BACHELOR'S DEGREE THESIS

Degree in Statistics

Title: Classification model applied to fraud detection within the insurance sector

Author: Gerard Casas Zamorano

Advisor: Jose Antonio Sánchez Espigares

Department: Department of Statistics and Operational Research

Academic year: 2022-2023

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Acknowledgements

I would like to start expressing my deep gratitude to Zurich for having allowed me to make this project. In particular, I would like to give thanks to Laura Blanco and Alex Aguye for having supported me during the whole process of gathering data. We have been through different procedures, talks and legal discussions, and I would not have been able to carry out this research project without their contribution and push.

I am also grateful to my tutor Josep Anton Sanchez for their helpful discussions and insights. I have had the pleasure of having an excellent guide throughout the project, and thanks to his feedback I have been able to conclude it in the way I expected. I am also glad that thanks to the development of the present thesis I have discovered the interest that I really have in the field of text analytics.

Finally, I would like to extend my appreciation to my family and friends for their unwavering support and understanding during this process. Their love and encouragement has been a constant source of motivation for me.

Abstract

Fraud is a worrying activity that affects several factors as a whole. With the objective of addressing this issue adequately, it is necessary to build methods that detect and prevent this type of activity. In this project, we propose a text analysis tool for the detection of fraudulent behavior and two models for its prediction, and where we have made use of Zurich Insurance PLC data of 2020 and 2021 to conduct this study. The results that have been obtained demonstrate that there is a lot of variability in the data, where the prediction of fraud through the models has obtained an accuracy of around 63%. Nevertheless, the study that has been carried out has allowed us to extract some relevant conclusions, together with some considerations for future research.

Keywords: insurance sector, text analytics, fraud detection, generalized linear model, machine learning model, k-fold cross validation.

Resumen y palabras clave

El fraude es un tipo de actividad preocupante que afecta a varios factores en su conjunto. Con el objetivo de abordar adecuadamente este tema, es necesario construir métodos que detecten y prevengan este tipo de actividad. En este proyecto se expone una herramienta de análisis de texto para la detección de comportamientos fraudulentos y dos modelos para su predicción, y donde se ha hecho uso de datos de Zurich Insurance PLC de 2020 y 2021 para la realización del estudio. Los resultados que se han obtenido han demostrado que existe un nivel elevado de variabilidad en los datos, donde los modelos para la predicción del fraude han obtenido una precisión de alrededor del 63%. No obstante, el estudio realizado nos ha permitido extraer algunas conclusiones relevantes, así como algunas consideraciones para investigaciones futuras.

Palabras clave: sector asegurador, analítica de texto, detección de fraude, modelo lineal generalizado, modelo de machine learning, validación cruzada k-fold.

Table of Contents

CHAPTER I

Introduction

Text analysis is a type of study which mostly consists of extracting valuable information from paragraphs, sentences, words, or any type of text data. Its applications have been broadly extended since the beginning of its use, from the identification of patterns in the patient historical data within the healthcare industry to the collection of customers reviews on internet of a manufacturing company. Nevertheless, the area of activity that will be covered is fraud detection, and more specifically in the insurance sector.

Fraud detection is a subject of matter. Its identification and prevention allow the company to obtain a greater economic benefit, but more importantly, a greater ethical reputation. This fact is translated into a bigger trust from the customers, which generally facilitates their retention in the company. This is the reason why the identification of fraud is an extremely important activity for a company.

One type of fraud that usually occurs within the insurance industry is the issuance of new policies with improved conditions for the customers, which is carried out to prevent the customer from leaving the company. The background explanation for this type of fraud will be detailed in the following sections, although it is important to highlight that this particular case involves slight modifications of the addresses associated to the customers' policies, which will be the starting point for this work.

The project will be focused primarily on the detection of fraud committed through addresses associated to the policies thanks to a self-made text analysis tool developed using the R statistical program. On the other hand, once the fraud has been identified, there will be a second section that will be based in a model construction. The main aim is to compare two models: one of them will be a generalized linear model, and more precisely, a logistic model, and the other one consists of a machine learning model, both with the intention of making a prediction of fraud.

The construction of these two models leads to the objective of this project, which is the identification of what combination of variables makes this fraud more prone to be committed, so that the conditions under which a new policy is issued could suggest the presence of fraud, and on other hand, have the tools to predict it.

CHAPTER II

Research Context

2.1. Spanish Insurance Market

The Spanish insurance market has a different type of dynamics compared to other insurance markets when it comes to premiums management. When a new policy is issued for the first time, it is done with a premium which is lower than what it should in terms of profitability with the idea of boosting the attraction of customers. Nevertheless, as years go by, the premium is increased in a small percentage, so that the policy becomes profitable for the company in few years. To clarify this, it is shown a brief graphical example of a 5-year lifetime policy with its premium evolution (in a more real situation, additional increases are applied depending on the characteristics and profitability of the policy, but in order to understand the case in a simpler way, they will not be exposed in this project).

Example 1:

- Household policy: "00000000"
- % Increase Y-o-Y: 1.5%
- Break-Even Point: 300$ (annual premium paid to be profitable)
- Average: it shows the average premium paid in the past years

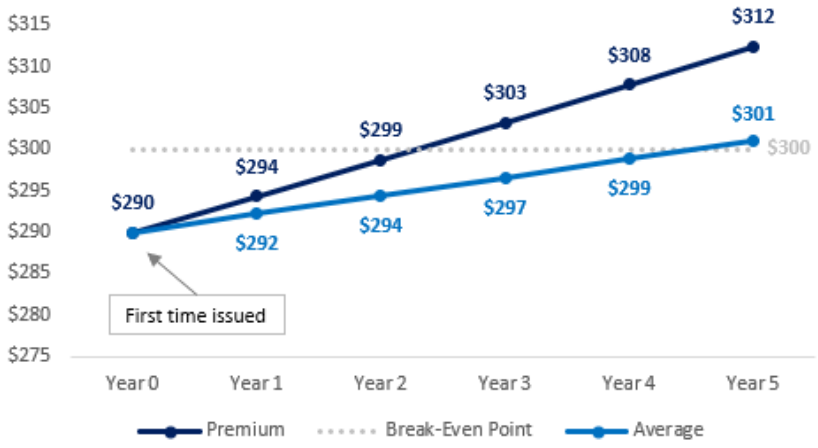

Figure 2.1: 5-Year policy lifetime

After having analyzed the graph, it is important to highlight the average premium obtained in year 5, which is 301$ > 300$. This data, therefore, reveals that policy "00000000" will be profitable from year 5 onward.

Nevertheless, if we pay attention to the evolution of the premiums over the years, we can notice that premium in year 0 is smaller than in other posterior year, and logically customers will be willing to pay premium at year 0 instead of premiums at any other year. Here is where fraud come over, but through intermediaries' actions.

2.2. Role of Intermediaries

Intermediaries are the main channels between customers and insurance companies' products. There are three different types of intermediaries: agents, brokers, and partners.

- Agents: it is a type of intermediary that sells exclusively insurance products of a particular company.
- Brokers: it is a type of intermediary that sells insurance products of different companies.
- Partners: it is a type of intermediary that takes advantage of its core activity to sell insurance products of a particular company. An example could be a bank, that along with a mortgage, it sells household insurances.

All of them deal directly with the end customer, although strategies to retain them are different. Agents for instance sell insurance products of a specific company, so their efforts must be more intense to retain customers. On the other hand, brokers for example, as they commercialize with other companies, they know that their actions to keep clients do not have to be that forceful because they could offer insurances from other companies.

Despite all of this, all of them have in common that they sell insurance products and, therefore, have an interest in profiting from them. They know that insurance companies' strategies must apply this annual percentage (together with other types of increases) to be profitable, so they have to find a way to justify this increase to customers and, if they are profitable, it can be offered a discount to not applying such a high increase. However, for every law there is a loophole.

2.3. Committed Fraud

Sometimes intermediaries do not have the necessary resources to retain customers and use alternatives not contemplated in the companies' rules to do it. There is a type of casuistry that is used in the insurance sector which is called making an authorized replacement. It occurs when the conditions of the policy are changed under authorization and, therefore, the premium is adjusted to these new conditions, which are variable depending on the company. It happens in particular cases and among the benefits obtained we can find a lower premium to be paid.

Nevertheless, on the other hand, there is the unauthorized replacement. It consists of issuing a new policy, with the same covers and guarantees, but making a slight change in the address associated to this policy, although it is still the same address. The issuance of this "new" policy makes the company system unable to recognize that it is in fact the same policy, so that it does not take into account its record. This situation leads to pay a premium as it is issued at year 0. Below there is an example of this type of replacement:

	Case 1	**Case 2**
Policy	"000000000"	"000000001"
Start date of the policy	29/10/2019	29/10/2022
End date of the policy	28/10/2022	-
Holder's ID	"123456789ABC"	"123456789ABC"
Address	**"Example 1 street, 2* 3a"**	**"Example one street, 2* 3a"**
Guarantees	"XXXXX"	"XXXXX"
Covers	"YYYYY"	"YYYYY"
Premium	303$	290$
No. claims during the period	4	0

Table 2.1: Fraud detection example

As we can see, the main insights that can be extracted from these cases are:

- Different number of policies.
- Start date of policy 2 is very close to the end date of policy 1.
- IDs are the same.
- Guarantees and covers are the same (it could happen that there are small differences).

- **The address 2 has a slight change that makes the system unable to identify that it is the same as address 1 (this is the main focus of a malicious replacement)**.
- As policy 2 has no record, the system cannot adjust a percentage increase to be applied to the paid premium due to the claim record.

It has been highlighted the addresses differences because if the intermediary tried to issue the new policy typing “Example 1 street, 2* 3a” instead of “Example one street, 2* 3a” as it is shown in case 2, then the system would not allow to proceed with this new issuance. This is the reason why the control of addresses plays an important role when it comes to the prevention of fraud.

Up to this point, it has been explained the context that there is behind the replacement fraud within the insurance industry, and how intermediaries’ actions are key for the development of this type of casuistry. It has been detailed the origin of these actions, which is based on a premium management focused on customer’s retention, and in addition, it has been exposed how an exhaustive control of addresses associated with the policies could lead to a containment of this type of fraud.

In the following section it will be explained the use of text analysis to identify this fraud, together with the main functions and code structured to perform this analysis.

CHAPTER III

Text Analysis

As indicated by [1], text analysis is “the process of using computer systems to read and understand human-written text for business insights”.

[1] also states that “the core of text analysis is training computer software to associate words with specific meanings and to understand the semantic context of unstructured data. This is similar to how humans learn a new language by associating words with objects, actions, and emotions”.

There is a different set of text analysis software that can be used in order to extract information of textual data, with their advantages and disadvantages. It seems reasonable to think that adding textual software to an analysis can be a source of added value, but sometimes companies cannot afford acquisition prices of these software, or they simply need solutions for very particular cases. In addition, a critical point could be the development of the logic that is built behind that software, either because it is a question of self-interest to know the reasoning it performs, or to understand that this logic fits the problem that needs to be treated. This is the reason why for the purposes of this project, it has been developed a text analysis code using R software to carry out this fraud identification.

Before moving on to the explanation of the code that allows this text analysis to be performed, it is important to dedicate a moment previously to explain the data the project is leaned on.

3.1. Data Set

The variables that will compose the data set are associated to the customer’s policy. These are:

- Policy (character).
- Intermediary (character).
- Type of building (character).
- Use of building (character).

- Years from construction (numeric): how many years have passed from the year the building was constructed.
- DT (character): it refers to the territorial directorate from which a policy is managed.
- Channel (character): it has been explained in previous chapters.
- Claims (numeric): number of claims associated to the policy.
- Loss (numeric): it refers to the loss that a specific policy has generated.
- Exposition (numeric): it means from 0 to 1 how much the policy has been exposed to the period*.
- Income (numeric): it refers to the income that a specific policy has generated within a period.
- Type of business (character): type of business under which a policy was issued.
- Product (character): type of technical product under which a policy is commercialized.
- Cancellation reason (character): the reason why the policy was cancelled.
- Continent capitals range (character): range of continent capitals covered in the policy.
- Content capitals range (character): range of content capitals covered in the policy.
- Housing location area (character): area in which the housing stipulated in the policy is located.
- Cluster (character): type of geographical variable that reveals information of the claim propensity and weather conditions.
- TOB (character): information related to the time that has passed from the issuance of the policy. TOB 1 means a "young" policy, while a TOB 5 means an "old" policy.
- Year (numeric): year analyzed.
- Fraud (numeric): if the policy is fraudulent or not (1 or 0, respectively).
- Old address (character): the old address associated to a policy.
- New address (character): the new address associated to a policy.

