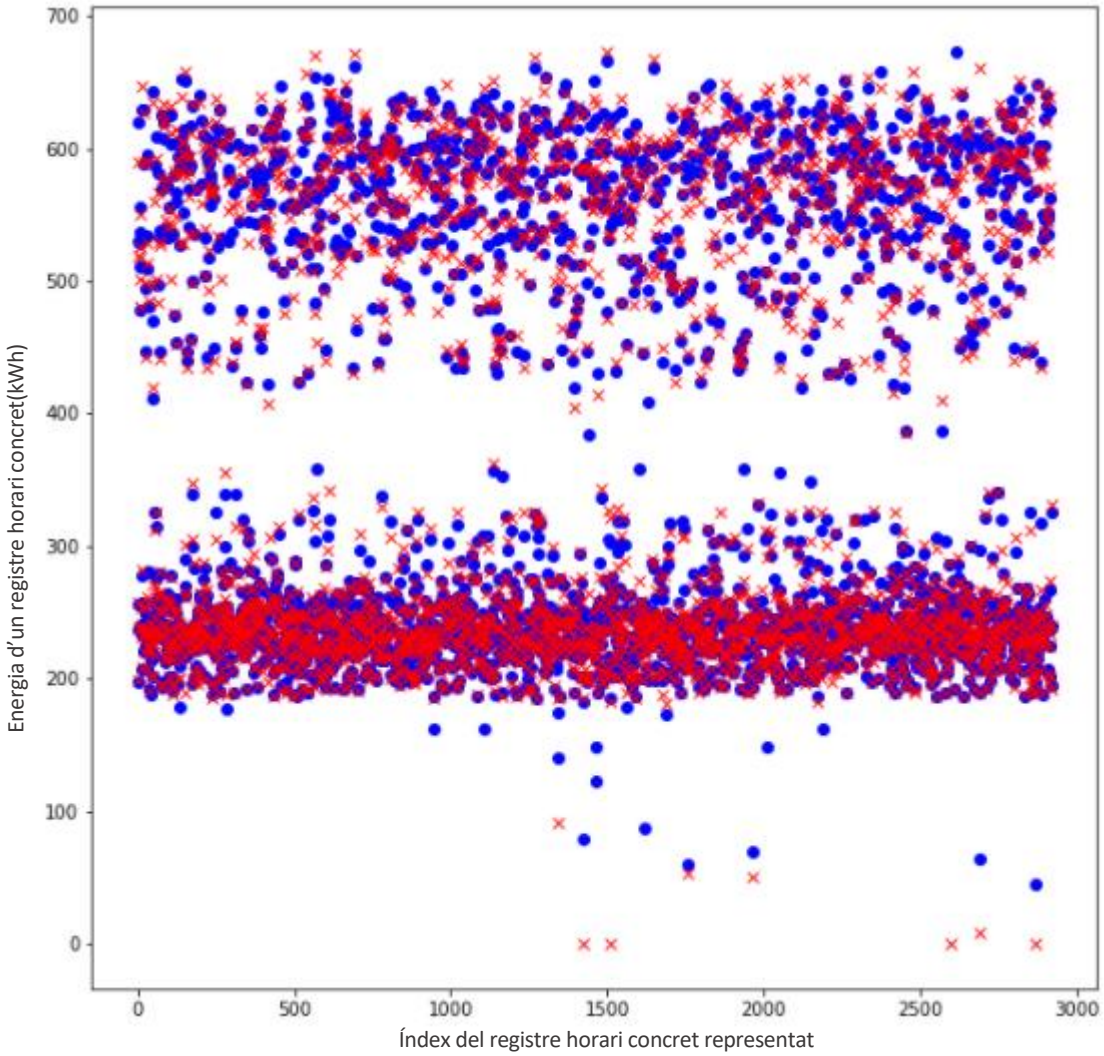


Figura 6-1.- Diagrama on s'hi representen els valors predits, amb punts blaus, i els valors reals, amb creus vermelles. Font pròpia, extret de l'entorn de programació.

En aquesta figura, es pot veure com el model presenta més error en la predicció d'aquells valors que surten dels dos rangs on es concentren la majoria de punts. Els rangs esmentats es poden observar de millor forma en la figura 6.2 i 6.3.

La causa d'aquest error en la predicció és la gran quantitat de mesures registrades dintre del rang entre 200 i 300 kWh que s'observa en la gràfica, que fan que el model tendeixi a agrupar més dades dintre d'aquest conjunt.

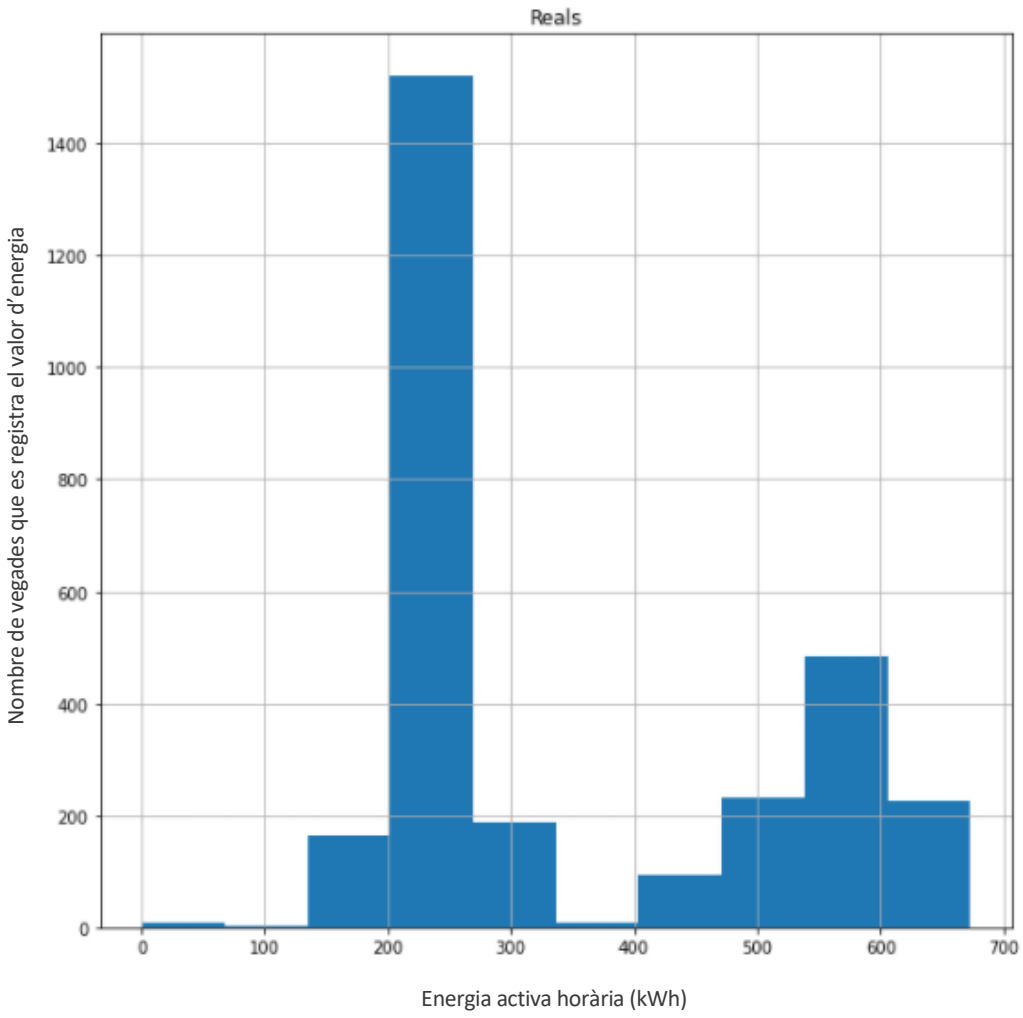


Figura 6-2.- Histograma del conjunt de dades reals d'energia de la part d'avaluació del model. Font pròpia.

Tal com es deduïa de la gràfica 6.1, el model concentra la majoria de prediccions en el conjunt entre 180 i 300 kWh aproximadament, fet causat pel motiu comentat en el paràgraf anterior. La resta de barres de l'histograma presenten força similitud en les seves magnituds entre totes dues gràfiques (6.2 i 6.3), fet que indica que la resta de dades es reparteixen de forma similar en els dos conjunts.

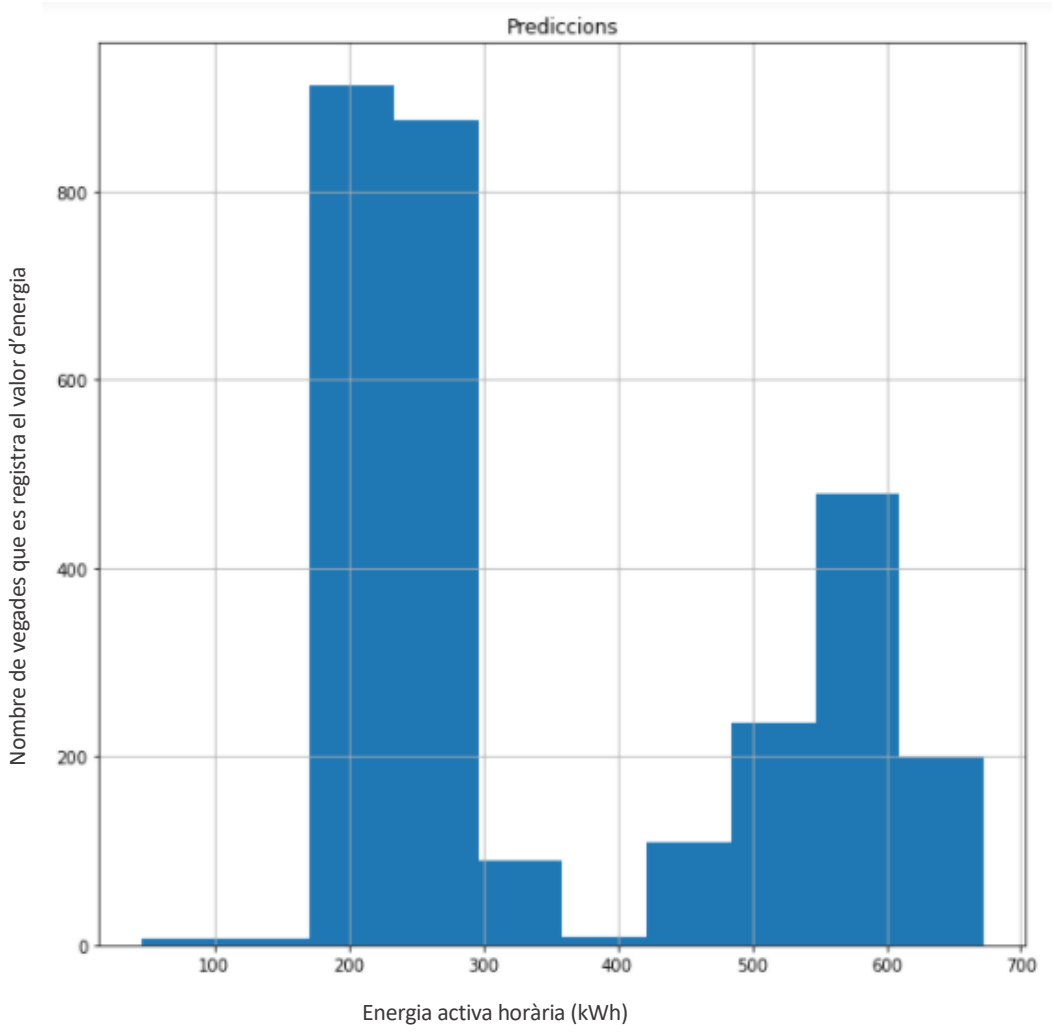


Figura 6-3.- Histograma del conjunt de prediccions de la part d'avaluació del model. Font pròpia.

