NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Artificial Intelligence for Fraud Detection in Motor Insurance Sector

Carolina Gonçalves Silva Simão

Work Project presented as partial requirement for obtaining the Master's degree in Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

ARTIFICIAL INTELLIGENCE FOR FRAUD DETECTION IN MOTOR INSURANCE SECTOR

by

Carolina Gonçalves Silva Simão

Work Project presented as partial requirement for obtaining the Master's degree in Advanced Analytics

Advisor: Rui Gonçalves

Junho 2021

ABSTRACT

One of the major problems in the insurance sector is related to fraud, aside from tax fraud, insurance fraud is the most practiced fraud in the world since insurance, by its nature is very susceptible to it. Fraud could be minimized by investigating each claim that occurs but that also means an increase of the costs for the insurance companies. The fraudulent clients or agents that will be caught with the investigation and the amount of money spent by looking into every new claim is not worth it. Insurance fraud is usually caught only when the fraudsters get greedy and it becomes obvious that they are involved in a scheme.

To minimize the investigation costs by only looking at suspicious claims, this project tries to identify the ones that are worth to scrutinize, through machine learning techniques. Five different predictive models will be used: Logistic Regression, Decision Tree, Random Forest, Neural Network and Gradient Boosting.

The goal is to build an optimal model that will determine which automobile claims have higher probability of being fraudulent. An efficient fraud management can reduce costs, minimize claims and increase profits. This goal was accomplished with a Gradient Boosting classifier with 400 estimators, that is able to predict correctly 49% of the fraudulent claims, with 75% less investigated claims.

There is still room for improvement by introducing the expected claim and investigation costs in the model. Since only the ones with significant costs would be worth to open an investigation, an even greater decrease in the number of investigated claims would be possible and, consequently, a decrease in the company's costs with claims. Also, it would be expected that the claims with higher costs are more likely fraudulent than the ones with small indemnities; hence, this variable could lead to a higher precision of the model. These two features will be available in the future.

Keywords: Fraud; Insurance; Automobile; Machine Learning; Python

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

An insurance contract is an agreement between the insurance company and the policyholder. This paper will have a higher focus on the motor insurance sector which means that the contract states that the company insures monetary provision over the vehicle of the policyholder, against an agreed payment, in the case of a car accident or random damaging event takes place (Viaene & Dedene, 2004).

A growing issue for the insurance companies is related to the fraudulent activity that may occur when a claim takes place, it is estimated to be between 5 and 10 per cent of the total yearly amount of indemnities paid for non-life insurance in most European countries (Viaene & Dedene, 2004). It is said that a fraud takes place when there is an intentional deception to secure unfair or unlawful gain, when individuals attempt to profit by failing to comply with the terms of the insurance agreement, and is a combination of motivation (the most common being the economic motivation) and opportunity, which makes insurance very prone to fraud. It can have different natures and types, can either be a delay or hurry in settlement, documents tampered, an individual work or of a team, the claim can be fake or inflated. In any of them, the aim is to gain unworthy benefit (Garde, 2017).

The insurance companies always stand in a position of disadvantage due to the absence of perfect information, the party that has more information often has a clear incentive to commit fraud and since the clients have private information known only to themselves, this clearly provides the opportunity to omit or lie about the facts and the circumstances in which the event happened. Fraud can be merely opportunistic, committed by normally honest people that under pressure or trough "rationalization" see a chance to receive money or it can be a carefully planned and premeditated scam (Viaene & Dedene, 2004). This is known as "The Fraud Triangle" as it is possible to see in figure 1.

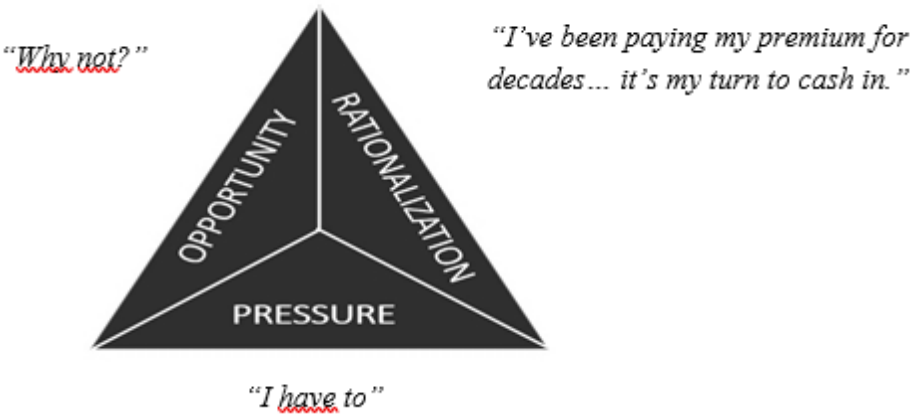


Figure 1 – The Fraud Triangle (source: Generali)