*If we are analyzing 2020, let's imagine a policy has been issued in May 2019. If this specific policy is then cancelled in May 2020, it will be said that its exposition to the period has been:

Exposition in 2020 of Policy X = $\frac{5}{12}$ = 0.41666

Since the data used has been anonymized, it will not be possible to explain the meaning of the different levels of character variables.

3.2. Code Logic

There are different specific text analysis functions that are used in the code, but the core idea that is built behind is based on a **% of coincidence**, which is made from a count of the letters present in addresses.

There are different sorts of information that do not add value to the coincidence address exercise because it is believed that will be included either the fraud is committed or not. An example of this could be the word "street". So, in order to go to the subject of matter, we will focus on information that is susceptible to suffer little changes.

In order to understand more deeply the code operations, an example is shown below:

Example 2:

Let's suppose there are two policies whose addresses are intended to be analyzed:

A1: "Example one street, 2* 3a"

A2: "Example 1 street, 2* 3a"

- Step 1: Remove general words. These are structures, words, or combination of words that do not make the difference between one direction and another.

A1: "Example one ~~street~~, 2* 3a"

A2: "Example 1 ~~street~~, 2* 3a"

A1: "Example one, 2* 3a"

A2: "Example 1, 2* 3a"

- Step 2: Make all letters lowercase. Since R software is case sensitive, it is important to adapt all letters to the same type.

A1: "~~E~~xample one, 2* 3a"

A2: "~~E~~xample 1, 2* 3a"

A1: “example one, 2* 3a”

A2: “example 1, 2* 3a”

- Step 3: Remove numbers. If the policy is intended to have the same address, numbers cannot be changed.

A1: “example one, ~~2~~* ~~3~~a”

A2: “example ~~1~~, ~~2~~* ~~3~~a”

A1: “example one, * a”

A2: “example , * a”

- Step 4: Remove symbols. It includes commas, periods, dashes, apostrophes, exclamation signs…

A1: “Example one a”

A2: “Example a”

- Step 5: Remove blanks. R recognizes blanks, so they need to be removed.

A1: “Exampleonea”

A2: “Examplea”

Once the debugging of the addresses is made, then we can proceed with the count of the letters. It is performed through a matrix like this:

Letters present in addresses	**A1**	**A2**	**% Coincidence (letter)**
A	2	2	1
E	3	2	0.667
M	1	1	1
P	1	1	1
L	1	1	1
X	1	1	1
O	1	0	0
N	1	0	0
Total	**11**	**8**	**5.667**

Table 3.1: % Coincidence Example

$Coincidence\ (letter)_i = \frac{min(A1_i, A2_i)}{max(A1_i, A2_i)}$, $\forall\ i \in$ letters of addresses

$Coincidence\ (addresses) = \frac{\sum_{i=1}^{n} Coincidence(letter)_i}{n}$ x 100 , where n is the total number of letters present in addresses

In this case:

$Coincidence\ (addresses) = \frac{5.667}{8}$ x 100 = 70.83%

The conclusion of this analysis reveals that addresses A1 and A2 have a 70.83% of coincidence. Nevertheless, the point here is: "What is the minimum % of coincidence from which we can consider that fraud has been committed?"

This question will be discussed in the next chapter, where it will be explained the classification logic that it has been used in order to determine if a policy is fraudulent or not.

3.3. Detection of Fraud

In order to make effective the identification of fraud, it has been said in the previous chapter that a minimum % of coincidence needs to be set up. Before assessing if that % of coincidence classifies correctly or not, it is important to mention that we must make use of a confusion matrix to analyze the results obtained. Find below what [2] states:

In statistics and machine learning arena, classification is a problem of labeling an observation from a finite number of possible classes. Binary classification is a special case of classification problem, where the number of possible labels is two. It is a task of labeling an observation from two possible labels. The dependent variable represents one of two conceptually opposed values (often coded with 0 and 1), for example:

- The outcome of an experiment - pass (1) or fail (0)
- The response of a question - yes (1) or no (0)
- Presence of some feature - absent (0) or present (1)

There can be four cases of a certain observation:

1. The response actually negative, the algorithm predicts it to be negative. This is known as true negative (TN).

2. The response actually negative, the algorithm predicts it to be positive. This is known as false positive (FP).
3. The response actually positive, the algorithm predicts it to be positive. This is known as true positive (TP).
4. The response actually positive, the algorithm predicts it to be negative. This is known as false negative (FN).

All the observations fall into one of the four categories stated above and form a **confusion matrix**.

	Predicted Negative (0)	**Predicted Positive (1)**
Actual Negative (0)	True Negative (TN)	False Positive (FP)
Actual Positive (1)	False Negative (FN)	True Positive (TP)

Table 3.2: General Confusion Matrix

If a confusion matrix is applied to the identification of fraud through the text analytics code, we obtain:

	Predicted Non-Fraudulent	**Predicted Fraudulent**
Actual Non-Fraudulent	True Negative (TN)	False Positive (FP)
Actual Fraudulent	False Negative (FN)	True Positive (TP)

Table 3.3: Case-Oriented Confusion Matrix

False negative and false positive are two types of mistakes whose repercussions are different:

- False Negative: fraud is not identified when there is actual fraud. It could lead to a very low impact on financial statements (bearing in mind the data set is based on property personal lines segment).
- False Positive: fraud is identified when the policy is not actually fraudulent. The identification usually leads to a clean-up.

The repercussions of both types of mistakes are different. False positive error leads to a clean-up of the policy, which means that this specific policy will be removed from the portfolio. The repercussion of this mistake is heavier than false negative, since not only could affect the quality of the policy portfolio, but also the company's reputation. In the end, we are accusing an intermediary of committing fraud, when it is not actually the case. Having said

that, it will be important to put more attention on protecting false positives of being committed, so that this type of mistake is 0.

In order to validate if the textual analysis algorithm classifies correctly, it has been extracted a sample of around 5% of the total observations that reveal if a policy is actually fraudulent or not.

3.4. Results

3.4.1. Exploratory Analysis

The data associated to the addresses contain information of those policies that are susceptible of having committed fraud. There are two data sets that will be under analysis: A2020 and A2021. Both contain the same information, although A2020 has data of 2020 and A2021 has data of 2021. Below it is shown some observations of both data sets:

```
## Table 3.4: Data of A2020
## ================================================
##       Policy         Old_Address       New_Address
## ------------------------------------------------
## 1 243,564,906,567 ssarenrcoubmetml     emmtnrsorua
## 2 243,585,653,567      duraeco           qdoerau
## 3 243,605,022,567       ianes         alvooiaonctn
## 4 243,672,568,567 nlaoaniomaÃ±duaidg nogolinudaiÃ±amdaa
## 5 243,678,417,567     iaistudnr          naiuitdrs
## 6 243,688,085,567      yunrfto           vebrota
## ------------------------------------------------
##
## Table 3.5: Data of A2021
## ================================================
##       Policy      Old_Address     New_Address
## ------------------------------------------------
## 1 432,580,506,567   rirrvaod        aanarrv
## 2 432,580,506,567   rorrviad      orvriartcdoo
## 3 432,584,276,567  rbdaarilna      ardlribnaa
## 4 432,628,065,567 cenueirrirad      areudir
## 5 432,706,630,567 hduioccrcot  ardrioaeclesipsian
## 6 432,942,813,567 baeetcosrquÂº        rtea
## ------------------------------------------------
```

A2020 contains 5929 rows and 3 columns, while A2021 contains 6097 rows and 3 columns. Both data sets contain 3 variables: “Policy”, which is showed in a numeric format, while “Old_Address” and “New_Address” are in a character format. It has also been observed that there is no missing data present neither in A2020 nor in A2021 (find attached code in Appendix A). Since these data sets contain actual addresses of policies, for the purposes of the analysis they have been through a process of randomization.

3.4.2. Fraud Identification

We will now focus on the whole procedure that leads to the fraudulent policy identification. It begins with the analysis of some samples, which contain around 5% of the original data sets with the actual information of fraud both for 2020 and 2021. These samples reveal if the policy is actually fraudulent, and it will be the starting point to decide which % of coincidence must be attributed to each data set with the intention of deciding whether a policy is fraudulent or not.

Since we have dealt with sensitive data, the whole procedure that has been explained in section 3.2. has already been applied. Therefore, fraud identification in this case will only be based on the recognition of letters and the computation of the % of coincidence. Nevertheless, find attached the code below in order to see how it was structured at first:

```
D$Coincidence <- NA

words1 <-
c("c//.","c\\.","pza\\.","pl\\.","plza.","rbla\\.","rd\\.","ctra\\.")

words2 <-
c("paseo","plz","numero","cami","pasaje","urb","edf","carrer","rua","ka
lea","avenida","avda","avinguda","piso","paseo","cuesta","callejon","ca
mpillo",

"travesia","carretera","karratu","ibili","etorbidea","parc","edifici","
edificio","subida")

DIguales <- subset(D, New_Adress==Old_Adress)
DTotal <- D
D <- subset(D, New_Adress!=Old_Adress)

for (i in 1:nrow(D)) {
```

```
  Documento1 <- Corpus(VectorSource(D$New_Adress[i]))
  suppressWarnings(doclimpio1 <- tm_map(Documento1, removeWords,
words1))
  suppressWarnings(doclimpio1 <- tm_map(doclimpio1,
content_transformer(tolower)))
  suppressWarnings(doclimpio1 <- tm_map(doclimpio1, removePunctuation))
  suppressWarnings(doclimpio1 <- tm_map(doclimpio1, removeNumbers))
  suppressWarnings(doclimpio1 <- tm_map(doclimpio1, removeWords,
c(stopwords("Spanish")[1:117],words2)))
  suppressWarnings(doclimpio1 <- tm_map(doclimpio1, stripWhitespace))
  X <- as.data.frame(unlist(doclimpio1))[1,1]
  X <- gsub(" ","",X)
  X <- as.data.frame(table(unlist(strsplit(X,""))))
  ################################################
  Documento2 <- Corpus(VectorSource(D$Old_Adress[i]))
  suppressWarnings(doclimpio2 <- tm_map(Documento2, removeWords,
words1))
  suppressWarnings(doclimpio2 <- tm_map(doclimpio2,
content_transformer(tolower)))
  suppressWarnings(doclimpio2 <- tm_map(doclimpio2, removePunctuation))
  suppressWarnings(doclimpio2 <- tm_map(doclimpio2, removeNumbers))
  suppressWarnings(doclimpio2 <- tm_map(doclimpio2, removeWords,
c(stopwords("Spanish")[1:117],words2)))
  suppressWarnings(doclimpio2 <- tm_map(doclimpio2, stripWhitespace))
  Y <- as.data.frame(unlist(doclimpio2))[1,1] # ifelse(Y=="",Y)
  Y <- gsub(" ","",Y)
  Y <- as.data.frame(table(unlist(strsplit(Y,""))))
  ################################################
  if((nrow(X)==0 & nrow(Y)!=0) | (nrow(X)!=0 & nrow(Y)==0)) {
    D$Coincidencia[i] <- 0
  } else {
    Z <- full_join(X,Y,by=c("Var1"="Var1"))
    Z[is.na(Z)] <- 0
    Z$Min <- apply(Z[,2:3],1,min)
    Z$Max <- apply(Z[,2:3],1,max)
    Z$Coincidence <- ifelse(Z$Min==0 & Z$Max==0,1,Z$Min/Z$Max)
    D$Coincidence[i] <- sum(Z$Coincidence)/nrow(Z)*100
  }
}
```

Figure 3.1: Text analytics original code

Once the code that allows to extract the % of coincidence has been applied (find attached code in Appendix A), we need to keep an eye on wrong classification. This is a critical point, because the policies that will be under analysis together with the rest of variables will be those which are actually fraudulent and non-fraudulent. Hence, the minimum and maximum

% of coincidence that is finally set will require a correct classification, since wrong classification can produce distorted analysis in the following chapters.