Finalment, en la figura 6.4, es presenta el diagrama de caixa que compara les prediccions i els registres reals. La figura 6.4 i taula 6.2 permeten veure com el model ajusta de forma extraordinària aquells valors que marquen la tendència del conjunt de dades, com la mediana i els quartils Q1 i Q4, representats en la taula 6.2 com 0,50, 0,25 i 0,75 respectivament, però erra a l'hora de predir aquells valors extraordinaris, que en aquest cas, és 0, per algun registre determinat que consti com a 0. Aquest registre podria ser derivat d'un error en el preprocessing, però no hauria d'influir en el model en tant que sol s'observen 5 casos en la gràfica 6.1, dintre d'un conjunt de dades de 2920 registres.

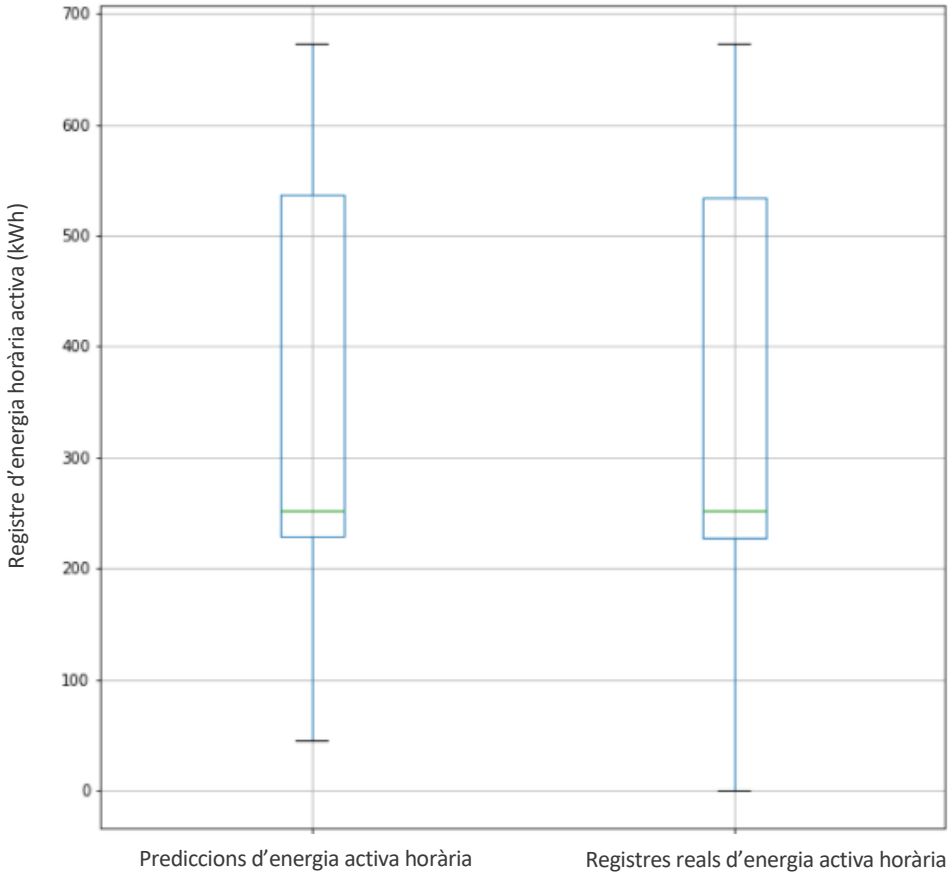


Figura 6-4.- Diagrames de caixa de les prediccions i dels registres reals. Font pròpia.

Taula 6-2.- Quartils corresponents al diagrama de caixa de la figura anterior. Font pròpia.

	Prediccions	Reals
0.00	45.600	0.00
0.25	228.275	228.00
0.50	252.050	252.00
0.75	536.825	534.25
1.00	672.200	673.00

Per tant, de forma sintetitzada, i tenint en compte els resultats exposats en aquest apartat, podem afirmar que l'algoritme òptim per aquesta aplicació és Random Forests, obtenint uns valors de correlació de fins a 0,998, i sense una verdadera rellevància del paràmetre "nombre d'estimadors" ni de la normalització de les dades, tot i que sembla que la no normalització porta a valors de r^2 lleugerament superiors.

7. Anàlisi de l'impacte ambiental

L'impacte ambiental d'aquest projecte, tot i girar entorn de l'anàlisi i predicció de la demanda elèctrica d'un campus, és gairebé nul. Es podria considerar l'impacte generat pel consum de l'ordinador on s'ha conduït l'estudi, però és menyspreable.

El fet que si podria afectar a l'impacte ambiental, i de forma favorable, seria les mesures que s'adoptin d'acord amb les conclusions extretes d'aquest treball, en tant que es podria optimitzar l'ús dels recursos disponibles per reduir la corba de consum, o bé millorar la gestió de la instal·lació fotovoltaica del campus i les bateries presents, per reduir el consum de xarxa i suplantar-lo amb producció d'energia neta, sense emissió de gasos d'efecte hivernacle.

Alhora, aquest estudi podria considerar-se suport del Pla UPC 2020 de sostenibilitat energètica, ja que forma part d'una de les seves línies estratègiques la implicació dels estudiants en l'estudi de l'eficiència energètica dels diferents campus.

Conclusions

En aquest treball s'ha elaborat un estudi al voltant de l'anàlisi de la demanda elèctrica del campus EEBE, per extreure'n conclusions que permetessin desenvolupar un model capaç de predir el consum de l'escola.

Per arribar a l'objectiu exposat, es fa una primera aproximació a l'estat de l'art del monitoratge del consum elèctric i l'ús d'intel·ligència artificial per la predicció de forma general, i també aplicada de forma específica en el camp de la gestió energètica. Amb això, s'han vist les eines principals que han permès posteriorment conduir aquest estudi.

Un cop es coneixen els mètodes i eines a emprar, s'ha fet una anàlisi del consum elèctric del campus en l'any 2019, per caracteritzar el comportament d'aquesta variable i alhora fer una primera aproximació a les dades a tractar. Aquesta revisió inicial ha permès descartar algunes hipòtesis, com la influència del calendari docent en el consum, ja que la presència d'alumnat no suposava variacions significatives en la demanda elèctrica, i també ha servit per intuir algunes de les variables que podien influir en el consum, com la festivitat d'una determinada jornada o bé si era cap de setmana.

Seguidament, s'ha desenvolupat el model de predicció, que amb les variables plantejades des d'un inici ja ha donat resultats força satisfactoris pel regressor "Random Forests", com es demostra en el capítol 6 d'aquesta memòria. Aquest algoritme, amb un coeficient de correlació entre predicció i registre real de r2 = 0,99, porta a concloure i demostrar que, efectivament, és possible realitzar una predicció de la demanda del campus, i així poder passar a nous estudis d'optimització energètica.

Tenint en compte el que s'ha exposat, l'objectiu d'aquest treball es considera com assolit, però s'ha de tenir en compte algunes limitacions al respecte, ja que per la predicció d'anys diferents de l'estudiat, s'haurien de reformular les equacions pel càlcul dels dies que són cap de setmana, i variar l'índex d'alguns festius, concretament els relacionats amb la Setmana Santa.

Pel que fa a futurs estudis derivats de l'aquí present, es podria considerar l'aplicació del model desenvolupat en altres anys, realitzant les petites variacions necessàries comentades en el paràgraf anterior, per així avaluar si realment s'adapta per la predicció del consum d'altres anys, fet que alhora serviria per veure si les dades varien notablement d'un any a altre. A més, tenint en compte que 2020 ha estat un any atípic, podria ser interessant veure la necessitat del model de noves variables, com per exemple el fet de ser període de confinament o no. Això portaria a poder avaluar si el model aquí desenvolupat podria servir per a una futura optimització de tota la gestió energètica del campus, conduint a incrementar la independència elèctrica del centre respecte a la xarxa per implementar fonts d'energia renovables aïllades.

Pressupost

El pressupost desglossat d'aquest projecte es presenta en les taules P-1 i P-2, on es pot diferenciar el cost dels recursos materials i el cost dels recursos humans respectivament.

Taula P-1.- Cost dels recursos materials. Font pròpia.

Concepte	Preu (€/unitat)	Amortització (mesos)	Temps d'ús (mesos)	Cost final imputable (€)
Ordinador portàtil	648,95	48	5	67,60
Llicència Office 365	69,00	12	5	28,75
SUBTOTAL				96,35
IVA (21%)				20,23
TOTAL				116,58

Taula P-2.- Cost dels recusos humans. Font pròpia.

Concepte	Preu (€/hora)	Hores de treball dedicades	Cost final imputable (€)
Documentació i anàlisi de dades	25	80	2000,00
Programació	25	140	3500,00
Redacció	25	70	1750,00
SUBTOTAL			7250,00
IVA (21%)			1522,50
TOTAL			8772,50

El preu final, obtingut realitzant el còmput del cost de recursos materials (116,58 €) i de recursos humans (8772,50 €), és de 8889,08 €.

Bibliografia

[1] AMAT RODRIGO, J. *Arboles de decision, Random Forest, Gradient Boosting y C5.0.* A: [en línia]. [Consulta: Octubre 2020]. Disponible a: https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting.

[2] CANCELO, J.R., ESPASA, A. I GRAFE, R., 2008. *Forecasting the electricity load from one day to one week ahead for the Spanish system operator*. A: International Journal of Forecasting. Vol. 24, núm. 4, p. 588-602. ISSN 01692070. DOI 10.1016/j.ijforecast.2008.07.005.

[3] DEXMA. *DEXMA - Software de Gestión y Eficiencia Energética.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://www.dexma.com/es/.

[4] FERRARA MARZO, A., 2019. *Estudi de la demanda de la instal·lació fotovoltaica aïllada de l ' ETSEIB.* A: [en línia]. [Consulta: Setembre 2020]. Disponible a: https://upcommons.upc.edu/handle/2117/167063

[5] GARRIGA GIBERT, E., 2020. *Estudi i predicció de la generació energètica de la instal· lació fotovoltaica de la biblioteca de l 'ETSEIB.* A: [en línia]. [Consulta: Setembre 2020] Disponible a: https://upcommons.upc.edu/handle/2117/186464

[6] GEMWEB. *Gestión y eficiencia energética - gemweb*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: http://www.gemweb.es/.

[7] HEIDMANN, L. *Unsupervised Machine Learning: Use Cases & Examples*. A: [en línia]. [Desembre 2020]. Disponible a: https://blog.dataiku.com/unsupervised-machine-learning-use-cases-examples.

[8] KURAMA, V. *Beginner's Guide To Unsupervised Learning With Python | Built In.* A: [en línia]. [Setembre 2020]. Disponible a: https://builtin.com/data-science/unsupervised-learning-python.

[9] MARIAS BARBER, L., 2019. *Anàlisi del consum elèctric a l'edifici H de l'ETSEIB*. A: [en línia]. [Consulta: Setembre 2020] Disponible a: https://upcommons.upc.edu/handle/2117/167448

[10] MARTÍNEZ HERAS, J. *Máquinas de Vectores de Soporte (SVM).* A: [en línia]. [Consulta: Novembre 2020]. Disponible a: https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/.

[11] MATHWORKS. *Machine Learning: Tres cosas que es necesario saber* - MATLAB & Simulink. A: [en línia]. [Consulta: Setembre 2020]. Disponible a: https://es.mathworks.com/discovery/machine-learning.html.

[12] MATPLOTLIB. *Pyplot tutorial — Matplotlib 3.3.3 documentation*. A: [en línia]. [Consulta: Octubre 2020]. Disponible a: https://matplotlib.org/tutorials/introductory/pyplot.html.

[13] MERKLE. *Cómo el Big Data puede decidir las elecciones de EEUU.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://www.merkleinc.com/es/es/blog/big-data-elecciones-eeuu.

[14] MORALES SÁNCHEZ, A.A. *Capítulo 3 Clasificadores Débiles-AdaBoost, Uso de características no lineales para identificar llantos de recién nacidos con un conjunto clasificador* [en línia]. Escuela de Ingeniería, Universidad de las Américas Puebla. Disponible a: http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/morales_s_aa/.

[15] NA8. *Algoritmo k-Nearest Neighbor | Aprende Machine Learning*. A: [en línia]. [Consulta: Septembre 2020]. Disponible a: https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/.