Insurance fraud is considered as a low-risk crime for the perpetrators since is not just hard to prove but also highly costly for the companies to prosecute it and, even if they do, light sentences are applied when compared to other criminal offences. As an example, there was a very recent fraud case in Portugal, in the end of 2020, where 73 individuals were accused of gaining around 1.5 million euros by deceiving insurance companies through staged accidents with more than 100 different cars. The scams were made from 2006 to 2010 and only by the end of 2020, 3 of the 73 prosecuted individuals were actually arrested (*Público*, 2020).

Due to this and to worsen the situation for the insurance sector, fraud can be committed not only by the costumers but also by the agents, the brokers and other drivers that are not the policyholders. Which means that the companies must be aware of not just external fraud but also internal (Viaene & Dedene, 2004). Fraud management, control and mitigation has been gradually gaining importance as a mean to decrease the costs for the insurance companies all over the world. Most insurers can only recently, with the improvement of the anti-fraud technologies, begin estimate how much they lose to claims. As usual, the United States of America is more advanced in this field, and the *Coalition Against Insurance Fraud* (coalition of insurance organizations, consumers and government agencies working together against fraud schemes) estimated that the fraud detection rate is increasing for most insurances, as it is possible to see in figure 2.

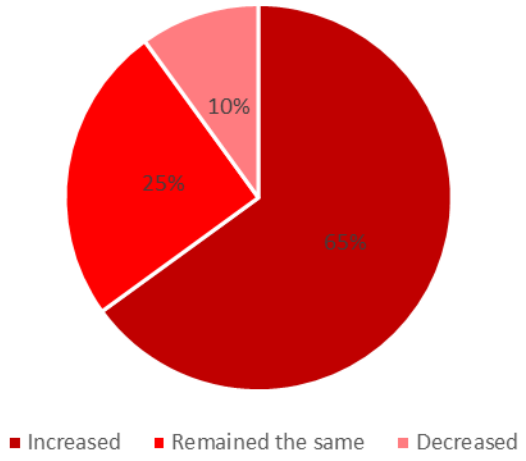


Figure 2 – Amount of Detected Frauds (source: Coalition Against Insurance Fraud)

Also, it is not possible to eradicate fraud completely due to its complexity and dynamism, besides having to be looked in order to be discovered, which is time consuming, is also less costly to negotiate with the client or simply pay the claim amount rather than sue the criminal. It is safe to say that is a combination between the cost of finding the fraudulent scheme and the cost of the claim, in order to choose the frauds that are worth to pursue (Viaene & Dedene, 2004).

Although fraud minimization is possible. As previously mentioned, an approach that has been gaining importance over the years is the formation of a “pattern”, an early warning

signal. This means that it has been found that there is a repetitive practice in a specific set of costumers that have more tendency to commit fraud, such as staged accidents, breaks in insurance periods, discrepancies between the description and the cause of the accident, non-payment or late payment of the insurance premium and other "red flags" situations. In order to find this called "pattern", documentation must be stored and available, databases must be formed and treated to check opportunistic frauds and artificial intelligence applied (Garde, 2017).

In *Tranquilidade*, now part of the *Generali Group*, fraud management has been evolving over the years. The company started with investigating the claims that the triage agent (a claim handler that works in the fraud department) was suspicious of being fraudulent to a set of business rules that summed up to a "fraud score". Only when this score is between 25 and 99 the claim is sent to the triage agent, and when equal or above 100 the case is directly sent to an investigator (a *Tranquilidade*'s internal expert). When investigated, the claim can be of four types: suspicious, suspicious with fraud belief, suspicious with evidence of fraud and withdrawal. In any of those scenarios, the fraud score is high enough to open an investigation, although in the first one it turns out to be a regular claim and the indemnity money is paid to the client. For a claim of type "suspicious with fraud belief", the investigator finds enough signs of fraud but can't prove it. The third type is the one that is actually considered a fraud, the expert finds evidence and is able to prove that the client is being dishonest, and no indemnity money is paid. The last type of claim happens when signs of fraud are found, but the client decides to back down on his claim.