3.4.2.1 Fraudulent policies

As it had been mentioned in previous chapters, it is critical to keep false positives under control. Arbitrarily, it we start with a minimum of 70% of coincidence to see the number of false positives obtained in 2020:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 164  50
##          1   3  80
##
##                Accuracy : 0.8215
##                  95% CI : (0.7732, 0.8634)
##     No Information Rate : 0.5623
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6223
##
##  Mcnemar's Test P-Value : 2.64e-10
##
##             Sensitivity : 0.9820
##             Specificity : 0.6154
##          Pos Pred Value : 0.7664
##          Neg Pred Value : 0.9639
##              Prevalence : 0.5623
##          Detection Rate : 0.5522
##    Detection Prevalence : 0.7205
##       Balanced Accuracy : 0.7987
##
##        'Positive' Class : 0
##
```

Figure 3.2: Confusion matrix output

We see that, with a % of coincidence of 70%, there are 3 false positives identified. We can also highlight the accuracy of the model, which reveals almost an 82%. It means that the model has predicted the same values that the actual ones the 82% of the times.

Since the idea is to obtain also an accurate model, we need to set a % of coincidence so that false positives are 0, but at the same time trying to maximize the accuracy of the model, which is translated in reaching the minimum % of coincidence to make false positives 0.

We can set a series of % of coincidence and extract its respective false positives, where we have observed that the minimum % to be settled is 75% (find attached code in Appendix A). If we check now the confusion matrix:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 167  61
##          1   0  69
##
##                Accuracy : 0.7946
##                  95% CI : (0.7441, 0.8391)
##     No Information Rate : 0.5623
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5599
##
##  Mcnemar's Test P-Value : 1.564e-14
##
##             Sensitivity : 1.0000
##             Specificity : 0.5308
##          Pos Pred Value : 0.7325
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5623
##          Detection Rate : 0.5623
##    Detection Prevalence : 0.7677
##       Balanced Accuracy : 0.7654
##
##        'Positive' Class : 0
##
```

Figure 3.3: Confusion matrix output

We can check that false positives are finally 0, although the accuracy of the model has decreased 3 percentage points approximately.

If this procedure is replied to 2021, we obtain the following:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 178  43
##          1   0  84
##
##                Accuracy : 0.859
##                  95% CI : (0.8148, 0.8961)
##     No Information Rate : 0.5836
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6951
##
##  Mcnemar's Test P-Value : 1.504e-10
##
##             Sensitivity : 1.0000
##             Specificity : 0.6614
##          Pos Pred Value : 0.8054
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5836
##          Detection Rate : 0.5836
##    Detection Prevalence : 0.7246
##       Balanced Accuracy : 0.8307
##
##        'Positive' Class : 0
##
```

Figure 3.4: Confusion matrix output

We have around an 86% of accuracy and using a minimum of 70% we already have 0 false positives. Applying the same procedure to obtain the minimum % of coincidence so that false positives are 0, we see that the minimum % of coincidence from which false positives are 0 is 66.7% (find attached code in Appendix A).

We have already established a minimum % of coincidence that must be applied for both years, so that fraud will be recognized from those specific levels. Nevertheless, as it has been mentioned before, it is important to also recognize a level from which fraud is not identified. This step will allow us to divide the final data set between fraudulent and non-fraudulent policies.

3.4.2.2 Non-fraudulent policies

For the identification of non-fraudulent policies, we will detail the same structure, but on the contrary. In other words, we will check if once a specific level of coincidence is set, we can say that there is no fraud identified. Again, using the same methodology, so that false negatives are 0.

If we start with 2020 and compute the confusion matrix:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  97   0
##          1  70 130
##
##                Accuracy : 0.7643
##                  95% CI : (0.7118, 0.8114)
##     No Information Rate : 0.5623
##     P-Value [Acc > NIR] : 3.226e-13
##
##                   Kappa : 0.5481
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5808
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6500
##              Prevalence : 0.5623
##          Detection Rate : 0.3266
##    Detection Prevalence : 0.3266
##       Balanced Accuracy : 0.7904
##
##        'Positive' Class : 0
##
```

Figure 3.5: Confusion matrix output

We can see that there are 0 false negatives that have been identified. Nevertheless, applying the same methodology showed above (find attached code in Appendix A), we could increase from 25% to 33.3% the top % that will allow us to say that there is no fraud in the addresses.

If we proceed now with 2021 and compute the confusion matrix:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 144   0
##          1  34 127
##
##                Accuracy : 0.8885
##                  95% CI : (0.8477, 0.9216)
##     No Information Rate : 0.5836
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7791
##
##  Mcnemar's Test P-Value : 1.519e-08
##
##             Sensitivity : 0.8090
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7888
##              Prevalence : 0.5836
##          Detection Rate : 0.4721
##    Detection Prevalence : 0.4721
##       Balanced Accuracy : 0.9045
##
##        'Positive' Class : 0
##
```

Figure 3.6: Confusion matrix output

We can see that there are 0 false negatives that have been identified, but if we again apply the same methodology, we can settle a maximum % of coincidence of 34.7% in 2021 (find attached code in Appendix A).

After having done all the computations, we have finally fixed the levels from which fraud will be detected. These are:

% Of Coincidence	2020	2021
Fraudulent Policy	> 75%	> 66.7%
Non-Fraudulent Policy	< 33.3%	< 34.7%

Table 3.6: % of Coincidence Levels

We can now expand this structure to the whole data sets of addresses, where we are finally able to classify if the policy is fraudulent or not according to the constructed matrix and build the final data set that will be used for the development of the models (find attached code in Appendix A).

If we analyze average premiums for fraudulent and non-fraudulent policies, we can see how a fraudulent policy has a higher average premium than a non-fraudulent policy, which explains the initiative of an intermediary to commit this fraud:

```
##
## Table 3.7: Mean Average Premium
## ====================
##   Prediction   AP
## --------------------
## 1     0      257.283
## 2     1      295.438
## --------------------
```

Up to this point, we have made an analysis of the whole process of fraud identification. We have settled the boundaries from which fraud is identified, which has allowed us to build the final data set that will be used from now on. It has been relevant to see how in fact average premiums are higher when we are talking about a fraudulent policy. In the next chapter, we will put more emphasis on the behavior of the variables that are present in the data set.

CHAPTER IV

Data Set Analysis

4.1. Exploratory Analysis

Before starting with the construction of the model, it is relevant to make an exploratory analysis of the series of variables present in the data set. It will give us some insights of which variables could be more important, if it is necessary to make some calculation to reduce the number of levels in categorical variables or maybe reduce the number of variables to be used.

We need to make a deep dive now on the data set, which contain information associated to the policy. It does have 6,663 rows and 20 columns (find attached code in Appendix B). It has also been observed that the data set contains 20,335 missing values, but we can check if all variables have the same amount of missing data.

```
##
## Table 4.1: Missing data per variable
## ====================================
##           Variables         Frequency
## ------------------------------------
## 1       Type_Business         0.870
## 2           Claims            0.860
## 3            Loss             0.860
## 4   Housing_Location_Area     0.140
## 5             DT              0.120
## 6          Channel            0.090
## 7          Cluster            0.050
## 8        Use_Building         0.040
## 9    Cancellation_Reason      0.010
## 10        Prediction            0
## 11       Intermediary           0
## 12      Type_Building           0
## 13 Years_From_Construction      0
## 14        Exposition            0
## 15          Income              0
## 16         Product              0
## 17 Continent_Capitals_Range     0
## 18  Content_Capitals_Range      0
## 19           TOB                0
## 20           Year               0
## ------------------------------------
```

After having analyzed the proportion of the missing data per variable for both data sets, we can see that there are 3 variables whose proportion of missing data is > 50%. These are: Loss, Claims and Type_Business.

It is important to mention that Claims and Loss missing data is equivalent to say that there is no claims and no loss associated to a specific policy. This means that it will be necessary to convert missing data for these two variables to 0.

The variable Type_Business, due to its high number of missing values, will be under analysis to decide how to be treated. Nevertheless, before starting with the treatment of variables, we must make firstly a deep dive on data.

Find attached below a brief statistical summary of the numerical variables present in the data set:

```
##
## Table 4.2: Statistical measures of numerical variables
## ======================================================
##          Variables          Min      Mean       Max
## ------------------------------------------------------
## 1       Prediction           0       0.455       1
## 2 Years_From_Construction    0      29.117      261
## 3         Claims             0       0.191       5
## 4          Loss              0      117.335   119,232
## 5       Exposition         0.002     0.436     1.002
## 6         Income          -513.140  122.296  2,532.340
## 7          Year            2,020   2,020.555   2,021
## ------------------------------------------------------
```

The behavior of the numeric variables has been interesting, and there are some issues that must be understood.

First of all, the response variable seems to be quite balanced. This fact suggests that the proportion of fraudulent and non-fraudulent policies present in the data is similar.

On the other hand, we have found negative values in Income. This is associated with the fact that when there are negative values present in this variable, it means that there have been in fact recoveries. Because of some misunderstanding the income associated to the policy was overcharged, and it appears a negative value to compensate.

Apart from this case, the exposition values could call the attention, since it has been seen that there are values higher than 1. This is produced because 2020 is a leap year, and exposition

is computed over 365. This means that if a policy has been fully exposed to 2020, the value showed will be the result of computing 366 / 365 = 1.0027, which coincides with the maximum value observed in the statistical summary of 2020.

We can analyze now correlations between numerical variables, since it could have influence when the model is created.

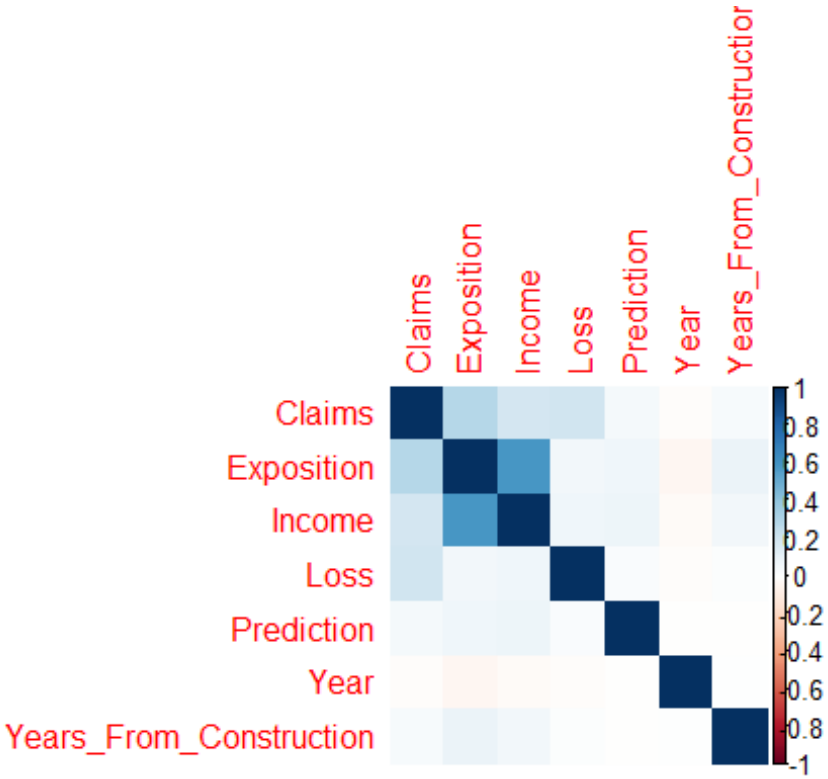

Figure 4.1: Correlation matrix

It seems that Income and Exposition keep a strong correlation between them. There are also slight correlation issues between Loss and Claims. Both types of correlations seem to have sense, since a policy is able to generate more income when its exposition is higher. On the other hand, if a policy has more claims than others, it could seem reasonable that this policy can generate in general more loss.

If we turn now to character variables and perform a brief statistical analysis, we can observe the proportion of the respective levels of variables:

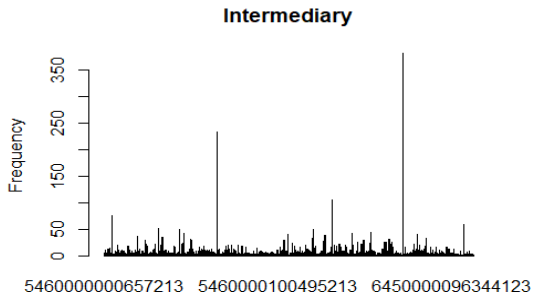

Figure 4.2: Intermediary bar plot

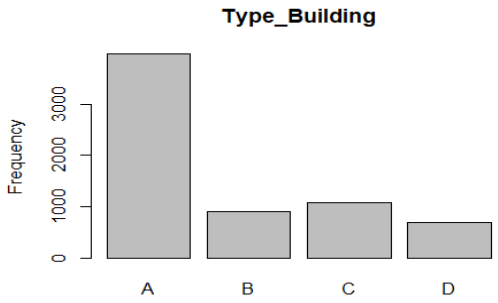

Figure 4.3: Type of building bar plot

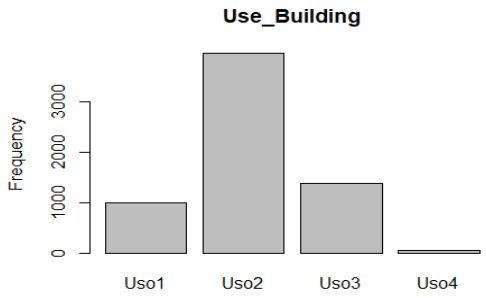

Figure 4.4: Use of building bar plot

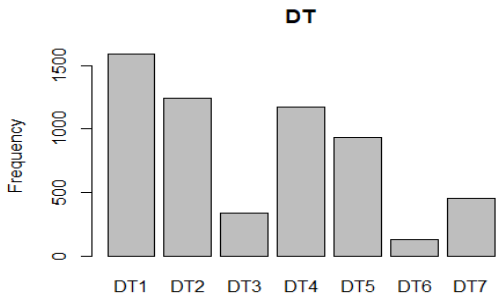

Figure 4.5: DT bar plot

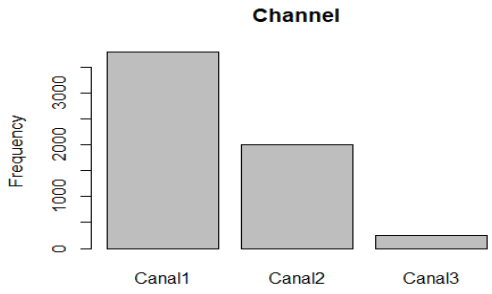

Figure 4.6: Channel bar plot

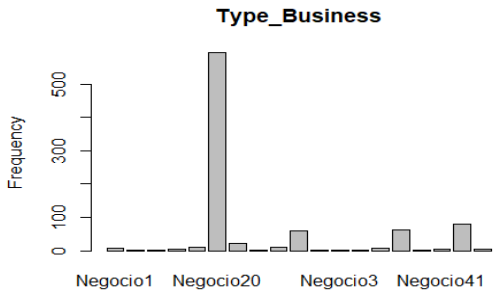

Figure 4.7: Type of business bar plot

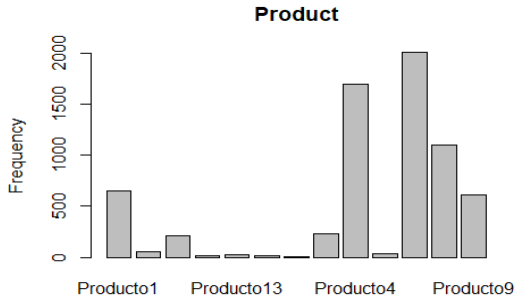

Figure 4.8: Product bar plot

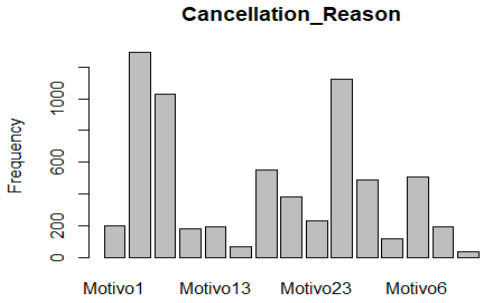

Figure 4.9: Cancellation reason bar plot

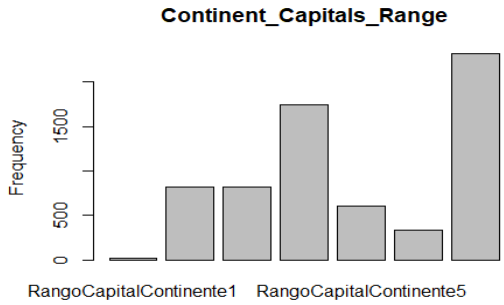

Figure 4.10: Continent capitals range bar plot

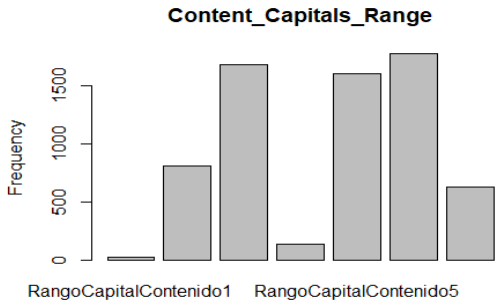

Figure 4.11: Content capitals range bar plot

Figure 4.12: Housing location Area bar plot

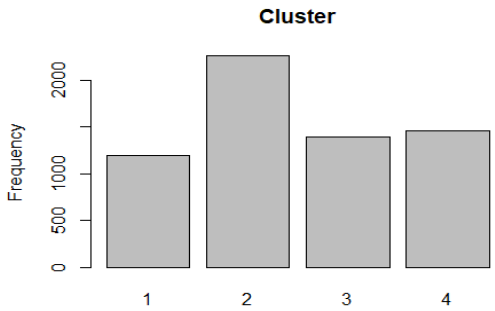

Figure 4.13: Cluster bar plot

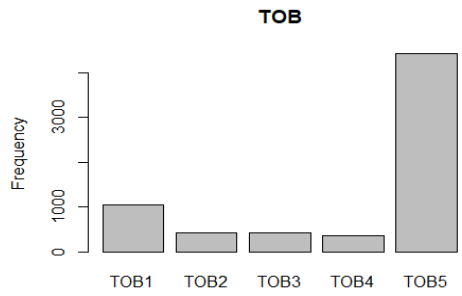

Figure 4.14: TOB bar plot

It can be seen that for both analysis that there are 3 variables which contain a specific dominance level: Intermediary, Type_Business, Housing_Location_Area.

In the next chapter, we will put emphasis on how this fraud can be predicted. This will be done, as it has been previously mentioned, through the construction of 2 types of models, which will be compared afterwards: a generalized linear model and a machine learning model.

CHAPTER V

Model Construction

5.1. Generalized Linear Model (GzLM)

A generalized linear model, according to [3], is made up of a linear predictor:

$$\eta_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$$

and two functions

- a link function that describes how the mean, $E(Y_i) = \mu_i$, depends on the linear predictor

  $g(\mu_i) = \eta_i$

- a variance function that describes how the variance, $var(Y_i)$ depends on the mean

  $var(Y_i) = \phi V(\mu)$

  where the dispersion parameter ϕ is a constant.

There are several families that belong to generalized lineal models, but for the purposes of the project we will focus on binomial family, which leans on whether each observation of a data set holds or does not hold a target characteristic (fraud, in this case).

If we suppose the variable of interest (Y_i) is fraud, so that:

$Y_i \sim Binomial(n_i, p_i)$

and we wish to model the proportions $\frac{Y_i}{n_i}$. Then

$$E\left(\frac{Y_i}{n_i}\right) = p_i$$

$$var\left(\frac{Y_i}{n_i}\right) = \frac{1}{n_i} p_i(1 - p_i)$$

Hence, the variance function is:

$$V(\mu_i) = \mu_i(1 - \mu_i)$$

And the link function must map from (0,1) –> $(-\infty, +\infty)$, where a common choice is:

$$g(\mu_i) = logit(\mu_i) = log\left(\frac{\mu_i}{1-\mu_i}\right)$$

and that will be used to transform the linear combination of the explanatory variables into the outcome that wants to be predicted. There are other types of link functions, but R chooses logit function by default since it is the most common and the most widely used.

Before starting with the construction of the model, it will be necessary to decide which variables will be included and how, since there are some of them which have a lot of levels with very low observations. It will be under analysis those variables that have already been identified in previous chapters, which are: intermediary, type of business, housing location area.

The treatment for these variables will be (find attached code in Appendix C):

- Intermediary: there are 2 clear dominant levels (although it is actually 1 level because the intermediary code has been randomized both for 2021 and 2021). The variable will be re categorized in order to obtain 2 levels. Nevertheless, for these two specific levels, there is a lack of information in the rest of variables, so *Intermediary* will be finally removed from the model.

- Type of Business: there is too much missing data, so that this variable will not be included in the model.

- Housing Location Area: there is 1 dominant level. The variable will be re categorized in order to obtain 2 levels.

Once we have already made all the transformations and decided which variables will be included in the model, we can finally start with its construction. Find below a summary of the model:

```
##
## Call:
## glm(formula = Prediction ~ Type_Building + Use_Building +
Years_From_Construction +
##     DT + Channel + Claims + Loss + Exposition + Income + Product +
##     Cancellation_Reason + Continent_Capitals_Range +
Content_Capitals_Range +
##     Housing_Location_Area + Cluster + TOB + Year, family =
"binomial",
##     data = BD2)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.9413  -1.0756  -0.4511   1.0578   2.5959
##
## Coefficients:
##                                                 Estimate Std.
Error z value
## (Intercept)                                    3.691e+02
2.140e+02   1.724
## Type_BuildingB                                -1.125e-02  9.352e-
02  -0.120
## Type_BuildingC                                -3.058e-01  1.038e-
01  -2.947
## Type_BuildingD                                -1.883e-01  1.280e-
01  -1.472
## Use_BuildingUso2                               2.924e-01  9.760e-
02   2.996
## Use_BuildingUso3                              -8.481e-02  1.127e-
01  -0.752
## Use_BuildingUso4                               3.950e-01  3.783e-
01   1.044
## Years_From_Construction                       -4.527e-03  1.689e-
03  -2.680
## DTDT2                                         -4.945e-02  9.726e-
02  -0.508
## DTDT3                                         -2.496e-01  1.431e-
01  -1.745
## DTDT4                                          7.061e-02  9.465e-
02   0.746
## DTDT5                                         -2.185e-02  1.021e-
01  -0.214
## DTDT6                                         -1.593e-01  4.348e-
01  -0.366
## DTDT7                                          5.951e-01  1.292e-
01   4.608
## ChannelCanal2                                 -1.604e-01  7.079e-
02  -2.265
## ChannelCanal3                                  9.046e-01  5.457e-
01   1.658
## Claims                                         1.169e-01  6.394e-
02   1.828
## Loss                                           6.034e-05  5.333e-
05   1.131
## Exposition                                     2.138e-01  1.497e-
01   1.428
## Income                                        -9.387e-04  3.478e-
04  -2.699
## ProductProducto2                              -5.455e-02  9.901e-
01  -0.055
## ProductProducto3                              -1.912e-01  1.915e-
01  -0.999
```

```
## ProductProducto4                                    -9.155e-02  1.078e-
01  -0.849
## ProductProducto5                                    -1.178e+01
1.970e+02  -0.060
## ProductProducto6                                    -1.788e-01  1.110e-
01  -1.611
## ProductProducto8                                    -7.542e-01  2.224e-
01  -3.391
## Cancellation_ReasonMotivo10                          1.504e-01  1.916e-
01   0.785
## Cancellation_ReasonMotivo11                         -4.426e-01  1.963e-
01  -2.255
## Cancellation_ReasonMotivo12                          1.650e-01  2.755e-
01   0.599
## Cancellation_ReasonMotivo13                         -3.807e-01  2.484e-
01  -1.533
## Cancellation_ReasonMotivo14                         -4.398e-02  3.282e-
01  -0.134
## Cancellation_ReasonMotivo2                          -7.574e-02  2.073e-
01  -0.365
## Cancellation_ReasonMotivo3                          -3.341e-01  1.871e-
01  -1.786
## Cancellation_ReasonMotivo4                          -5.010e-02  2.086e-
01  -0.240
## Cancellation_ReasonMotivo5                           1.643e-01  2.851e-
01   0.576
## Cancellation_ReasonMotivo6                          -8.723e-01  2.042e-
01  -4.272
## Cancellation_ReasonMotivo7                          -1.928e+00  2.856e-
01  -6.750
## Cancellation_ReasonMotivo9                          -3.473e-01  4.634e-
01  -0.749
## Continent_Capitals_RangeRangoCapitalContinente2 -1.532e+00  6.270e-
01  -2.444
## Continent_Capitals_RangeRangoCapitalContinente3 -1.396e+00  6.100e-
01  -2.288
## Continent_Capitals_RangeRangoCapitalContinente4 -9.735e-01  5.978e-
01  -1.628
## Continent_Capitals_RangeRangoCapitalContinente5 -9.092e-01  5.879e-
01  -1.546
## Continent_Capitals_RangeRangoCapitalContinente6 -1.216e+00  6.182e-
01  -1.966
## Continent_Capitals_RangeRangoCapitalContinente7 -1.076e+00  6.046e-
01  -1.780
## Content_Capitals_RangeRangoCapitalContenido2    -8.034e-01  5.748e-
01  -1.398
## Content_Capitals_RangeRangoCapitalContenido3    -5.739e-01  5.535e-
01  -1.037
## Content_Capitals_RangeRangoCapitalContenido4    -2.975e-01  5.600e-
01  -0.531
```