[16] ORELLANA ALVEAR, J., 2018. *Arboles de decision y Random Forest.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://bookdown.org/content/2031/arboles-de-decision-parte-i.html.

[17] PANDAS. *pandas - Python Data Analysis Library.* A: [en línia]. [Consulta: Septembre 2020]. Disponible a: https://pandas.pydata.org/.

[18] SCIKIT. *Decision Trees — scikit-learn 0.24.0 documentation*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://scikit-learn.org/stable/modules/tree.html#regression.

[19] SCIKIT. *Gradient Boosting regression — scikit-learn 0.24.0 documentation*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html.

[20] SCIKIT. *Nearest Neighbors — scikit-learn 0.24.0 documentation*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-regression.

[21] SCIKIT. *scikit-learn: machine learning in Python — scikit-learn 0.24.0 documentation*. A: [en línia]. [Consulta: Septembre 2020]. Disponible a: https://scikit-learn.org/stable/.

[22] SCIKIT. *sklearn.ensemble.AdaBoostRegressor — scikit-learn 0.24.0 documentation*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html.

[23] SCIKIT. *sklearn.linear_model.LinearRegression — scikit-learn 0.24.0 documentation.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html.

[24] SPAINML. C*omo funciona Gradient Boosting. A: [en línia]*. [Consulta: Desembre 2020]. Disponible a: https://spainml.com/blog/como-funciona-gradient-boosting/.

[25] UPC, UNIVERSITAT POLITÈCNICA DE CATALUNYA. *Butlletí UPC 166 — Legislació i normatives — UPC. Universitat Politècnica de Catalunya.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://www.upc.edu/normatives/ca/butlleti-upc/hemeroteca/2015-2016/butlleti-upc-166

[26] UPC, UNIVERSITAT POLITÈCNICA DE CATALUNYA, 2018. *Calendari Acadèmic EEBE curs 2018/2019.* A: [en línia]. [Consulta: Septembre 2020] Disponible a: https://eebe.upc.edu/ca/estudis/estudis-de-master/documents-masters/academic_calendar_2018_19.pdf

[27] UPC, UNIVERSITAT POLITÈCNICA DE CATALUNYA 2018. *Calendari Acadèmic EEBE curs 2019/2020*. A: [en línia]. [Consulta: Septembre 2020]. Disponible a: https://eebe.upc.edu/ca/estudis/calendaris-academics/documents/calendari_academic_19_20.pdf

[28] UPC, UNIVERSITAT POLITÈCNICA DE CATALUNYA. *Informe SIRENA 2019. Avaluació del consum d'energia i aigua de la UPC. — UPC Energia 2020 - Comunitats sostenibles — UPC. Universitat Politècnica de Catalunya.* A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://www.upc.edu/energia2020/ca/noticies/informe-sirena-2019-avaluacio-del-consum-d2019energia-i-aigua-de-la-upc

[29] WATTABIT. *Wattabit Energy Management Solutions | Soluciones energéticas*. A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://wattabit.com/

[30] ZAMBRANO, J. *¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del machine learning y la automatización inteligente*. A: [en línia]. [Consulta: Septembre 2020]. Disponible a: https://medium.com/@juanzambrano/aprendizaje-supervisado-o-no-supervisado-39ccf1fd6e7b

[31] ZEMSANIA, G.G. *Big Data Marketing, la analítica de datos al servicio de la publicidad* . A: [en línia]. [Consulta: Desembre 2020]. Disponible a: https://zemsaniaglobalgroup.com/big-data-marketing-la-analitica-de-datos-al-servicio-de-la-publicidad/

Annex A: Codi per l'aplicació de regressors

A1. Càrrega de dades

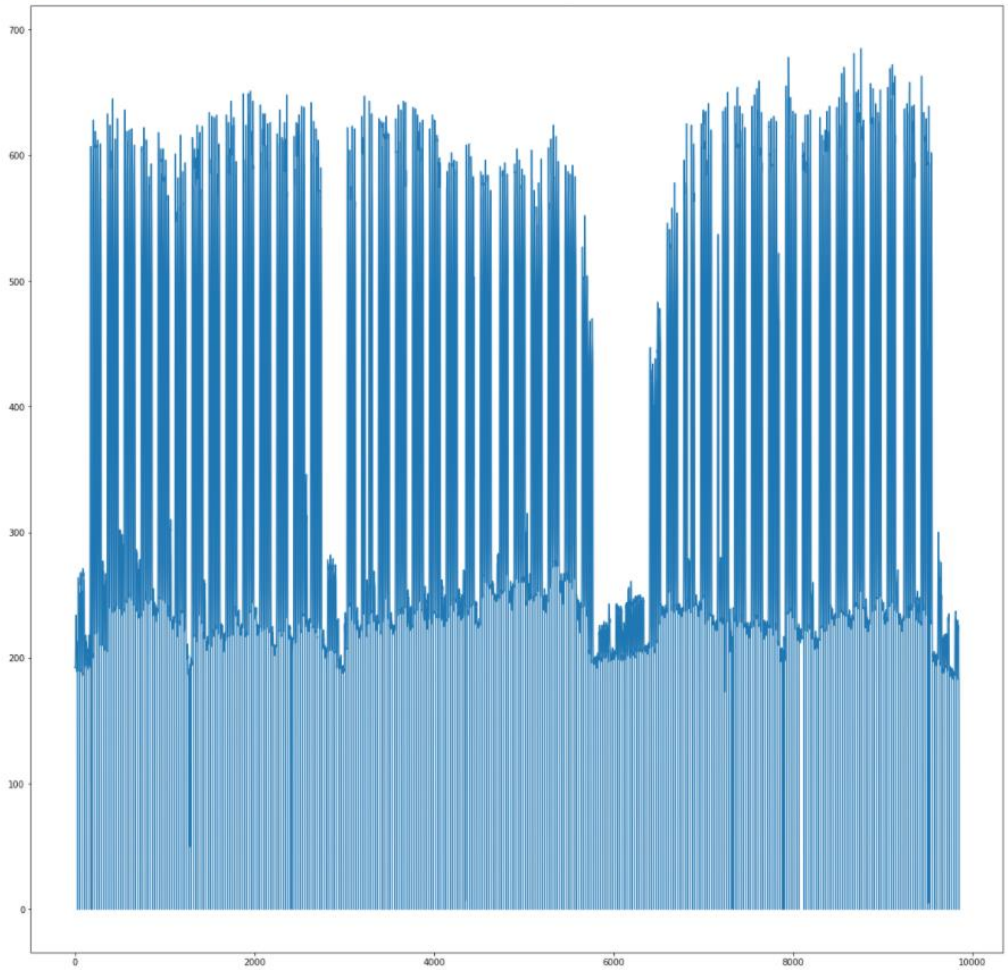
In [1]:
```
# Primer de tot, carreguem biblioteques necessàries
import pandas as pd
import numpy as np
import csv
import sys
import matplotlib.pyplot as plt
import numpy as np

# Ara, carreguem les dades en un pandas dataframe mitjançant l'arxiu en format csv.
P_horaria = pd.read_csv('Horaria de potencia activa COMES.csv')
P_horaria.shape
P_horaria.head()
```

Out[1]:

	Hora	Energia	Qualitat de la mesura
0	0.0	192.0	Real
1	1.0	193.0	Real
2	2.0	193.0	Real
3	3.0	192.0	Real
4	4.0	197.0	Real

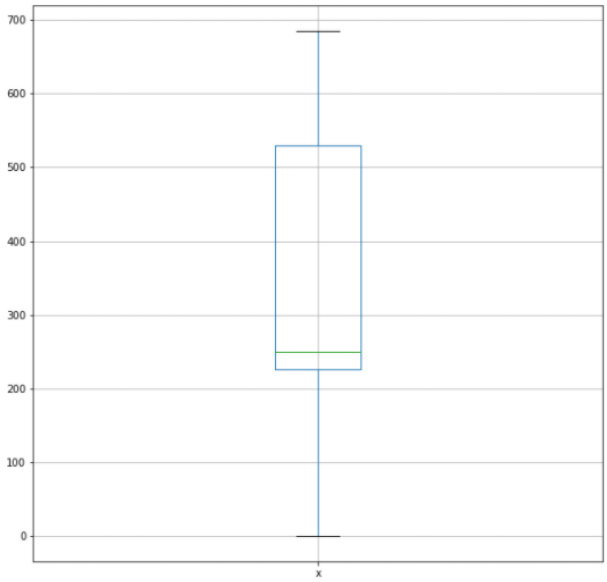
In [2]:
```
plt.figure(figsize=(20, 20))
plt.plot(list(P_horaria.index.values),list(P_horaria.iloc[:,1]))
```

Out[2]:
```
[<matplotlib.lines.Line2D at 0x1ac59fb5128>]
```



In [3]:
```
# Alternativa de representacio de les dades d'energia, per punts
#plt.figure(figsize=(20, 20))
#plt.plot(list(P_horaria.index.values),list(P_horaria.iloc[:,1]), 'bo')
```

In [4]:
```
#Afegim un boxplot per veure la distribució de les dades.
bp1 = pd.DataFrame.boxplot(P_horaria.iloc[:,1],figsize=(10,10))
```

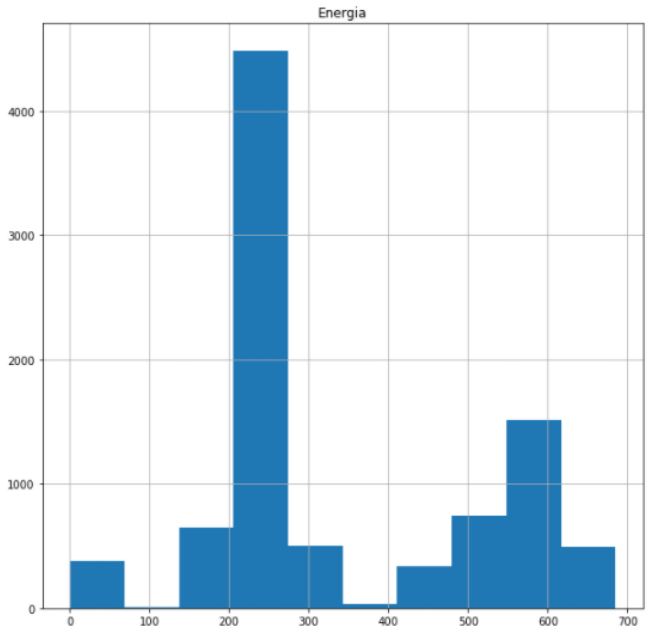


In [5]:
```
quantils_inicial = P_horaria.iloc[:,1].quantile([0.01,0.25,0.5,0.75,0.99])
quantils_inicial
```

Out[5]:
```
0.01      0.0
0.25    226.0
0.50    249.0
0.75    529.0
0.99    642.0
Name: Energia, dtype: float64
```

In [6]:
```
P_horaria.hist(column=['Energia'],figsize=(10,10))
```

Out[6]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001AC5A27F6D8>]],
      dtype=object)
```



In [7]:
```
#Per afegir una columna amb el nombre de dies mitjançant aquesta mateixa eina, creem un vector de 1 a 365
diesaux =np.arange(1,366)
dies = np.array([])
print(diesaux)
```

```
[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
 361 362 363 364 365]
```

In [8]:
```
# Com cada dia té 24 hores, però hi ha 25 files per dia, multipliquem cada valor del vector anterior [1,2,3...] perquè
# apareixi 25 cops.
for i in diesaux:
    for a in np.arange(25):
        #print(i)
        dies = np.append(dies,i)
#print(dies)
```

In [9]:
```
# Afegim aquesta nova columna al nostre dataframe. OJO! tenim 2 files buides al excel entre un dia i un altre, s'ha de tindre en
compte.
print(P_horaria.iloc[25])
P_horaria.dropna(axis=0,how='any',inplace=True)
print(P_horaria.iloc[25])
#Amb això, hem eliminat les files buides del DataFrame.
P_horaria['Dia'] = dies
display(P_horaria.head())
display(P_horaria.iloc[24])
print(P_horaria.iloc[25])
```

```
Hora                      NaN
Energia                   NaN
Qualitat de la mesura     NaN
Name: 25, dtype: object
Hora                        0
Energia                   190
Qualitat de la mesura    Real
Name: 27, dtype: object
```

	Hora	Energia	Qualitat de la mesura	Dia
0	0.0	192.0	Real	1.0
1	1.0	193.0	Real	1.0
2	2.0	193.0	Real	1.0
3	3.0	192.0	Real	1.0
4	4.0	197.0	Real	1.0

```
Hora                       24
Energia                     0
Qualitat de la mesura    Real
Dia                         1
Name: 24, dtype: object

Hora                        0
Energia                   190
Qualitat de la mesura    Real
Dia                         2
Name: 27, dtype: object
```

In [10]:
```
#Ara classificarem els dies segons si són en cap de setmana o no, una de les variables d'influència principals que s'obtenen
# en l'anàlisi gràfica de les dades.