The goal of this project is to improve this previous scenario of "human-made" rules by constructing a predictive model that will try to find a pattern for fraudulent clients. The human touch wasn't completely disregarded, since it was by taking these rules into consideration that some of the input variables were chosen. The objective is to predict if a claim is most likely a fraud or not, on a daily basis and in real time, in order to minimize the suspicious claims that turned out to be legitimate and to increase the number of proven fraudulent cases. That will lead to both a decrease in unnecessary investigation costs and in unfair indemnifications paid. To do so, applications of big data, data mining and machine learning techniques are the key.

First, the dataset is built by gathering information related to the client, the vehicle, the policy, the claim and the fraud criminal record. *Tranquilidade* has several tables available in an SQL server with all the necessary information, only the essential variable selection and joining of the data must be done. Then the data preparation will take place, the dataset will be treated and studied, the missing values found and analyzed, text mining application in descriptive variables will be necessary and the outliers will be mannered. In this phase, an undersampling procedure will also take place since the biggest challenge in the machine learning approach to fraud management is the highly unbalanced dataset and in *Tranquilidade* the scenario is not different. There are not many proven fraud claims which means that the target value will be positive only for a small number of records. Another setback is the great amount of non-labelled data, it is only known if the target variable is positive or negative for a small number of claims where a fraud investigation was carried out. Along with preparation

of the dataset, the variables will also be studied through some graphic and correlation analysis and the relevancy of each of them to the model will be deliberated.

After mitigating the problems and preparing the data, the project will proceed to the actual modelization part. Some supervised machine learning algorithms will be implemented such as Logistic Regression, Decision Tree, Random Forest, Neural Networks and Gradient Boosting. If the results are not satisfactory and that is due to the small amount of data, in an attempt of labeling some of the unlabeled data, an unsupervised machine learning technique known as clustering will be carried out.

To decide which model is the best fit for our case, machine learning metrics (accuracy, recall and precision) will be calculated and compared with each other. The confusion matrices will also be provided for helping the choice of the best model. After checking which one retrieves the best results, the model will be ready to be implemented. Along with the IT department of the company, a time and cost-efficient implementation procedure will be discussed. The trained model will be provided along with the imperative code to apply it when a new claim event enters the company's system, after discussing how will the necessary input features be collected.

2. LITERATURE REVIEW

2.1. INSURANCE FRAUD

Viaene and Dedene give an accurate description of an insurance contract: an arrangement in which an insurance company and a policyholder make an agreement where a monetary provision is made in case of a loss of an insurable object or person, in some specific scenarios or events, in return of a payment of a premium.

When talking about fraud in the insurance sector, it means there is an illegal act on the part of either the insurance party or the claimant party. If the fraud is committed by the insurance party, it can be by selling policies from non-existent companies, failing to submit premiums or by churning policies to create new contracts instead of renewing the existing ones. Although, the most common fraudulent activity in the insurance sector is attempted by the clients, it can consist in false or exaggerated claims, organized frauds, faked robbery or even kidnapping, falsified medical history, phantom passenger, staged accidents and so on. Several authors, including Galeotti, Rabitti and Vannuci, believe that in recession periods the number of frauds and the value of them tend to be higher. Overall, fraud is an intentional deception to secure unfair or unlawful gain, or to deprive a victim of a legal right, at any time during the period of the policy.

The *Portuguese Insurance Association (APS – Associação Portuguesa de Seguradores)* defines fraud as practice of intentional acts or omissions, even in the attempted form, with the goal of obtaining an unlawful advantage for themselves or for third parties, at the time of the contract celebration or while the contract is valid. Insurance fraud can be of several different types, internal or external, internal when is committed by employees of the insurance company (agents, brokers, managers) and external if the crime is carried out by policyholders, claimants or third parties involved. Fraud can also be executed at underwriting or when a claim takes place. Furthermore, fraud can de "soft" when committed by usually honest people that find an opportunity to gain some unfair money, or "hard" when planned or deliberated (Viaene & Dedene, 2004).