```
## Content_Capitals_RangeRangoCapitalContenido5    -2.129e-01  5.496e-
01  -0.387
## Content_Capitals_RangeRangoCapitalContenido6     1.067e-01  5.462e-
01   0.195
## Content_Capitals_RangeRangoCapitalContenido7     6.405e-02  5.441e-
01   0.118
## Housing_Location_AreaOther                       -2.633e-01  2.087e-
01  -1.261
## Cluster2                                         -7.539e-02  8.685e-
02  -0.868
## Cluster3                                         -6.641e-02  1.010e-
01  -0.657
## Cluster4                                         -7.512e-02  1.013e-
01  -0.742
## TOBTOB2                                          -4.597e-01  1.893e-
01  -2.428
## TOBTOB3                                          -1.831e-01  2.043e-
01  -0.896
## TOBTOB4                                           1.224e-01  2.639e-
01   0.464
## TOBTOB5                                           8.942e-02  2.400e-
01   0.373
## Year                                             -1.818e-01  1.060e-
01  -1.716
##                                                  Pr(>|z|)
## (Intercept)                                      0.084635 .
## Type_BuildingB                                   0.904262
## Type_BuildingC                                   0.003211 **
## Type_BuildingD                                   0.141135
## Use_BuildingUso2                                 0.002738 **
## Use_BuildingUso3                                 0.451773
## Use_BuildingUso4                                 0.296370
## Years_From_Construction                          0.007366 **
## DTDT2                                            0.611132
## DTDT3                                            0.081072 .
## DTDT4                                            0.455666
## DTDT5                                            0.830635
## DTDT6                                            0.714099
## DTDT7                                            4.07e-06 ***
## ChannelCanal2                                    0.023504 *
## ChannelCanal3                                    0.097346 .
## Claims                                           0.067491 .
## Loss                                             0.257876
## Exposition                                       0.153157
## Income                                           0.006957 **
## ProductProducto2                                 0.956061
## ProductProducto3                                 0.317952
## ProductProducto4                                 0.395763
## ProductProducto5                                 0.952317
## ProductProducto6                                 0.107171
```

```
## ProductProducto8                                    0.000697 ***
## Cancellation_ReasonMotivo10                         0.432560
## Cancellation_ReasonMotivo11                         0.024138 *
## Cancellation_ReasonMotivo12                         0.549304
## Cancellation_ReasonMotivo13                         0.125390
## Cancellation_ReasonMotivo14                         0.893384
## Cancellation_ReasonMotivo2                          0.714849
## Cancellation_ReasonMotivo3                          0.074055 .
## Cancellation_ReasonMotivo4                          0.810203
## Cancellation_ReasonMotivo5                          0.564442
## Cancellation_ReasonMotivo6                          1.94e-05 ***
## Cancellation_ReasonMotivo7                          1.48e-11 ***
## Cancellation_ReasonMotivo9                          0.453566
## Continent_Capitals_RangeRangoCapitalContinente2 0.014542 *
## Continent_Capitals_RangeRangoCapitalContinente3 0.022118 *
## Continent_Capitals_RangeRangoCapitalContinente4 0.103425
## Continent_Capitals_RangeRangoCapitalContinente5 0.122001
## Continent_Capitals_RangeRangoCapitalContinente6 0.049254 *
## Continent_Capitals_RangeRangoCapitalContinente7 0.075039 .
## Content_Capitals_RangeRangoCapitalContenido2    0.162253
## Content_Capitals_RangeRangoCapitalContenido3    0.299769
## Content_Capitals_RangeRangoCapitalContenido4    0.595211
## Content_Capitals_RangeRangoCapitalContenido5    0.698486
## Content_Capitals_RangeRangoCapitalContenido6    0.845140
## Content_Capitals_RangeRangoCapitalContenido7    0.906283
## Housing_Location_AreaOther                          0.207242
## Cluster2                                            0.385350
## Cluster3                                            0.510960
## Cluster4                                            0.458252
## TOBTOB2                                             0.015179 *
## TOBTOB3                                             0.370041
## TOBTOB4                                             0.642668
## TOBTOB5                                             0.709414
## Year                                                0.086246 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7171.6  on 5176  degrees of freedom
## Residual deviance: 6498.4  on 5118  degrees of freedom
## AIC: 6616.4
##
## Number of Fisher Scoring iterations: 10
```

Figure 5.1: Generalized linear model summary

After having computed the summary of model, we need to bear in mind that we can use several methods to test how good the model fits the data. Some examples are these:

- Deviance statistic

As [4] mentioned, "Let M_A be a model with q parameters nested in model M_B with $p > q$ parameters and $\widetilde{\pi_A}$ and $\widetilde{\pi_B}$ the fitted probabilities for both models. Then, the parameters for M_B are those common (β_1) to M_A and those specific (β_2), i.e.",

$\beta_B^T = (\beta_1^T, \beta_2^T)$ = and $\beta_A^T = \beta_1^T$

Following [4], "the test for GzLM equivalent to classical F Test in linear regression compares the scaled deviances between 2 hierarchical (nested) models through their difference":

$\Delta D_{AB} = D(y, \widetilde{\pi_A})$ - $D(y, \widetilde{\pi_B}) = 2l(\widetilde{\pi_B}, y)$ - $2l(\widetilde{\pi_A}, y) \sim \chi^2_{p-q}$

- AIC

AIC (Akaike Information Criteria), proposed by [5], is defined as a "trade-off between a goodness of fit provided by a model (M) and the number of parameters p (as an indicator for model complexity)":

$AIC_M = 2p$ - $2l(\widetilde{\pi_M}, y)$

- BIC

BIC (Bayesian Information Criteria) is proposed by [6] and "it takes into account the sample size":

$BIC_M = \log(n)\, p$ - $2l(\widetilde{\pi_M}, y)$

- Calibration plot

As [7] states, a calibration plot is a "graphical tool to assess the model goodness of fit. It represents the observed probabilities (with their 95% CI) as function of the predicted probabilities. If most of the CIs crosses the identity line, the model is validated".

- Generalized R^2

Deviance for a GzLM, as [7] also mentions, "plays a similar role to the Residual Sum of Squares in classical regression. Hence, it is possible to define a Generalized R^2 or pseudo-R^2", so that:

$0 \leq R^2 \leq 1$, where $R^2 = 1 - \frac{D(y,\pi_A)}{D(y,\pi_0}$

"It is a measure that details the % of total variance of the data that can be explained by a model". The main problem of comparing Generalized R^2 is that when a model has more variables, it usually contains a higher value than a model with less variables.

In this section it will be developed a model according to the *lex parsimoniae* in Latin. As [8] states, "parsimonious models have optimal parsimony, or just the right amount of predictors needed to explain well the model".

In order to choose a final model according to the trade-off of minimum AIC / BIC and lowest predictors as possible, it will be used a Stepwise Algorithm. It will be specified in the parameters of the model that the inclusion or exclusion of variables will depend on BIC values, since it is stricter than AIC and returns models with less predictors, which matches with the parsimony law mentioned above.

Once this algorithm is applied, we can confirm through ANOVA that the remaining variables are statistically significant according to a level of significance of 5%.

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Prediction
##                          LR Chisq Df Pr(>Chisq)
## Use_Building              35.744  3  8.482e-08 ***
## Years_From_Construction    9.129  1   0.002516 **
## Channel                   29.750  2  3.467e-07 ***
## Cancellation_Reason      171.796 12  < 2.2e-16 ***
## Content_Capitals_Range    72.564  6  1.217e-13 ***
## TOB                      103.999  4  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 5.2: Simplified generalized linear model ANOVA

In addition, we can briefly validate the model through residuals and analysis of outliers.

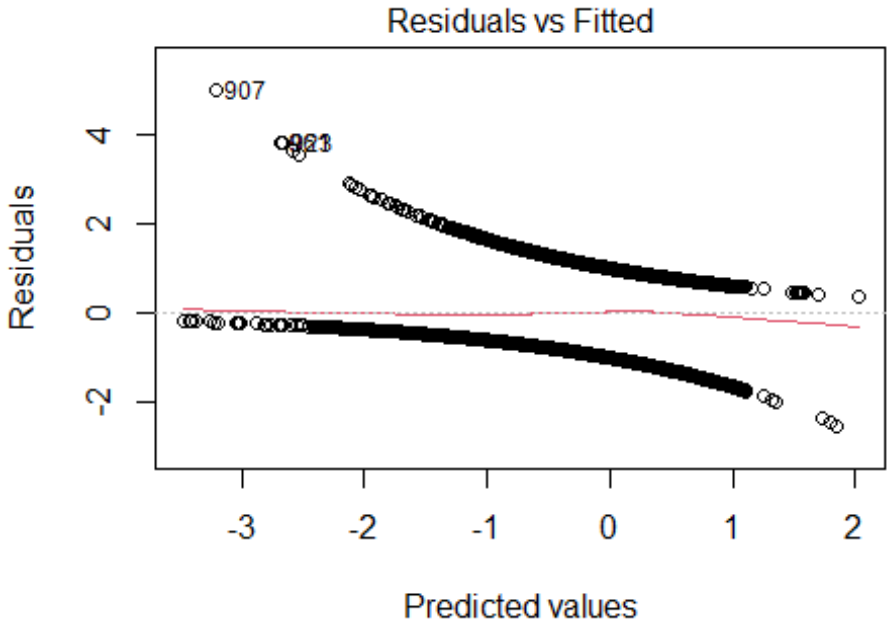

Figure 5.3: Residuals vs fitted values plot

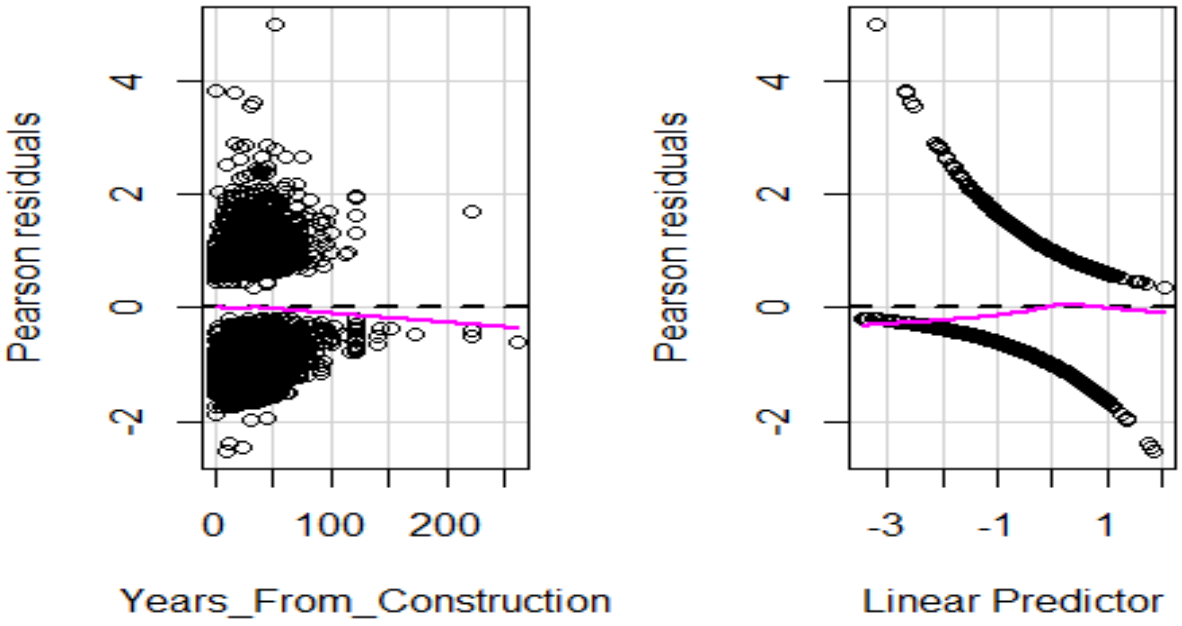

Figure 5.4: Pearson residuals plot

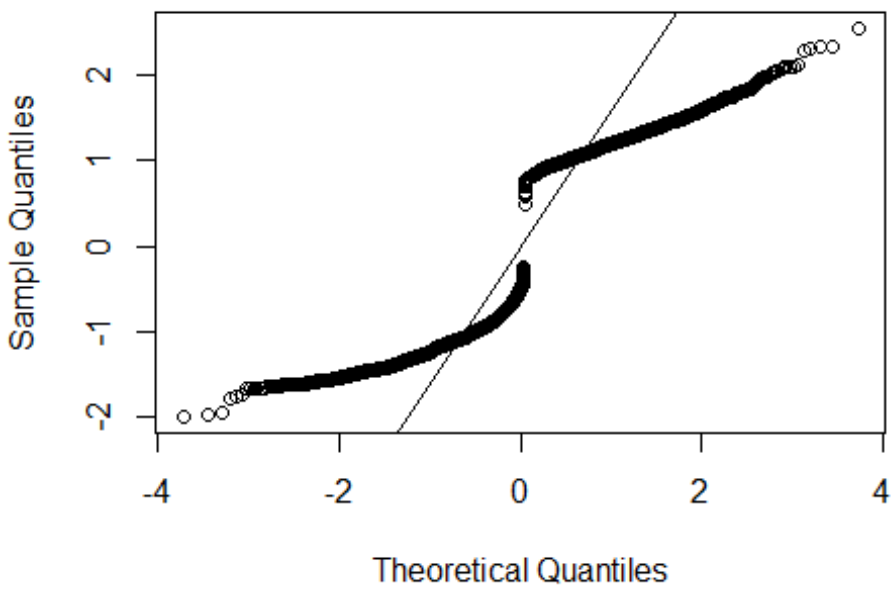

Figure 5.5: Normal Q-Q Plot

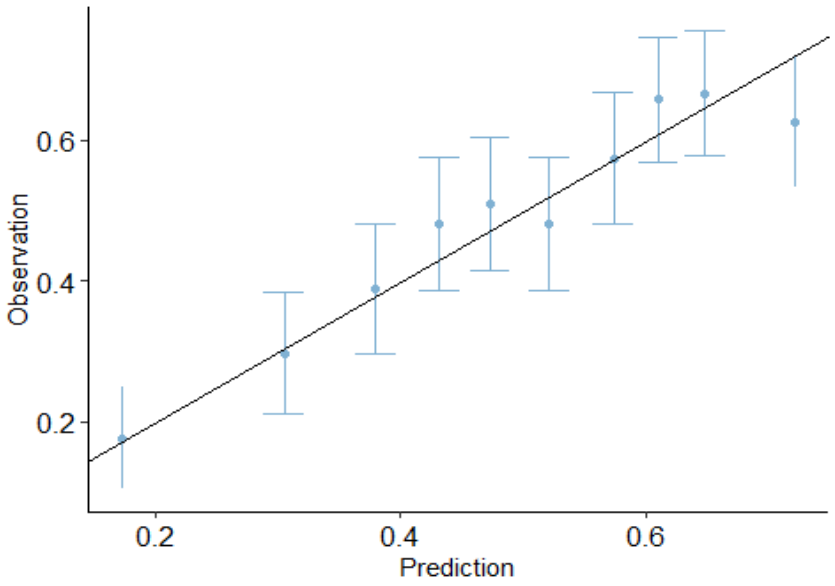

Figure 5.6: Calibration plot for testing data

We have not seen any strange pattern in residual plots and, taking into account the calibration plot, since most of the confidence intervals crosses the identity line, we can finally say that the model is validated.

We can finally compare the last model with the first constructed model using the BIC metric mentioned above, which suggests that Model 2 is preferable to Model 1 (find attached code in Appendix C).

Model	**BIC**
Model 1	7002.918
Model 2	6848.743

Table 5.1: Models Comparison (BIC)

5.2. GzLM Results

We have the final validated model that will be used to make predictions. In order to test its accuracy, we will make use again of a confusion matrix, where we have seen in previous chapters that shows predicted response versus observed response, and AUC / ROC curves, which according to [9] is:

A performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC (Area Under the Curve) represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

By analogy, the Higher the AUC, the better the model is at distinguishing between fraudulent policies and non-fraudulent policies.

For each threshold, ROC curve represents the true positive rate (Sensitivity) vs the false negative rate (1-Specificity). Therefore:

- Sensitivity: it represents the proportion of observations predicted as positive when the observed outcome is positive.
- Specificity: it represents the proportion of observations predicted as negative when the observed outcome is negative.

Hence, we can compute and draw the cutoff from which accuracy is maximized and the final confusion matrix, which are:

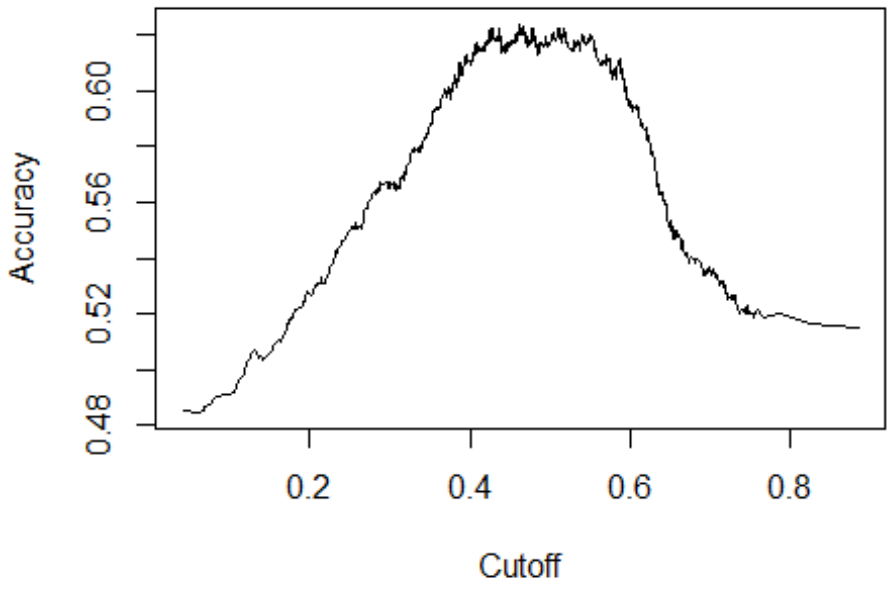

Figure 5.7: Cutoff vs Accuracy

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 298 257
##          1 149 375
##
##                Accuracy : 0.6237
##                  95% CI : (0.5941, 0.6527)
##     No Information Rate : 0.5857
##     P-Value [Acc > NIR] : 0.005978
##
##                   Kappa : 0.2511
##
##  Mcnemar's Test P-Value : 1.094e-07
##
##             Sensitivity : 0.6667
##             Specificity : 0.5934
##          Pos Pred Value : 0.5369
##          Neg Pred Value : 0.7156
##              Prevalence : 0.4143
##          Detection Rate : 0.2762
##    Detection Prevalence : 0.5144
##       Balanced Accuracy : 0.6300
##
##        'Positive' Class : 0
##
```