#Creem dos vectors:
# En un, afegirem el nº de cada dia de l'any que és cap de setmana, i serà finde_aux.
# En l'altre, afegirem 1 o 0 segons si el dia és cap de setmana o no respectivament. S'anomenarà finde.
finde = np.array([])
finde_aux=np.array([])
for i in range(53):
    finde_aux = np.append(finde_aux,5+7*i)
    finde_aux = np.append(finde_aux,6+7*i)

for i in P_horaria['Dia'].values:
    if i in finde_aux:
        finde = np.append(finde,1)
    else:
        finde = np.append(finde,0)

# La linea de codi següent permet veure tot un array sencer, que si és massa llarg, pot mostrar-se truncat.
# np.set_printoptions(threshold=sys.maxsize)


#Finalment, afegim aquests valors a una nova columna anomenada "Cap de setmana"
P_horaria['Cap de setmana'] = finde

#Per comprovar que funciona, establim els parametres d'impressió de la matriu en il·limitats, i llavors imprimim.
# pd.set_option('display.max_rows', None)
# Hem establert el nombre màxim de files en "None" (no hi ha nombre màxim)
# pd.set_option('display.max_columns', None)
# pd.set_option('display.width', None)
# pd.set_option('display.max_colwidth', -1)
# P_horaria
```

In [11]:
```
#Què tenim per ara? Una taula amb la informació que podem veure a continuació.
P_horaria
```

Out[11]:

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana
0	0.0	192.0	Real	1.0	0.0
1	1.0	193.0	Real	1.0	0.0
2	2.0	193.0	Real	1.0	0.0
3	3.0	192.0	Real	1.0	0.0
4	4.0	197.0	Real	1.0	0.0
...	...	...	...	...	...
9848	20.0	198.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0
9849	21.0	189.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0
9850	22.0	183.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0
9851	23.0	183.0	No disponible	365.0	0.0
9852	24.0	0.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0

9125 rows × 5 columns

In [14]:
```
# Una altra de les variables influents en el consum energètic: Si es tracta d'un dia laborable o no. Afegirem una nova
# columna per classificar els dies en laborables o festius. Primer, creem un vector amb els dies festius. Com ho fem?
# Amb l'ajuda del calendari acadèmic, i sense comptar el fet que compti el febrer com a "festiu", tot i ser "no lectiu",
# S'agafa el nº de dia de l'any que és festiu. Per exemple, el 109 es correspon amb divendres sant, 19 d'abril.
# El dia 267 es la mercè (24/09), 227 el 15 d'agost.

festius_aux = np.array([1,6,109,112,121,161,175,227,254,267,285,305,340,342,359,360])
```

In [15]:
```
#Afegirem aquesta columna de festius.
#El codi és el mateix que per afegir els caps de setmana, estalviant el bucle que afegia 1 o 0 si era o no cap de setmana
festius = np.array([])

for i in P_horaria['Dia'].values:
    if i in festius_aux:
        festius = np.append(festius,1)
    else:
        festius = np.append(festius,0)

P_horaria['Festius'] = festius
P_horaria
```

Out[15]:

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana	Festius
0	0.0	192.0	Real	1.0	0.0	1.0
1	1.0	193.0	Real	1.0	0.0	1.0
2	2.0	193.0	Real	1.0	0.0	1.0
3	3.0	192.0	Real	1.0	0.0	1.0
4	4.0	197.0	Real	1.0	0.0	1.0
...	...	...	...	...	...	...
9848	20.0	198.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9849	21.0	189.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9850	22.0	183.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9851	23.0	183.0	No disponible	365.0	0.0	0.0
9852	24.0	0.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0

9125 rows × 6 columns

In [16]:
```
# Com que la fila de la hora 24 sempre és energia 0, la esborrarem en tots els dies, per netejar dades i alliberar memòria.
# (La fila de hora 0 és la mateixa i si té dades)

P_horaria[P_horaria['Hora'] != 24]
```

Out[16]:

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana	Festius
0	0.0	192.0	Real	1.0	0.0	1.0
1	1.0	193.0	Real	1.0	0.0	1.0
2	2.0	193.0	Real	1.0	0.0	1.0
3	3.0	192.0	Real	1.0	0.0	1.0
4	4.0	197.0	Real	1.0	0.0	1.0
...	...	...	...	...	...	...
9847	19.0	226.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9848	20.0	198.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9849	21.0	189.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9850	22.0	183.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9851	23.0	183.0	No disponible	365.0	0.0	0.0

8760 rows × 6 columns

In [17]:
```
P_horaria = P_horaria[P_horaria['Hora'] != 24]
```

In [18]:
```
P_horaria
```

Out[18]:

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana	Festius
0	0.0	192.0	Real	1.0	0.0	1.0
1	1.0	193.0	Real	1.0	0.0	1.0
2	2.0	193.0	Real	1.0	0.0	1.0
3	3.0	192.0	Real	1.0	0.0	1.0
4	4.0	197.0	Real	1.0	0.0	1.0
...	...	...	...	...	...	...
9847	19.0	226.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9848	20.0	198.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9849	21.0	189.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9850	22.0	183.0	Calculada a partir de la curva cuarto-horaria	365.0	0.0	0.0
9851	23.0	183.0	No disponible	365.0	0.0	0.0

8760 rows × 6 columns

A2. Normalització

Proves i normalització de les dades

In [19]:
```
#Realitzarem una barreja de les dades, per veure com es comporta per ara
from sklearn.utils import shuffle
P_horaria = pd.DataFrame(shuffle(P_horaria))
P_horaria.head()
```

Out[19]:

	Hora	Energia	Qualitat de la mesura	Dia	Cap de setmana	Festius
2606	14.0	230.0	Real	97.0	1.0	0.0
9618	6.0	255.0	Calculada a partir de la curva cuarto-horaria	357.0	0.0	0.0
4009	13.0	615.0	Calculada a partir de la curva cuarto-horaria	149.0	0.0	0.0
6188	5.0	201.0	No disponible	230.0	1.0	0.0
6723	0.0	237.0	Calculada a partir de la curva cuarto-horaria	250.0	1.0	0.0

In [20]:
```
#Establirem un llindar per tal de crear el nostre model. Alguns dies seran per configurar el model, i els altres, per fer la
#prova de si funciona o no.
llindar = P_horaria.shape[0]*2//3
llindar
```

Out[20]: 5840

In [21]:
```
# Tot seguit, crearem un nou dataframe, que contindrà exclusivament la variable Energia, que es la que pretem predir.
E_horaria = P_horaria.loc[:,'Energia']
# I ara, esborrem la variable Energia, que es la que volem predir, de la nostra taula inicial
del P_horaria['Energia']
P_horaria
```

Out[21]:

	Hora	Qualitat de la mesura	Dia	Cap de setmana	Festius
2606	14.0	Real	97.0	1.0	0.0
9618	6.0	Calculada a partir de la curva cuarto-horaria	357.0	0.0	0.0
4009	13.0	Calculada a partir de la curva cuarto-horaria	149.0	0.0	0.0
6188	5.0	No disponible	230.0	1.0	0.0
6723	0.0	Calculada a partir de la curva cuarto-horaria	250.0	1.0	0.0
...	...	...	...	...	...
9639	0.0	Calculada a partir de la curva cuarto-horaria	358.0	0.0	0.0
7218	9.0	Calculada a partir de la curva cuarto-horaria	268.0	0.0	0.0
3790	10.0	Real	141.0	0.0	0.0
111	3.0	Real	5.0	1.0	0.0
4228	16.0	Calculada a partir de la curva cuarto-horaria	157.0	0.0	0.0

8760 rows × 5 columns

In [22]:
```
# Ara, començarà el tractament estadístic de les dades. Per facilitar-lo, esborrarem la columna "Qualitat de la mesura",
# ja que no aportaria res al model.
del P_horaria['Qualitat de la mesura']
P_horaria
```

Out[22]:

	Hora	Dia	Cap de setmana	Festius
2606	14.0	97.0	1.0	0.0
9618	6.0	357.0	0.0	0.0
4009	13.0	149.0	0.0	0.0
6188	5.0	230.0	1.0	0.0
6723	0.0	250.0	1.0	0.0
...	...	...	...	...
9639	0.0	358.0	0.0	0.0
7218	9.0	268.0	0.0	0.0
3790	10.0	141.0	0.0	0.0
111	3.0	5.0	1.0	0.0
4228	16.0	157.0	0.0	0.0

8760 rows × 4 columns

In [23]:
```
mitja = P_horaria.iloc[:llindar,:].mean()
mitja
```

Out[23]:
```
Hora               11.496062
Dia               182.302397
Cap de setmana      0.284075
Festius             0.042123
dtype: float64
```

In [24]:
```
std = P_horaria.iloc[:llindar,:].std()
std
```

Out[24]:
```
Hora               6.941834
Dia              105.163721
Cap de setmana     0.451011
Festius            0.200888
dtype: float64
```

In [25]:
```
P_horaria2 = (P_horaria.iloc[:,:]-mitja)/std
P_horaria2.head()
```

Out[25]:

	Hora	Dia	Cap de setmana	Festius
2606	0.360703	-0.811139	1.587376	-0.209686
9618	-0.791731	1.661196	-0.629863	-0.209686
4009	0.216649	-0.316672	-0.629863	-0.209686
6188	-0.935785	0.453556	1.587376	-0.209686
6723	-1.656055	0.643735	1.587376	-0.209686

In [26]:
```
P_horaria2 = pd.concat([P_horaria2, E_horaria.T],axis=1)
P_horaria2.head()
```

Out[26]:

	Hora	Dia	Cap de setmana	Festius	Energia
2606	0.360703	-0.811139	1.587376	-0.209686	230.0
9618	-0.791731	1.661196	-0.629863	-0.209686	255.0
4009	0.216649	-0.316672	-0.629863	-0.209686	615.0
6188	-0.935785	0.453556	1.587376	-0.209686	201.0
6723	-1.656055	0.643735	1.587376	-0.209686	237.0

In [27]:
```
# Amb això, s'han normalitzat les dades i es podria passar a l'aplicació de models.
```

A3. Aplicació de models regressors

Aplicació de models

In [28]:
```
# Primer de tot, s'ha de definir la funció regressió.
#Per evaluar els models, utilitzarem els criteris de r2 (a maximitzar) i error quadràtic mig (a minimitzar)
from sklearn.metrics import r2_score, mean_squared_error
def regressio(nom,rgs):
    print(nom)
    rgs.fit(P_horaria2.iloc[:llindar,:-1],P_horaria2.iloc[:llindar,-1])

    prediccions = pd.Series(rgs.predict(P_horaria2.iloc[llindar:,:-1]), name = 'Prediccions')
    reals = pd.Series(P_horaria2.iloc[llindar:,-1], name='Reals')
    reals.index = range(P_horaria2.shape[0]-llindar)

    resultatr2 = round(r2_score(reals,prediccions),3)
    print('Error r2:',resultatr2)
    print('E.Q.M. :',round(mean_squared_error(reals,prediccions), 3))
    EQM = round(mean_squared_error(reals, prediccions), 3)


    return resultatr2, EQM, prediccions, reals
```

In [29]:
```
#També farem una taula de resultats, per comparar-los finalment.
noms = []
Resultats = []
```

In [30]:
```
%%time
from sklearn.linear_model import LinearRegression
result = regressio('Regressió Lineal:', LinearRegression())
#print(result)
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('Regressió Lineal')
```