According to Clark (1989) fraudulent clients can be of different categories as well. The opportunistic, the ones that commit "soft" frauds when they see an opportunity to do so, the amateurs, carrying out "hard" frauds but not in significant numbers and the professionals which usually belong to criminal organizations with the goal of execute frauds and other types of crime. Usually the schemes executed by professionals have a higher impact in the financial status of the company than the remaining ones but, on the other hand, the "soft" frauds are higher in quantity.

Like Maio (2013) mentioned, insurance fraud is a complex, wide and dynamic problem. The major downside of insurance fraud, and also the reason why it is so common, is the high cost of dealing with such type of crime. The high expense of trying to prove a fraud, leads to not being prosecuted since the cost of doing so can be superior when compared to the claim indemnity that will have to be paid to the policyholder. In addition, even if the client is proven guilty, the sentences of such crime tend to be quite soft on the criminal. Usually it is of more

interest to the insurance companies to make a deal with the fraudulent client in order to reduce the monetary impact of the judicial means and by making the client pay, completely or partially, the claim cost. Also, it would be possible for the company to resign all the client's current policies and to keep him for signing new ones, since it would be in the company' "blacklist" (Niemi, 1995).

Fraud is often called a "victimless crime", although that might not be entirely true as Maio (2013) pointed out. These false claims have to be considered in the companies' provisions, leading to an increase in the monetary amount the insurance companies have to take into consideration to cover the losses of these fraudulent claims and, consequently, the insurance premiums presented to the clients (honest ones included) are increased. Many opportunistic fraudsters believe they are not doing anything wrong but that is not exactly true.

2.2. MOTOR INSURANCE FRAUD

One of the fields in insurance that is highly affected by the problem referred above is the motor field. Depending on the countries, the estimated percentage of fraudulent motor claims in Europe can go up to 20% (*Generali Italy*, 2020).

The motor insurance is mandatory for any individual that owns a car, and currently almost every adult has a car. That massification leads to automobile insurance being one of the best seller products which contributes to this sector being one of the most affected by fraud. Furthermore, a motor insurance contract can be quite complex since it has several obligatory coverages but also many optional ones, it can insure only the damages caused to others or cover also the damages in the insured vehicle, the occupants and the driver and even the cats and dogs that were in the car when the claim event took place.

Like in any other insurance field, in motor insurance, fraud can be internal (made by an agent, broker or any other employee) or external (executed by a client) with the external being the most frequent one. Fraud can happen at the underwriting by giving false information such as birth date, driver's license date, usual driver of the car, place of residence or by contracting a policy after the claim event and only report it later. It can also occur at the claim event by giving false declarations about the accident, exaggerating the situation or by falsifying the invoice (often, the false invoices are due to an arrangement with the car repair shop in case of material damages or with the doctor in case of body injuries). The fraudulent event can be "soft" when an accident happens and the client takes advantage of it to, for example, repair other damages in the vehicle that weren't caused by the accident, or "hard", if the claim is planned or staged.

Only recently fraud in the motor insurance field gained importance. Companies are now more aware of the amount of financial loss that fraud is responsible for. Although, the money that would be spent to investigate all the suspicious claims is not sustainable, especially when most of them are legitimate or fraudulent but impossible to prove. To reduce the investigation costs, the number of investigated frauds must be decreased to a minimum. This decline can be

accomplished if the insurance companies only investigate the claims that have a high probability of being fraudulent. This is where artificial intelligence comes in.

2.3. Artificial Intelligence in Motor Insurance Fraud

"Artificial intelligence would be the ultimate version of Google. The ultimate search engine that would understand everything on the web. It would understand exactly what you wanted, and it would give you the right thing. We're nowhere near doing that now. However, we can get incrementally closer to that, and that is basically what we work on." — Larry Page, the co-founder and developer of Google.