Figure 5.8: Confusion matrix output

We can observe that predictions and actual values are highly balanced, which remains consistent with the fact that the response variable is balanced as well. We can draw now the ROC curve:

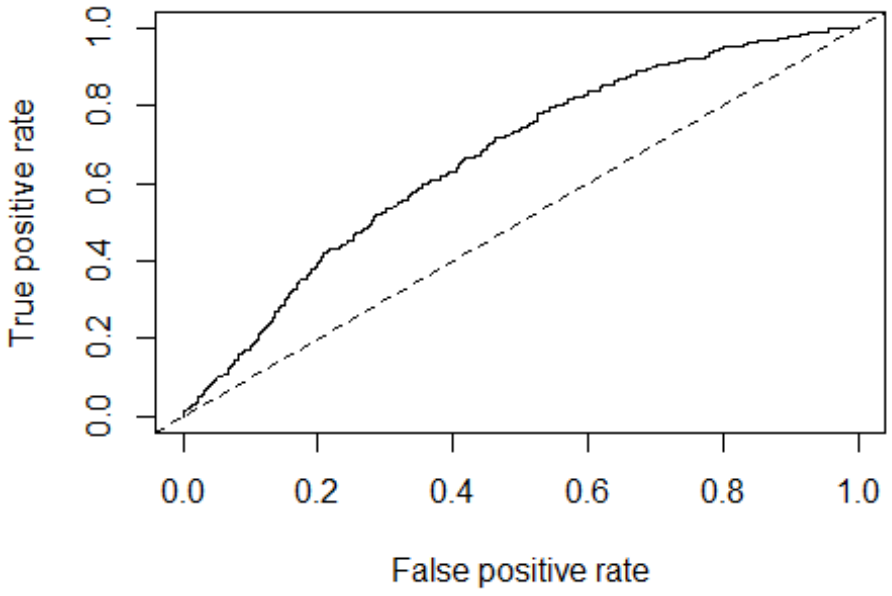

Figure 5.9: ROC Curve

Since the ROC curve does not rise very rapidly towards the upper left-hand corner of the graph, we should be aware that the model may not perform very well. We can also make use of AUC to make a deep dive on it. If we compute AUC (find attached code in Appendix C), we obtain 0.6263, which seems coherent with the conclusions extracted when the ROC curve was plotted.

We have seen that our model has finally obtained an accuracy of 62.37%. In addition, according to the interpretation of the AUC of a ROC curve, we could say the model performs not as well as we would expect to. Nevertheless, taking into account that the response variable is balanced, the model is capable of accurate more than if we randomly assign that a policy is fraudulent, which will bring to an estimated accuracy of ~50%. In addition, it is important to consider that according to [10] "some fields of study have inherently greater amount of inexplicable variation, which means that R^2 values are bound to be lower". If we test the Generalized R^2 of our model (find attached code in Appendix C), we obtain a value of 8.28,

which is actually very low. This also relates with what [10] refers to, that "people are just harder to predict than things like physical processes".

[10] affirms that "even if you have a low R-squared value but the independent variables are statistically significant, you can still draw important conclusions about the relationships between the variables". [10] also says that "Statistically significant coefficients continue to represent the mean change in the dependent variable given a one-unit shift in the independent variable. Clearly, being able to draw conclusions like this is vital".

We can make use of the following command to extract those conclusions of the variables for the obtained model:

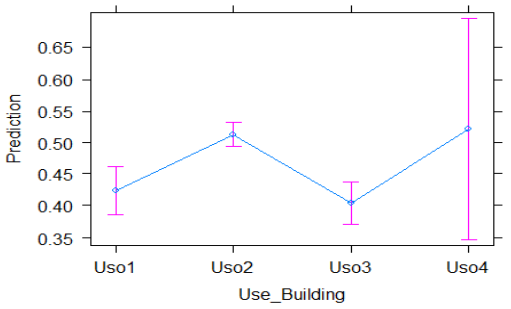

Figure 5.10: Use of building effect plot

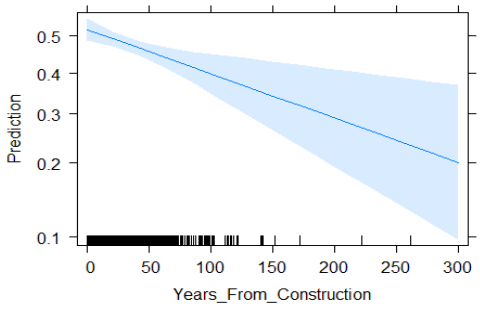

Figure 5.11: Years from constr. effect plot

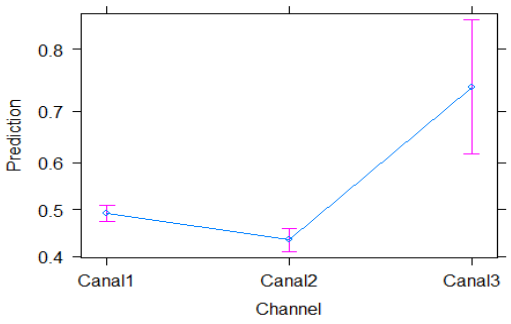

Figure 5.12: Channel effect plot

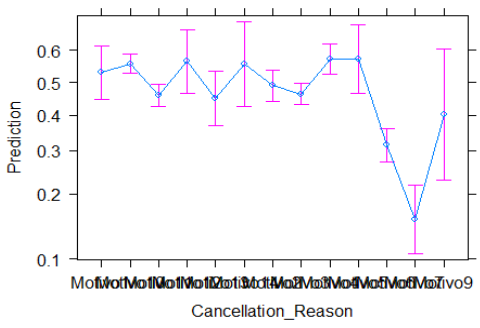

Figure 5.13: Cancellation reason effect plot

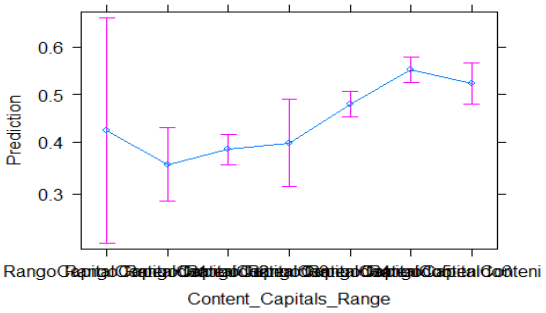

Figure 5.14: Content capitals range effect plot

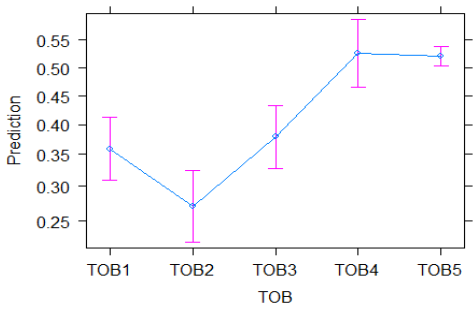

Figure 5.15: TOB effect plot

- Use_Building: the plot shows that *Uso2* makes more prone to commit fraud than the rest of the uses. *Uso4* is plotted with a high variability, which can be explained with the fact that there are low observations and whose prediction is distributed for all its observations.
- Years_From_Construction: we can see that a downtrend is drawn in the plot. It means that in general, if the living place is older, it less likely to commit fraud.
- Channel: if the policy is distributed through channel 3, there is more probability of committing fraud than if it is distributed through channel 1 or 2.
- Cancellation_Reason: most of the cancellation reasons seem to have the same behavior. Nevertheless, cancellation reason 7 is associated with a very low probability of committing fraud.
- Content_Capitals_Range: the first issue that calls the attention is the high variability that *RangoCapitalContenido1* shows. It can also be associated to the fact that it contains a very low proportion of data. On the other hand, graph suggests that the higher the content capitals range insured, the higher the probability of committing fraud.
- TOB: Time on book, as it has been explained in previous chapters, is associated with the age of the policy. In this case, the plot reveals that older policies are more prone to be fraudulent than younger policies.