```
Regressió Lineal:
Error r2: 0.293
E.Q.M. : 18032.817
Wall time: 140 ms
```

In [31]:
```
%%time
from sklearn.neighbors import KNeighborsRegressor
k = 1
result = regressio('kNN (k=1)', KNeighborsRegressor())
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('kNN (k=1)')
```

```
kNN (k=1)
Error r2: 0.937
E.Q.M. : 1599.14
Wall time: 137 ms
```

In [32]:
```
%%time
from sklearn.tree import DecisionTreeRegressor
result = regressio('Arbres de decisió:', DecisionTreeRegressor())
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('Arbres de decisió')
```

```
Arbres de decisió:
Error r2: 0.986
E.Q.M. : 364.162
Wall time: 104 ms
```

In [33]:
```
%%time
from sklearn.ensemble import RandomForestRegressor
result = regressio ('Random Forests:',RandomForestRegressor(n_estimators=10))
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('Random Forests')
```

```
Random Forests:
Error r2: 0.989
E.Q.M. : 278.609
Wall time: 230 ms
```

In [34]:
```
%%time
from sklearn.ensemble import AdaBoostRegressor
result = regressio('Adaboost:', AdaBoostRegressor(n_estimators=10))
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('Adaboost')
```

```
Adaboost:
Error r2: 0.679
E.Q.M. : 8175.381
Wall time: 103 ms
```

In [35]:
```
%%time
from sklearn.ensemble import GradientBoostingRegressor
result = regressio('Gradient Boosting:', GradientBoostingRegressor(n_estimators=10,max_depth=50))
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('Gradient Boosting')
```

```
Gradient Boosting:
Error r2: 0.871
E.Q.M. : 3299.147
Wall time: 894 ms
```

In [36]:
```
%%time
from sklearn.svm import SVR
result = regressio('SVR:',SVR(gamma='scale'))
Resultatstupla = (result[0],result[1])
Resultats.append(Resultatstupla)
noms.append('SVR')
# Resultats
```

```
SVR:
Error r2: 0.688
E.Q.M. : 7945.986
Wall time: 3.67 s
```

In [37]:
```
Resultatsdf = pd.DataFrame(data=Resultats,index=[noms],columns=['r2','mse'])
Resultatsdf
```

Out[37]:

	r2	mse
Regressió Lineal	0.293	18032.817
kNN (k=1)	0.937	1599.140
Arbres de decisió	0.986	364.162
Random Forests	0.989	278.609
Adaboost	0.679	8175.381
Gradient Boosting	0.871	3299.147
SVR	0.688	7945.986

In [39]:
```
%%time

# Per comparar resultats, farem el següent: 5 execucions per a cada nombre d'estimadors.
estimadors=[]
ResultatsRFS=[]
MSEs = []
Rcuads=[]
for i in [5,10,15,20,25]:
    Comparaciomitja= []
    Rcuadsmitja=[]
    for j in [1,2,3,4,5]:
        result = regressio('Random Forests (n='+str(i)+'), execució numero '+str(j)+':', RandomForestRegressor(n_estimators=i))
        ResultatsRFS.append(result)
        estimadors.append('Random Forests (n='+str(i)+'), execució numero '+str(j)+':')
        Comparaciomitja.append(result[1])
        Rcuadsmitja.append(result[0])
    MSEs.append(np.asarray(Comparaciomitja).mean())
    Rcuads.append(np.asarray(Rcuadsmitja).mean())

# Comparativa = pd.DataFrame(data=ResultatsRFS,index=[estimadors],columns=['r2','mse'])
ComparativaErr = pd.DataFrame(data=Rcuads,index=['Mitja per 5 estimadors','Mitja per 10 estimadors','Mitja per 15 estimadors','Mitja per 20 estimadors','Mitja per 25 estimadors'], columns=['r2'])
ComparativaErr['mse']=MSEs
ComparativaErr
```

In [41]:
```
%%time
#Tornem a executar el codi, i farem una gràfica per comparar realitat i predicció del model.
from sklearn.ensemble import RandomForestRegressor
result = regressio ('Random Forests:',RandomForestRegressor(n_estimators=10))
Resultats.append(result)
noms.append('Random Forests')

# pd.set_option('display.max_rows', None)
# pd.set_option('display.max_columns', None)
# pd.set_option('display.width', None)
# pd.set_option('display.max_colwidth', -1)
PredvsReals = pd.DataFrame(result[2])
PredvsReals['Reals'] = result[3]
PredvsReals
```

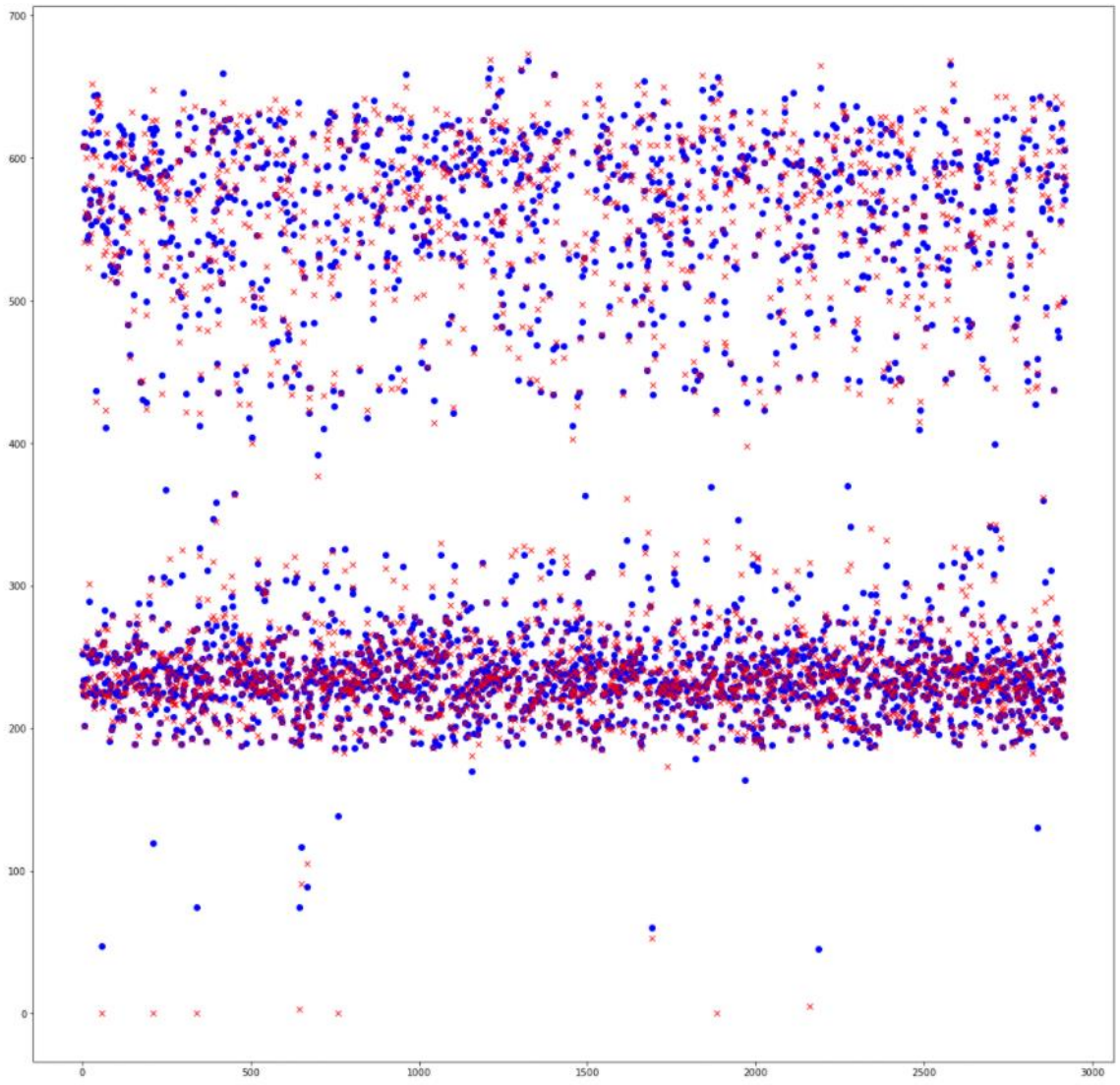
```
Random Forests:
Error r2: 0.99
E.Q.M. : 251.381
Wall time: 207 ms
```

Out[41]:

	Prediccions	Reals
0	229.5	227.0
1	252.0	254.0
2	233.7	231.0
3	608.0	608.0
4	223.8	226.0
...	...	...
2915	195.8	196.0
2916	571.1	583.0
2917	605.5	607.0
2918	194.4	195.0
2919	581.0	588.0

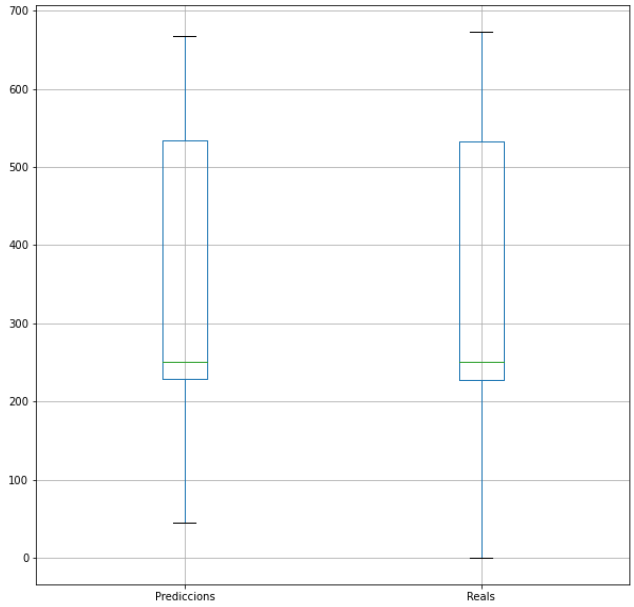
2920 rows × 2 columns

In [42]:
```
plt.figure(figsize=(20, 20))
#plt.plot(list(PredvsReals.index.values),list(PredvsReals.iloc[:,-1]), c='black', linewidth=0.5)
plt.plot(list(PredvsReals.index.values),list(PredvsReals.iloc[:,0]), 'bo')
plt.plot(list(PredvsReals.index.values),list(PredvsReals.iloc[:,-1]), 'rx')
```



In [44]:
```
pd.DataFrame.boxplot(PredvsReals,figsize=(10,10))
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b3d8d7c5f8>
```



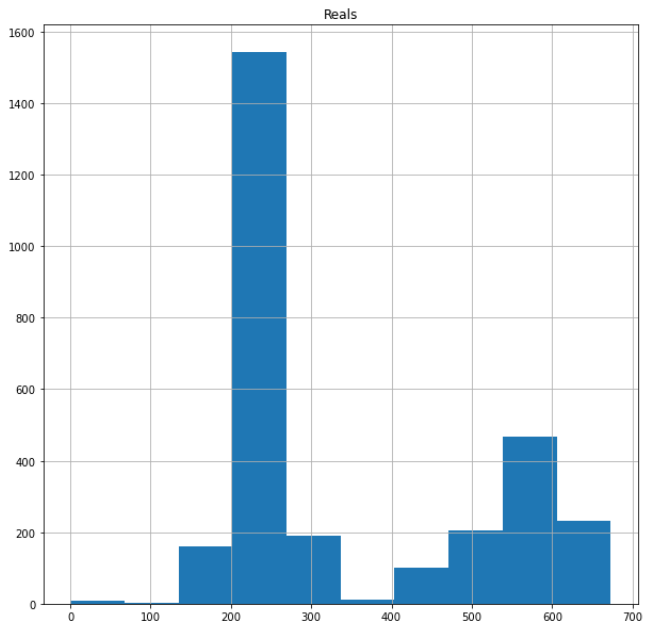
In [45]:
```
#Fem una taula amb la representació dels valors dels quartils per a cada boxplot, ja que a simple vista no s'observen
#degudament les diferències
quantils = PredvsReals.quantile([0.01,0.25,0.5,0.75,0.99])
quantils
```

Out[45]:

	Prediccions	Reals
0.01	188.400	189.0
0.25	228.600	228.0
0.50	251.200	251.0
0.75	533.425	532.0
0.99	638.429	639.0

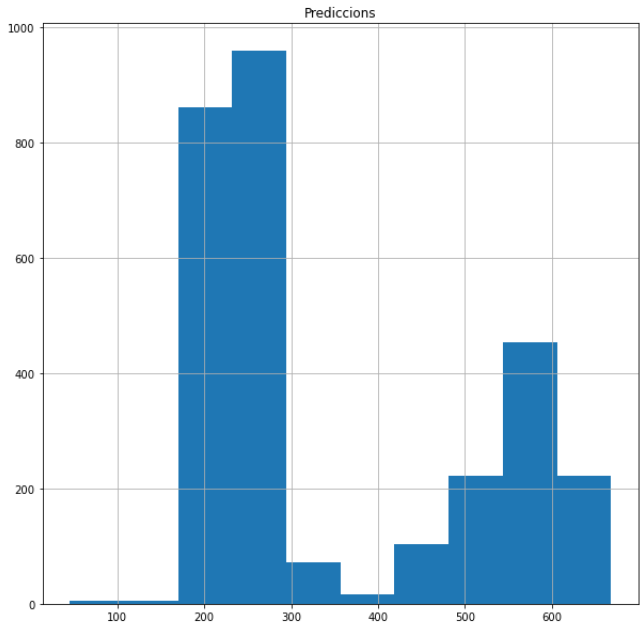
In [46]:
```
PredvsReals.hist(column=['Reals'],figsize=(10,10))
```

Out[46]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B3D8E522B0>]],
      dtype=object)
```



```
In [47]: PredvsReals.hist(column=['Prediccions'],figsize=(10,10))
```

```
Out[47]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B3D8EE4DA0>]],
       dtype=object)
```