Garde (2017) defined claims management as a cost fruitful, profit creator and highly competitive after sales customer service, and even though it is not realistic to eliminate fraud completely, minimization is achievable. Most companies started to try to predict frauds with a set of business rules. These rule-based approach stands in fraudulent activities that can be detected by looking at straightforward and evident signals. For example, if the third-party involved is of the same family as the policyholder or if the claim happened right after the beginning of the insurance contract. The business rules are often manually written by fraud analysts and have to be adjusted frequently and new rules have to be added very often.

Although, this approach cannot detect correlations, process data in real time and is not able to find subtle and hidden "patterns" in the user behavior that are not so evident, but still a possible fraudulent behavior. With artificial intelligence these problems are overstepped. As Sharma (2019) explained, artificial intelligence automatically learns from data to control machines, make forecasts or predict actions, learns from previous history and applies it to new samples. With a machine learning based approach it is possible to find hidden and implicit correlations in the data, to automatically detect fraudulent situations, real-time processing and on top of that, is more efficient since it will be less time consuming and less expensive for the insurance companies. This is possible due to the ability of machine learning algorithms to learn form historical fraud patterns and to recognize them in future transactions. Moreover, machine learning methods perform better as more samples are available in the dataset for the model to be trained from, which is not true for the rule-based approach (Kovalenko, 2020).

The rule-based approach is considered a deterministic model, which despite from the disadvantages already discussed, it also has some advantages since it is simple to implement, it is based on the knowledge of experts in the field and the rules and the weights related to each of them is easily explained to the claim handlers. In the predictive model, the machine learning based one, it can sometimes be hard to understand and explain why a claim is being predicted as fraudulent (*Generali Italy*, 2020). Despite the disadvantages, artificial intelligence for fraud detection in insurance business is gaining importance, some insurance companies in Portugal have already a model in place and the remaining ones are already working to implement one.

The biggest technical issues the companies are facing that make the implementation of a fraud detection model harder are related to the unbalanced dataset, since only a small number of claims are proven frauds, and to the partially labeled dataset, the claims for which the target variable (fraud or non-fraud) is known are the ones where a specific investigation was carried out. Gepp, Wilson, Kumar and Bhattacharya pointed out that any fraud detection models needs to minimize these two types of misclassifications errors: missed fraudulent claims, that increase the companies' expenses by paying unfair indemnities, and claims wrongly classified as fraudulent, that wastes money in investigation. Due to the imbalanced dataset a simple approach can have more than 99% accuracy if assumes all claims as not fraudulent because of the low proportion of fraudulent claims which is why the class imbalance and the misclassification of the data samples must be considered and taken care of. This problem can be overstepped in techniques to deal with class imbalance and semisupervised techniques. After mitigating the problems, several models can be chosen to train from the data and then applied to new records.

From the statistical point of view, predicting if a claim is a fraud is possible by finding the characteristics that makes those claims different from the legitimate ones which means that fraud detection is usually handled as a classification problem (Galeotti, Rabitti & Vannucci, 2020). In *Generali Group* the most used algorithms for classification are Logistic Regression, Tree-based algorithms and Supervised Neural Networks. In *Tranquilidade* Gradient Boosting was also suggested. McCrathy, Ceccucci and Halawy (2019) proposed to their students a teaching case that should analyze motor insurance frauds using SAS, where the goal was to build an optimal predictive model to decide which of the motor claims had higher probability of being frauds, the students had to use several different classifiers as Logistic Regression, Decision Trees, Neural Networks and Gradient Boosting. The dataset had 22 input variables and after the data cleansing and pre-processing (check the missing or imprecise values, outliers impact and consideration and which variables are relevant for the analysis), the model was trained with the historical data and the new records were scored from the model which had the best fit. Benedek and László (2019) also did the same process but using an SQL server. A similar approach was made in this project, although Python was chosen to be the data analysis tool.

3. DATA COLLECTION

In *Tranquilidade* the databases are stored in postgreSQL. From the start, it was known that several "fields" of information, as illustrated in figure 3, would be necessary: information related to the policy, to the client, to the insured vehicle, to the claim event and the fraud history of the client, agent and vehicle.