These results mark the end of this first part of modelling construction, where we have had the possibility of building an generalized linear model that gives a 62.37% of accuracy and

that has allowed us to extract some relevant conclusions of which and how variables make more likely the possibility of having committed fraud.

5.3. Machine Learning Model

What if instead of obtaining important explanations of what makes fraud more prone, we go straight to the point and make predictions of this fraud? We have seen that the generalized linear model has obtained an accuracy of 62.37%, and taking into account the repercussions that categorizing a fraudulent policy could have, we should try to build a model whose accuracy is the highest possible. This is the reason why in this section a machine learning model will be built, and it will be done through XGBoost.

As it is stated in [11], "we go through cycles that repeatedly builds new models and combines them into an **ensemble** model". [11] also says the following:

We start the cycle by taking an existing model and calculating the errors for each observation in the data set. We then build a new model to predict these errors. We add predictions from this error-predicting model to the "ensemble of models."

To make a prediction, we add the predictions from all previous models. We can use these predictions to calculate new errors, build the next model, and add it to the ensemble.

There's one piece outside that cycle. We need some base prediction to start the cycle. In practice, the initial predictions can be naive. Even if its predictions are wildly inaccurate, subsequent additions to the ensemble will address those errors.

Before starting with the construction of the model, we need to split the data set into testing and training subsets. After running de model (find attached code in Appendix C), we can observe the following output of the first two rounds:

```
## [1]  train-error:0.337744
## [2]  train-error:0.323791
```

Figure 5.16: Machine learning first two training errors

It can be seen from the output obtained that we have an improvement in the second round of training, so that we can apply more rounds (find attached code in Appendix C).

Although we have observed that errors are becoming lower and lower for each round, we have seen that test error reveals a 0.356 value. Since testing errors are much higher than training errors (find attached code in Appendix C), we could conclude that there is an issue of model over-fitting, which means the model relies too much on randomness/noise in the training set to make classifications. As a result, it can be seen that the model cannot extend to a new data set.

In order to fix this, we will add additional parameters with the intention of increasing the accuracy as much as possible. After having tuned the model (find attached code in Appendix C), we can see that the model has increased its accuracy from 59.31% to 61.17%.

We can also see the importance of each variable to the built model. In this case:

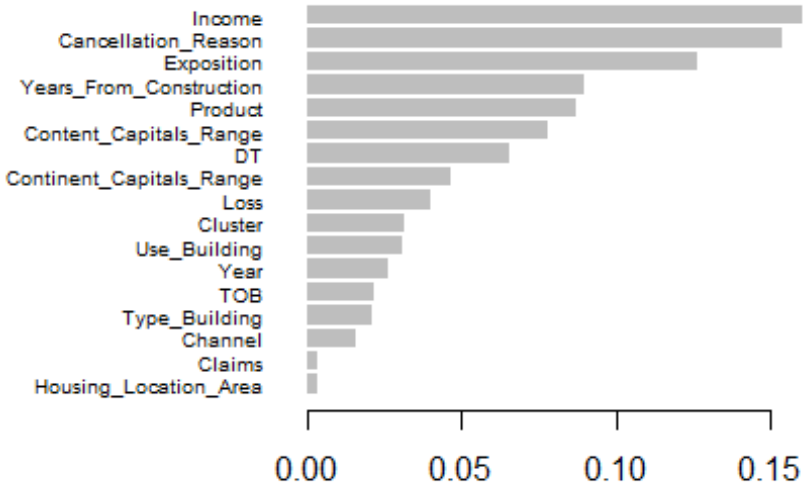

Figure 5.17: Machine learning variables importance

Income and Cancellation_Reason have been the most important variables, which means that these two variables have contributed the most in increasing the accuracy of the model. Nevertheless, unlike the generalized linear model, we cannot explain how these two variables influence the final prediction.

5.4. K-Fold Cross Validation Comparison

After having obtained the accuracy results above, it seems that both models have performed similarly, although the generalized linear model has performed slightly better. If we compare the results obtained, we have:

Model	**Accuracy**
Generalized Linear Model	62.37%
Machine Learning Model	61.17%

Table 5.2: Models accuracy comparison

Nevertheless, in order to have more consistent results and to test whether the models' performance difference is significant or not, it could be interesting to make use of a tool called K-Fold Cross Validation. As we have observed in [12], K-Fold Cross Validation is:

Where a given data set is split into a ***K*** number of sections/folds where each fold is used as a testing set at some point. If we take the scenario of 5-Fold cross validation (K=5), the data set is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set.

Thus, if we apply this methodology to the present study case (find attached the code in Appendix C), we obtain the following results:

```
##         GLM        ML
## 1 0.6255969 0.5978988
## 2 0.6475170 0.6212269
## 3 0.6395233 0.6454816
## 4 0.6282288 0.6236162
## 5 0.6640316 0.6709486
##
## Table 5.3: K-Fold CV Comparison
## ======================================
## Statistic N Mean  St. Dev.  Min   Max
## --------------------------------------
## GLM       5 0.641  0.016   0.626 0.664
## ML        5 0.632  0.028   0.598 0.671
## --------------------------------------
```

The results that have been obtained show that training with new data has been beneficial for both models. In fact, the accuracy of the generalized linear model has increased in average around 2 percentage points, from 62.4% to 64.1%. On the other hand, machine learning model has also increased its accuracy in average around 2 percentage points, from 61.2% to 63.2%, but it has obtained a higher variability in the performance.

After having computed the 5-Fold Cross Validation, we can finally validate if the difference of the accuracy of the generalized linear model and the machine learning model is significant or not. Since both confidence intervals intersect each other, we can say that there is no statistical evidence that allows us to affirm that the generalized linear model is significantly better, so that we can assume that both models obtain the same results.

CHAPTER VI

Conclusions

The objective of this study was to develop a model using different tools which allows to detect some type of pattern in data, so that fraud associated to policies could be predicted. We have made use of text analysis techniques to detect this fraud, and after making some preprocessing and cleaning the data, we have been able to build both a generalized lineal model and a machine learning model to predict the outcome. It has been observed that both models have obtained accuracy levels around 63%, so that we should be aware of that if they are intended to be used to predict.

Machine learning model and generalized lineal model have got similar accuracy levels. Therefore, both of them could be applied to make predictions. Nevertheless, since generalized linear model is more informative in terms of explanatory variables, it could be preferable to be chosen rather than the machine learning model. With its aid, we have been able to extract some relevant conclusions related to what makes fraud more prone to be committed, so that they may be taken into account when new policies are issued.

In general, we have observed that the more expensive the average premium of the policy is, the higher the probability of committing fraud. Most of the conclusions extracted in section 5.1. are related to this fact. For instance, older policies (TOB) with a high content capitals range insured contain average premiums that are more expensive, and both characteristics make fraud more prone to be committed. This is consistent with the idea that intermediaries need to use resources to make premiums cheaper in order to retain customers. We have also seen additional factors which make increase the probability of committing fraud, like living in an older house or making *Uso2* of the living.

Despite this project has allowed to obtain relevant conclusions, there have been also several limitations and extensions that should be taken into account. Firstly, the data. It could be interesting to analyze more data of past years, in such a way that we could extract trends in data and feed models with more information. In general, a higher quantity of data can allow the model to make more accurate predictions. On the other hand, the models themselves. In this project, it has been compared two models and although results in terms of accuracy have been similar, the introduction of new models would have allowed for a more extensive and varied comparison.

To conclude, the present project has been helpful in order to extract some insights of how insurance sector works in Spain and we have had the possibility to make a deep dive in a worrying issue, which is the detection of fraud. Thanks to the different data analysis techniques, we have been able to delve a little bit deeper into how this fraud occurs and in what way.

CHAPTER VII

Bibliography

[1] AWS, *What is text analysis*. Amazon, 2022.Available: https://aws.amazon.com/es/what-is/text-analysis/

[2] Md Riaz Ahmed Khan, *ROCit: An r package for performance assessment of binary classifier with visualization*. R Foundation for Statistical Computing, 2020.Available: https://cran.r-project.org/web/packages/ROCit/vignettes/my-vignette.html

[3] Heather Turner, *Introduction to generalized linear models*. ESRC National Centre for Research Methods, UK; Department of Statistics - University of Warwick, UK, 2008.Available: https://statmath.wu.ac.at/courses/heather_turner/glmCourse_001.pdf

[4] Yihui Xie, *Generalized linear models*, 2nd ed. London: Chapman; Hall, 2015.Available: https://books.google.es/books?id=h9kFH2_FfBkC&printsec=frontcover&dq=McCullagh+(1989)&hl=ca&sa=X&redir_esc=y#v=onepage&q=McCullagh%20(1989)&f=false

[5] H. Akaike, *A new look at the statistical model identification*. IEEE Transactions on Automatic Control, 1974.

[6] G. Schwarz, *Estimating the dimension of a model*. Annals of Statistics, 1978.

[7] D. Hosmer D and S. Lemeshow, *Goodness-of-fit tests for the multiple logistic regression model*. Commun Stat Part A Theor Meth, 1980.

[8] Stephanie Glen, *Parsimonious model: Definition, ways to compare models*. StatisticsHowTo.com: Elementary Statistics for the rest of us!Available: https://www.statisticshowto.com/parsimonious-model/

[9] Sarang Narkhede, *Understanding AUC - ROC curve*. 2018.Available: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[10] Jim Frost, *How to interpret r-squared in regression analysis*. Statistics by Jim, 2018.Available: https://statisticsbyjim.com/regression/interpret-r-squared-regression/

[11] Rachel Tatman, *Machine learning with XGBoost (in r)*. Kaggle, 2019.Available: https://www.kaggle.com/code/rtatman/machine-learning-with-xgboost-in-r#Tuning-our-model

[12] Krishni, *K-fold cross validation*. DataDrivenInvestor, 2018.Available: https://medium.datadriveninvestor.com/k-fold-cross-validation-6b8518070833