Annex B. Gràfiques de la mitjana del consum energètic diari de cada mes

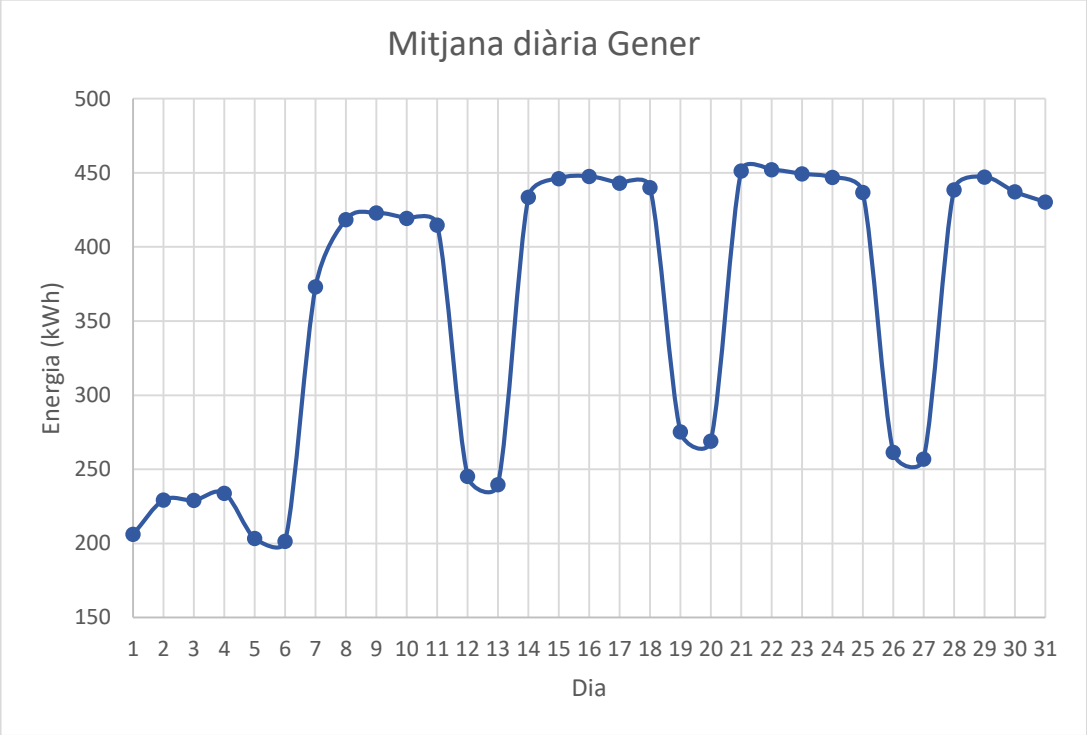


Figura B-0-1.- Corba de la mitjana del consum energètic diari del mes de Gener de 2019. Font pròpia.

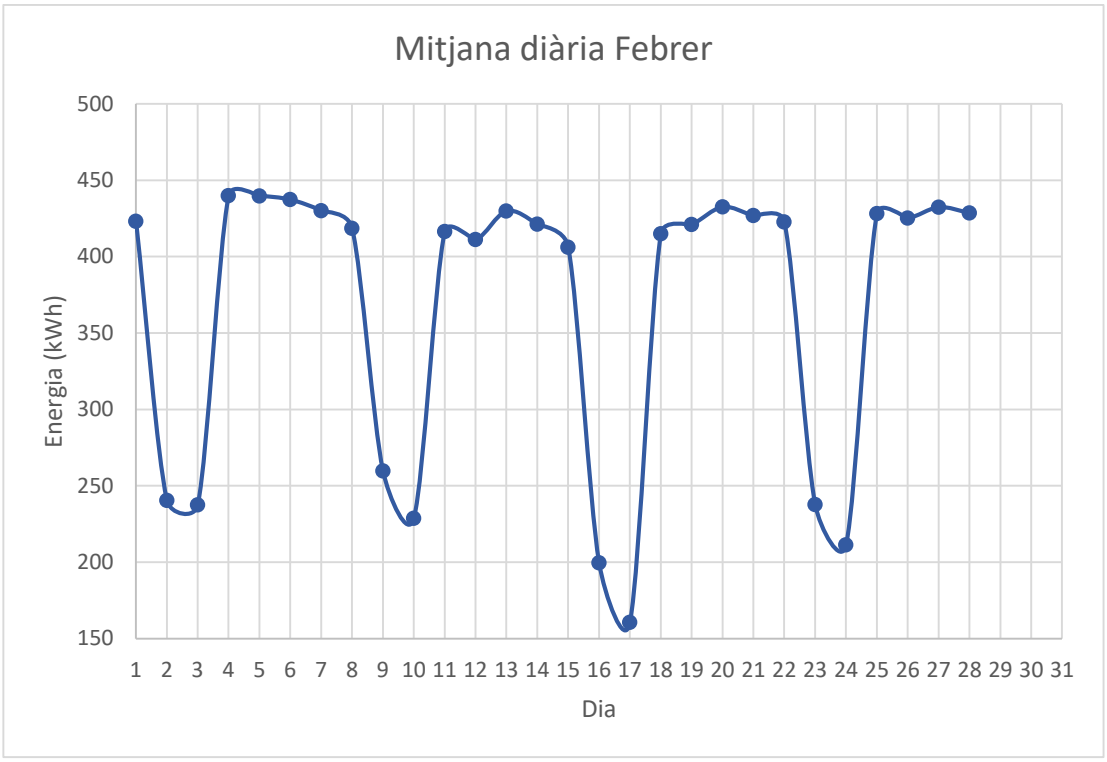


Figura B-0-2.- Corba de la mitjana del consum energètic diari del mes de Febrer de 2019. Font pròpia.

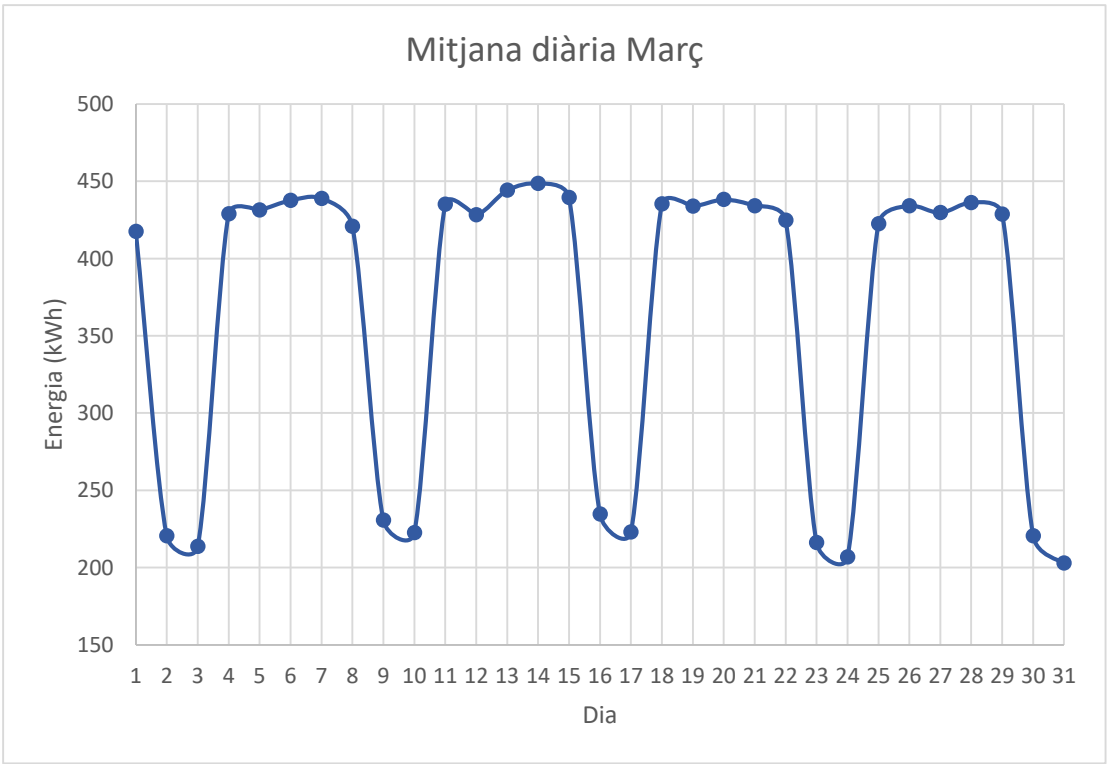


Figura B-0-3.- Corba de la mitjana del consum energètic diari del mes de Març de 2019. Font pròpia.

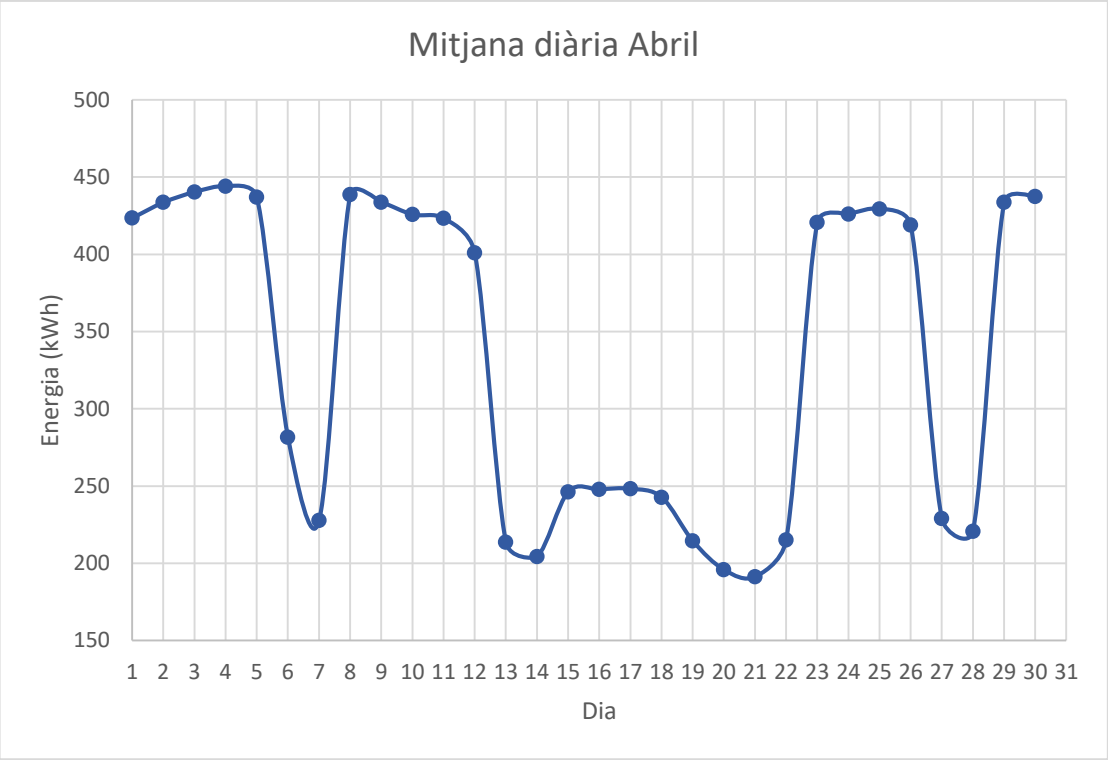


Figura B-0-4.- Corba de la mitjana del consum energètic diari del mes d'Abril de 2019. Font pròpia.

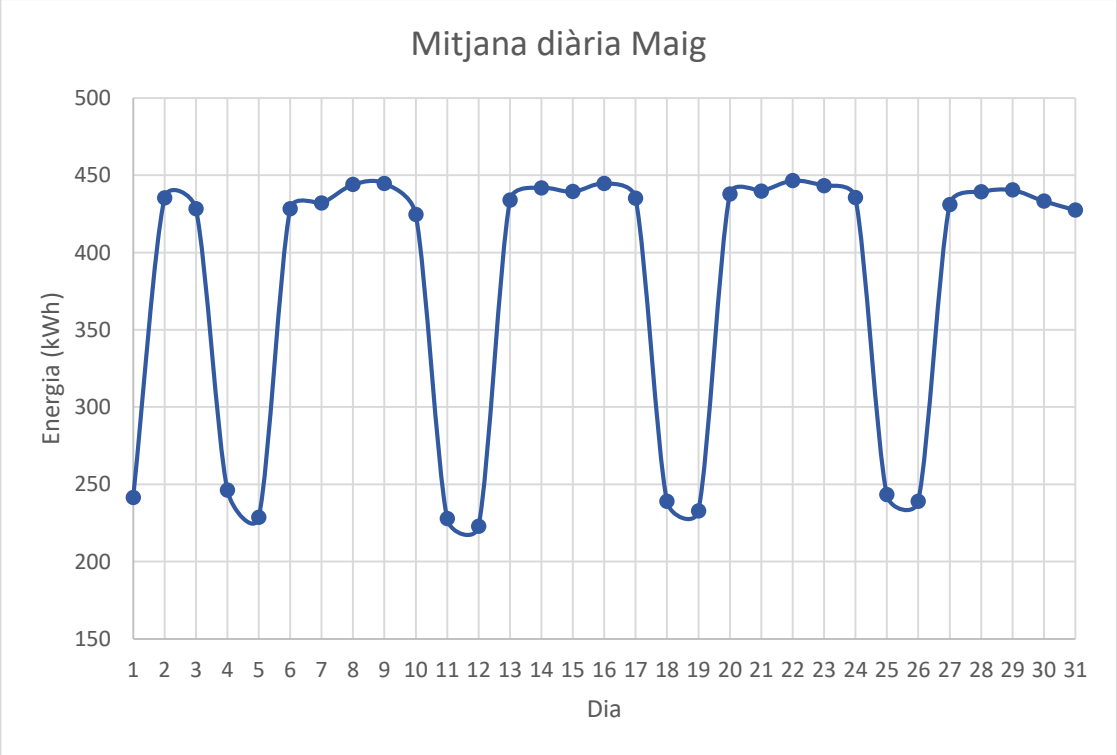


Figura B-0-5.- Corba de la mitjana del consum energètic diari del mes de Maig de 2019. Font pròpia.

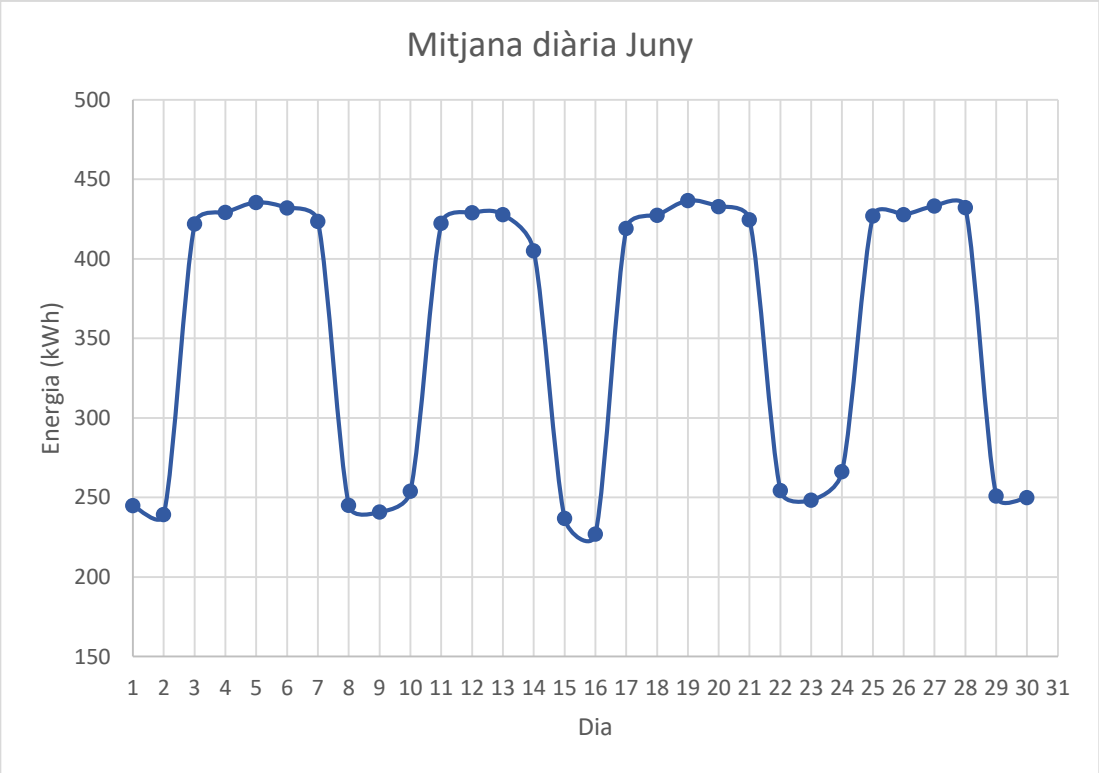


Figura B-0-6.- Corba de la mitjana del consum energètic diari del mes de Juny de 2019. Font pròpia.

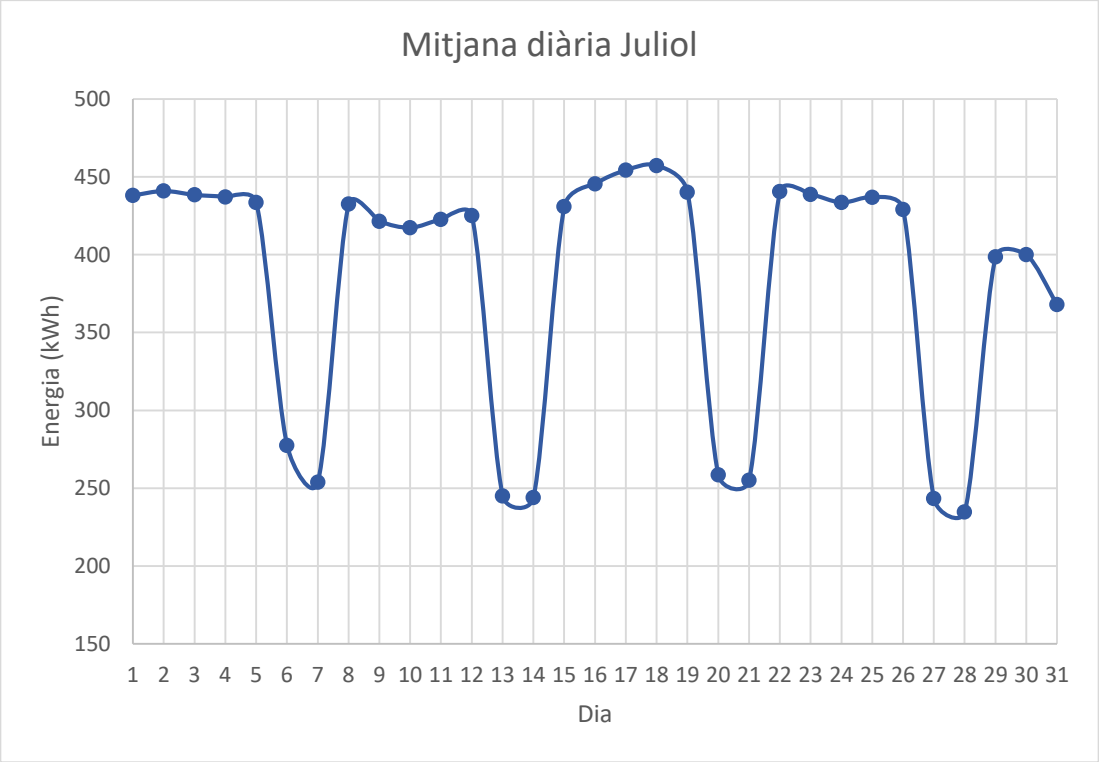


Figura B-0-7.- Corba de la mitjana del consum energètic diari del mes de Juliol de 2019. Font pròpia.

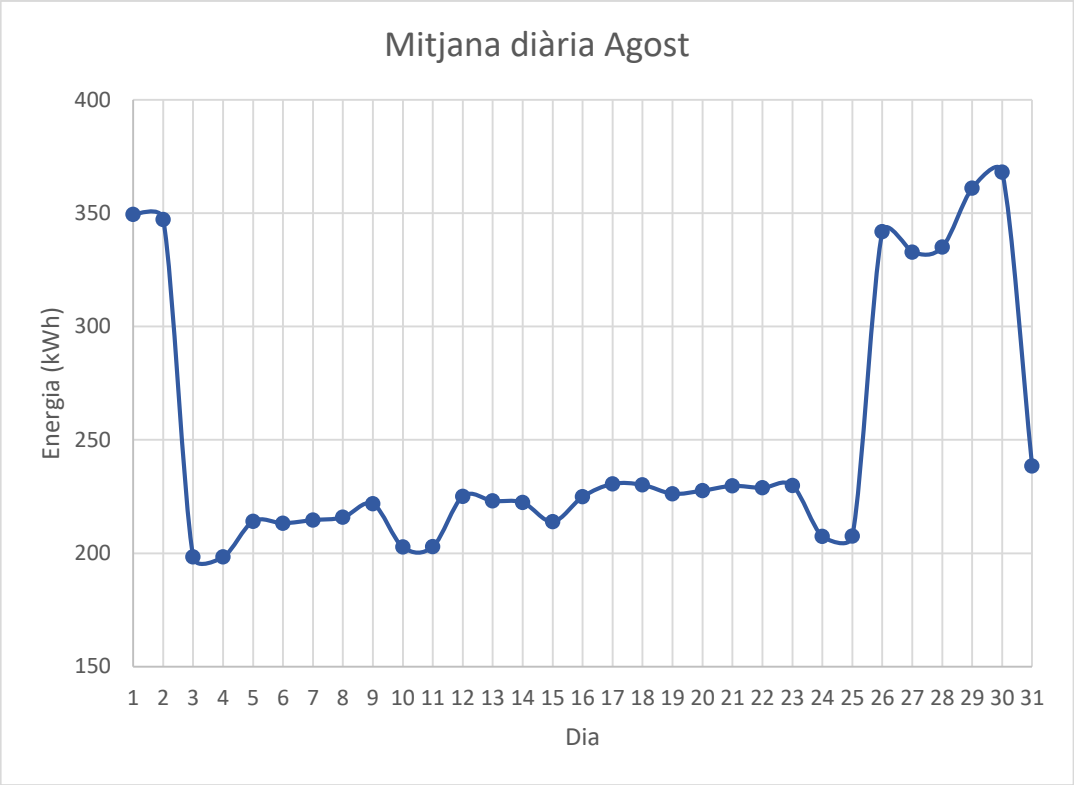


Figura B-0-8.- Corba de la mitjana del consum energètic diari del mes d'Agost de 2019. Font pròpia.