CHAPTER VIII

Appendix

Appendix A

```
# 3.4. Results

# 3.4.1. Exploratory Analysis

str(A2020); str(A2021)
dim(A2020); dim(A2021)
sum(is.na(A2020)); sum(is.na(A2021))

# 3.4.2 Fraud Identification

Sample2020 <- read_excel("Sample_2020.xlsx")
Sample2021 <- read_excel("Sample_2021.xlsx")

Sample2020$Coincidence <- NA

for (i in 1:nrow(Sample2020)) {
  X <-
as.data.frame(table(unlist(strsplit(Sample2020$Old_Address[i],""))))
  ################################################
  Y <-
as.data.frame(table(unlist(strsplit(Sample2020$New_Address[i],""))))
  ################################################
  if((nrow(X)==0 & nrow(Y)!=0) | (nrow(X)!=0 & nrow(Y)==0)) {
    D$Coincidence[i] <- 0
  } else {
    Z <- full_join(X,Y,by=c("Var1"="Var1"))
    Z[is.na(Z)] <- 0
    Z$Min <- apply(Z[,2:3],1,min)
    Z$Max <- apply(Z[,2:3],1,max)
    Z$Coincidence <- ifelse(Z$Min==0 & Z$Max==0,1,Z$Min/Z$Max)
    Sample2020$Coincidence[i] <- sum(Z$Coincidence)/nrow(Z)*100
  }
}

Sample2021$Coincidence <- NA

for (i in 1:nrow(Sample2021)) {
  X <-
as.data.frame(table(unlist(strsplit(Sample2021$Old_Address[i],""))))
```

```
  ################################################
  Y <-
as.data.frame(table(unlist(strsplit(Sample2021$New_Address[i],""))))
  ################################################
  if((nrow(X)==0 & nrow(Y)!=0) | (nrow(X)!=0 & nrow(Y)==0)) {
    D$Coincidence[i] <- 0
  } else {
    Z <- full_join(X,Y,by=c("Var1"="Var1"))
    Z[is.na(Z)] <- 0
    Z$Min <- apply(Z[,2:3],1,min)
    Z$Max <- apply(Z[,2:3],1,max)
    Z$Coincidence <- ifelse(Z$Min==0 & Z$Max==0,1,Z$Min/Z$Max)
    Sample2021$Coincidence[i] <- sum(Z$Coincidence)/nrow(Z)*100
  }
}

rm(X,Y,Z,i)

# 3.4.2.1. Fraudulent policies

c <- seq(70,90,0.1)
v <- c()
i <- 1
for (i in 1:length(c)) {
  w1$Prediction <- ifelse(w1$Coincidence>c[i],1,0)
  CM <- confusionMatrix(as.factor(w1$Prediction),as.factor(w1$Fraud))
  CM
  v <- c(v,CM$table[2,1])
}

df <- data.frame(Coincidence=c, False_Positives=v)
level2020_1 <- min(subset(df,False_Positives==0)$Coincidence)
level2020_1

# 3.4.2.1. Fraudulent policies

c <- seq(60,80,0.1)
v <- c()
i <- 1
for (i in 1:length(c)) {
  w1$Prediction <- ifelse(w1$Coincidence>c[i],1,0)
  CM <- confusionMatrix(as.factor(w1$Prediction),as.factor(w1$Fraud))
  CM
  v <- c(v,CM$table[2,1])
}

df <- data.frame(Coincidence=c, False_Positives=v)
level2021_1 <- min(subset(df,False_Positives==0)$Coincidence)
level2021_1
```

```
# 3.4.2.2. Non-fraudulent policies

c <- seq(20,40,0.1)
v <- c()
i <- 1
for (i in 1:length(c)) {
  w1$Prediction <- ifelse(w1$Coincidence<c[i],0,1)
  CM <- confusionMatrix(as.factor(w1$Prediction),as.factor(w1$Fraud))
  CM
  v <- c(v,CM$table[1,2])
}

df <- data.frame(Coincidence=c, False_Negatives=v)
level2020_2 <- max(subset(df,False_Negatives==0)$Coincidence)
level2020_2

# 3.4.2.2. Non-fraudulent policies

c <- seq(10,40,0.1)
v <- c()
i <- 1
for (i in 1:length(c)) {
  w1$Prediction <- ifelse(w1$Coincidence<c[i],0,1)
  CM <- confusionMatrix(as.factor(w1$Prediction),as.factor(w1$Fraud))
  CM
  v <- c(v,CM$table[1,2])
}

df <- data.frame(Coincidence=c, False_Negatives=v)
level2021_2 <- max(subset(df,False_Negatives==0)$Coincidence)
level2021_2

# 3.4.2.2. Non-fraudulent policies

A2020$Coincidence <- NA

for (i in 1:nrow(A2020)) {
  X <- as.data.frame(table(unlist(strsplit(A2020$Old_Address[i],""))))
  ################################################
  Y <- as.data.frame(table(unlist(strsplit(A2020$New_Address[i],""))))
  ################################################
  if((nrow(X)==0 & nrow(Y)!=0) | (nrow(X)!=0 & nrow(Y)==0)) {
    D$Coincidence[i] <- 0
  } else {
    Z <- full_join(X,Y,by=c("Var1"="Var1"))
    Z[is.na(Z)] <- 0
    Z$Min <- apply(Z[,2:3],1,min)
    Z$Max <- apply(Z[,2:3],1,max)
```

```
    Z$Coincidence <- ifelse(Z$Min==0 & Z$Max==0,1,Z$Min/Z$Max)
    A2020$Coincidence[i] <- sum(Z$Coincidence)/nrow(Z)*100
  }
}

A2021$Coincidence <- NA

for (i in 1:nrow(A2021)) {
  X <- as.data.frame(table(unlist(strsplit(A2021$Old_Address[i],""))))
  ################################################
  Y <- as.data.frame(table(unlist(strsplit(A2021$New_Address[i],""))))
  ################################################
  if((nrow(X)==0 & nrow(Y)!=0) | (nrow(X)!=0 & nrow(Y)==0)) {
    D$Coincidence[i] <- 0
  } else {
    Z <- full_join(X,Y,by=c("Var1"="Var1"))
    Z[is.na(Z)] <- 0
    Z$Min <- apply(Z[,2:3],1,min)
    Z$Max <- apply(Z[,2:3],1,max)
    Z$Coincidence <- ifelse(Z$Min==0 & Z$Max==0,1,Z$Min/Z$Max)
    A2021$Coincidence[i] <- sum(Z$Coincidence)/nrow(Z)*100
  }
}

rm(X,Y,Z,i,w)

A2020$Prediction <-
ifelse(A2020$Coincidence>level2020_1,1,ifelse(A2020$Coincidence<level20
20_2,0,-1))

A2021$Prediction <-
ifelse(A2021$Coincidence>level2021_1,1,ifelse(A2021$Coincidence<level20
21_2,0,-1))

table(A2020$Prediction)
table(A2021$Prediction)

BD2020 <- A2020[order(A2020$Prediction, decreasing=TRUE),]
BD2020 <- BD2020[!duplicated(BD2020$Policy),]
BD2020 <- subset(BD2020,Prediction!=-1)
BD2020 <- BD2020 %>% left_join(P2020,by=c("Policy"="Policy"))

BD2021 <- A2021[order(A2021$Prediction, decreasing=TRUE),]
BD2021 <- BD2021[!duplicated(BD2021$Policy),]
BD2021 <- subset(BD2021,Prediction!=-1)
BD2021 <- BD2021 %>% left_join(P2021,by=c("Policy"="Policy"))

BD <- rbind(BD2020,BD2021)
```

Appendix B

```
# 4.1. Exploratory Analysis

BD <- BD %>% select (-Policy, -Old_Address, -New_Address, -Coincidence)
BD$TOB <- as.factor(BD$TOB)
levels(BD$TOB) <- c("TOB5","TOB1","TOB2","TOB3","TOB4")
BD$TOB <- as.character(BD$TOB)
str(BD)
dim(BD)
sum(is.na(BD))
```

Appendix C

```
# 5.1. Generalized Linear Model (GzLM)

sum(is.na(BD$Intermediary))/nrow(BD); barplot(table(BD$Intermediary))
sum(is.na(BD$Type_Business))/nrow(BD); barplot(table(BD$Type_Business))
sum(is.na(BD$Housing_Location_Area))/nrow(BD);
barplot(table(BD$Housing_Location_Area))

# 5.1. Generalized Linear Model (GzLM)

as.vector(as.data.frame(sort(table(BD$Intermediary),decreasing=TRUE))[1
,1])
as.vector(as.data.frame(sort(table(BD$Intermediary),decreasing=TRUE))[2
,1])

BD$Intermediary <-
ifelse(BD$Intermediary==as.vector(as.data.frame(sort(table(BD$Intermedi
ary),decreasing=TRUE))[1,1])|BD$Intermediary==as.vector(as.data.frame(s
ort(table(BD$Intermediary),decreasing=TRUE))[2,1]),"Interm1","Interm2")

table(BD$Intermediary)
table(BD$Housing_Location_Area)
BD$Housing_Location_Area <-
ifelse(BD$Housing_Location_Area!="N","Other",BD$Housing_Location_Area)

BD2 <- BD
BD2 <- subset(BD2, Intermediary=="Interm2")
BD2 <- BD2 %>% select(-Intermediary, -Type_Business)

BD2 <- na.omit(BD2) # we omit NA's to avoid issues in the model

# 5.1. Generalized Linear Model (GzLM)
```

```
# BIC

AIC(lm1,k=log(nrow(BD2)))
AIC(lm2,k=log(nrow(BD2))) # lower

# We can test if Income is a statistically significant variable in lm2,
since it was statistically significant in lm1

lm3 <- update(lm2, . ~ . + Income)
anova(lm2, lm3, test="Chisq") # it is not

rm(lm3)

# 5.2. GzLM Results

# AUC

roc <- roc(test$Prediction, Prediction)
auc(roc) # we obtain the accuracy

# 5.2. GzLM Results

rsquared <- (1-lm3$deviance/lm3$null.deviance)*100
rsquared

# 5.3. Machine Learning Model

set.seed(20)

# arbitrarily, 80% of the data will be for training
sample2 <- sample(c(TRUE, FALSE), nrow(BD2), replace=TRUE,
prob=c(0.8,0.2))
train  <- BD2[sample2, ]; train <- data.matrix(train)
test   <- BD2[!sample2, ]; test <- data.matrix(test)

dtrain <- xgb.DMatrix(data = train[,-1], label=train[,"Prediction"])
dtest <- xgb.DMatrix(data = test[,-1], label=test[,"Prediction"])

model <- xgboost(data = dtrain,
                 nround = 2, # number of times to improve the naive
model by adding additional models
                 eval_metric = "error",
                 objective = "binary:logistic")

model <- xgboost(data = dtrain,
                 nround = 100, # number of times to improve the naive
model by adding additional models
```

```
                  eval_metric = "error",
                  objective = "binary:logistic")

# generate predictions for our held-out testing data
pred <- predict(model, dtest)

predi <- prediction(pred,test[,"Prediction"])
plot(acc.perf <- performance(predi,measure="acc"))
ind <- which.max( slot(acc.perf, "y.values")[[1]] )
cutoff <- slot(acc.perf, "x.values")[[1]][ind]

err <- mean(as.numeric(pred > cutoff) != test[,"Prediction"])

# accuracy
1-err

# train an xgboost model
model_tuned <- xgboost(data = dtrain, # the data
                  max.depth = 10, # the maximum depth of each decision
tree
                  nround = 100, # max number of boosting iterations
                  eval_metric = "error",
                  early_stopping_rounds = 3, # if we dont see an
improvement in this many rounds, stop
                  gamma = 5, # add a regularization term
                  objective = "binary:logistic") # the objective
function

# generate predictions for our held-out testing data
pred <- predict(model_tuned, dtest)

predi <- prediction(pred,test[,"Prediction"])
plot(acc.perf <- performance(predi,measure="acc"))
ind <- which.max( slot(acc.perf, "y.values")[[1]] )
cutoff <- slot(acc.perf, "x.values")[[1]][ind]

# get & print the classification error
err <- mean(as.numeric(pred > cutoff) != test[,"Prediction"])

# accuracy
1-err



# 5.4. K-Fold Cross Validation Comparison

set.seed(1234)
k <- 5
index <- sample(1:k, nrow(BD2), replace=TRUE, prob=rep(1,k))
table(index)
```

```
GLM <- c()
MLM <- c()

for (i in 1:k) {
  trainGLM <- subset(BD2, index!=i)
  testGLM <- subset(BD2, index==i)

  trainML <- data.matrix(trainGLM)
  testML <- data.matrix(testGLM)
  dtrain <- xgb.DMatrix(data = trainML[,-1],
label=trainML[,"Prediction"])
  dtest <- xgb.DMatrix(data = testML[,-1], label=testML[,"Prediction"])

  # GENERALIZED LINEAR MODEL

  lm3 <- update(lm2, . ~ ., data = trainGLM)
  predic <- predict(lm3,newdata=testGLM, type="response")

  predi <- prediction(predic,testGLM$Prediction)
  plot(acc.perf<-performance(predi,measure="acc"))
  ind<-which.max( slot(acc.perf, "y.values")[[1]] )
  cutoff = slot(acc.perf, "x.values")[[1]][ind]

  Prediction <- ifelse(predic>=cutoff,1,0)
  CM <- confusionMatrix(as.factor(testGLM$Prediction),
as.factor(Prediction))
  CM

  GLM <- c(GLM, as.numeric(CM$overall[1]))

  # MACHINE LEARNING MODEL

  model_tuned <- xgboost(data = dtrain,
                 max.depth = 10,
                 nround = 100,
                 eval_metric = "error",
                 early_stopping_rounds = 3,
                 gamma = 5,
                 objective = "binary:logistic")

  pred <- predict(model_tuned, dtest)

  predi <- prediction(pred,testML[,"Prediction"])
  plot(acc.perf<-performance(predi,measure="acc"))
  ind<-which.max( slot(acc.perf, "y.values")[[1]] )
  cutoff = slot(acc.perf, "x.values")[[1]][ind]

  err <- mean(as.numeric(pred > cutoff) != testML[,"Prediction"])
```

```
  MLM <- c(MLM, 1-err)
}
```