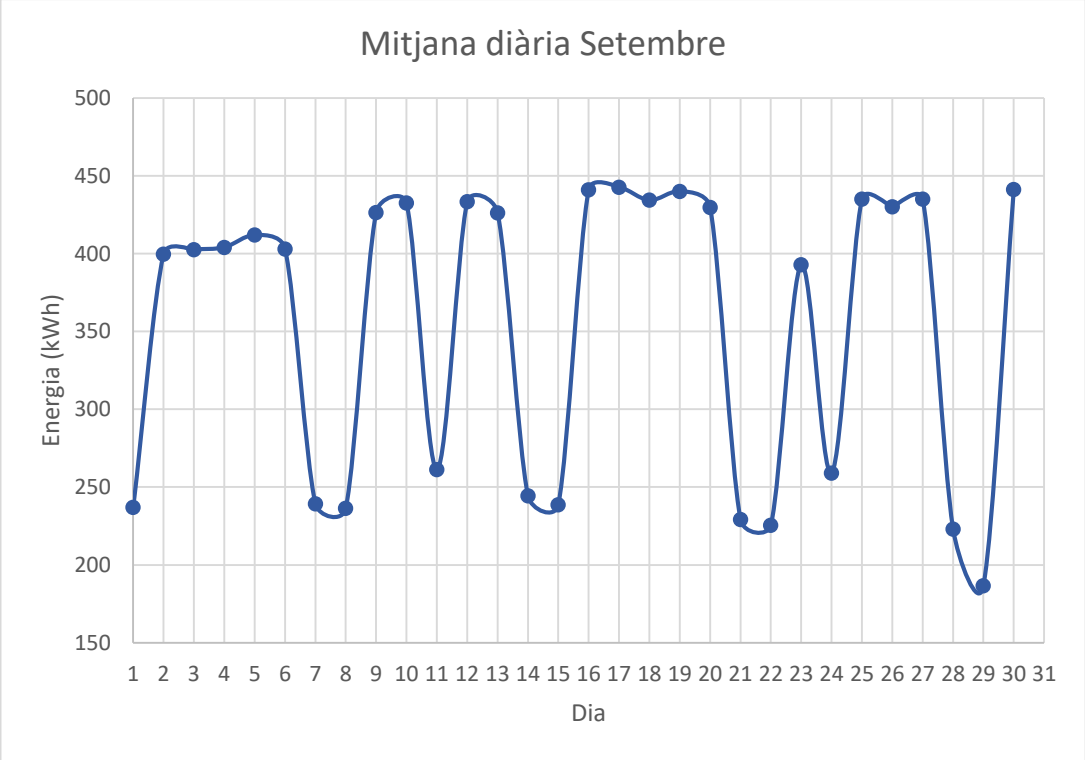


Figura B-0-9.- Corba de la mitjana del consum energètic diari del mes de Setembre de 2019. Font pròpia.

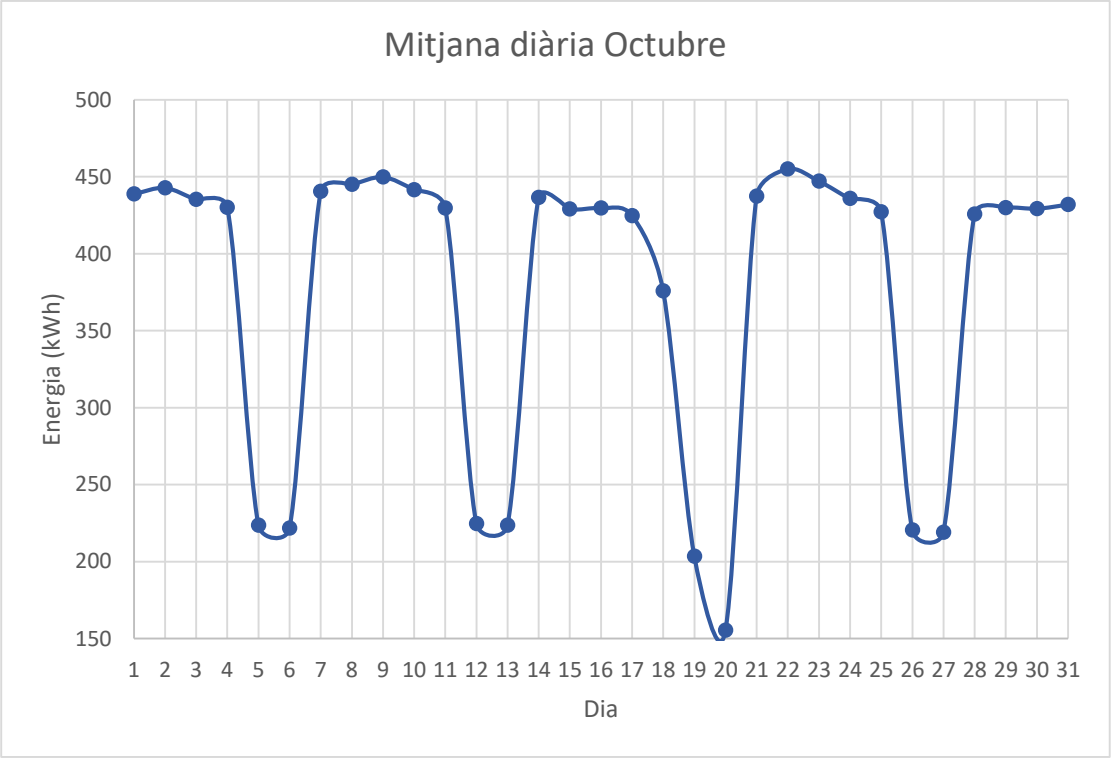


Figura B-0-10.- Corba de la mitjana del consum energètic diari del mes de Octubre de 2019. Font pròpia.

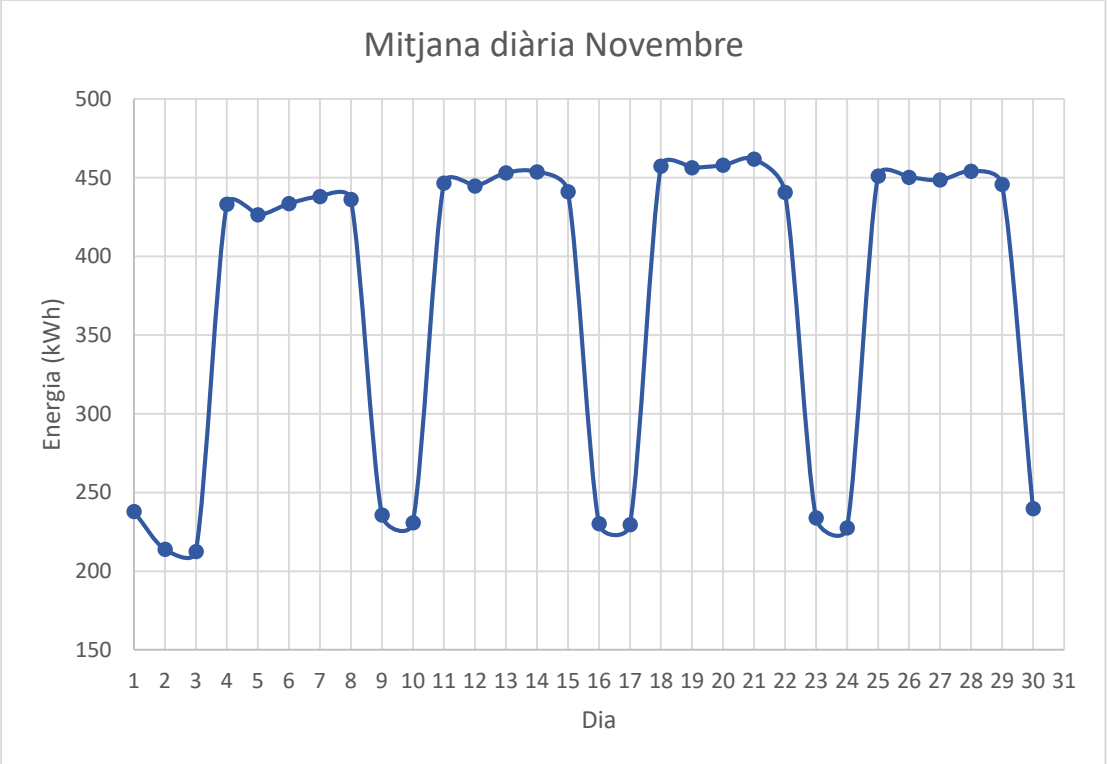


Figura B-0-11.- Corba de la mitjana del consum energètic diari del mes de Novembre de 2019. Font pròpia.

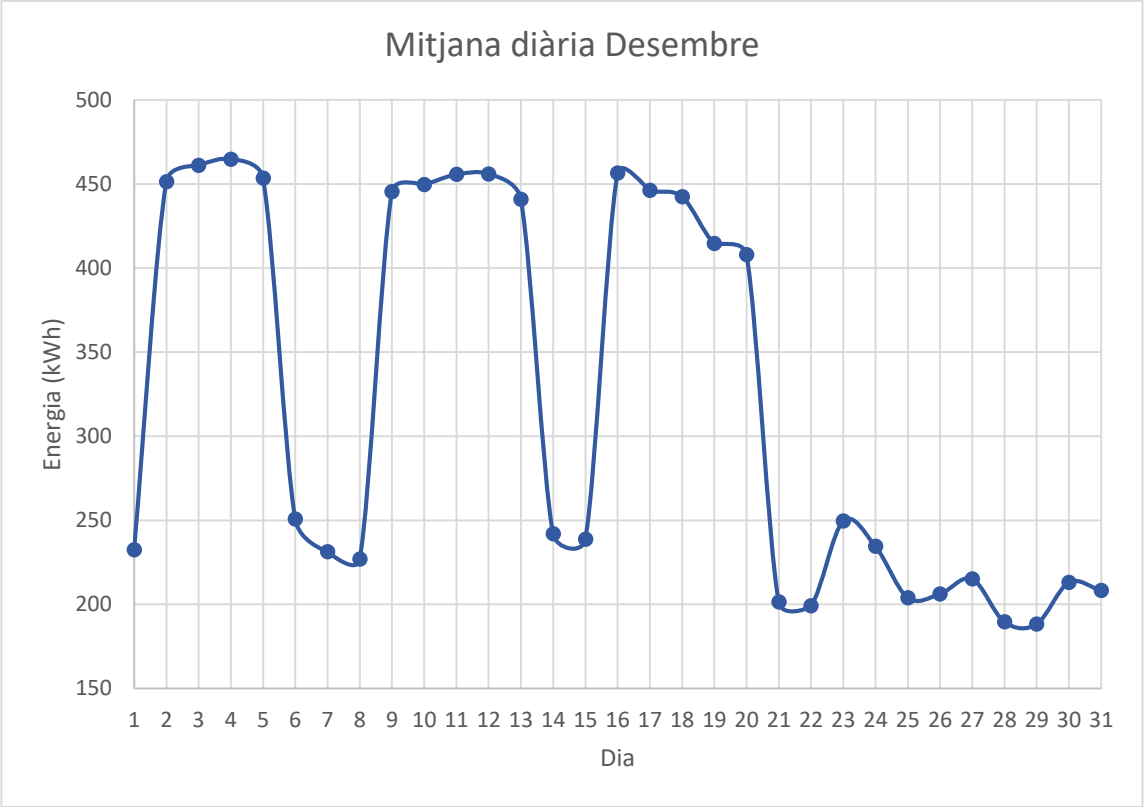


Figura B-0-12.- Corba de la mitjana del consum energètic diari del mes de Desembre de 2019. Font pròpia.