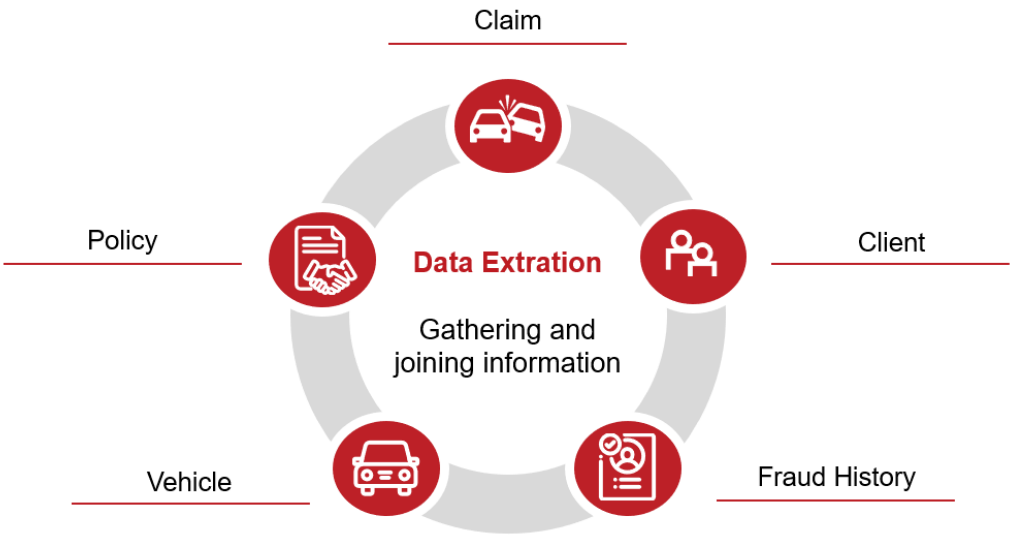


Figure 3 – Data Gathering and Extraction (source: author)

The choice of the variables that should be in the dataset was made by, besides looking at the information available in the several tables and check if made sense to make it part of the input features, considering the variables that were necessary to create business rules. These rules were created by a team of people with a considerable "know-how" of the insurance business and claim events, and it wouldn't be reasonable to just disregard them. Therefore, by looking at business rules, it was possible to understand which variables would be useful for the fraud models.

There are several business rules currently in place, having some with higher "fraud scores" (refer to *table 1*) than others, such as: the day of the week and time of the claim, time between current and last claim, age and brand of the vehicle, number of previous claims with the same vehicle or with the same policyholder, location and cause of the claim, last change in the insurance contract, begin date/period begin date/end date of the policy close to the claim event.

Rules	Scores
Begin date of the policy close to the claim event	15
Time of claim event	5
Day of the week of the claim	5
Number of previous claims with the same vehicle	15
Number of previous frauds with the same vehicle	20
Claim involving motocicles	5
Brand of the vehicle	10
Location of the claim	15
Number of previous frauds with the same policyholder	10
…	…

Table 1 – Business Rules (source: Tranquilidade)

As time goes by, more knowledge about the fraud tendencies and patterns is gained, and the scores are updated by the *Tranquilidade*'s fraud experts' team. Many of the input variables were chosen in order to comply with these rules but, not all were available or well fulfilled in the system which made them impossible to treat.

Along with this information, there is a 'hasfraud' variable that states whether the claim investigation turned out to be fraud 'Y' or if the claim was legitimate 'N'. This feature is obviously the target variable, with a small adjustment to become a binary variable with value '1' if 'hasfraud'='Y' and '0' otherwise. After selecting the relevant variables, the data from the several tables was joined still in PostgreSQL and ended up with 43 features, that are possible to identify further ahead. Then, it was exported in a *SAS* format, to hereafter be imported in Python.

3.1. POLICY DATA

Having the list of all the claims with fraud information (with the target variable 'hasfraud' fulfilled), the first data to be extracted was the one with respect to the policy. With the claim number, the policy number was easily found in order to link with policy information table. Here, 12 variables at the policy level were found:

- Policy Number: policy number associated to each claim.
- Begin Date of the Policy: the date in which the insurance contract started.
- End Date of the Policy: the date in which the insurance contract will end (if not renewed).
- Period Begin Date: the date in which the policy contract was renewed.

- Line of Business: which insurance field the contract corresponds to. For this project, this variable was used as a filter to retrieve only the automobile insurance claims (lineofbusiness='AU').

- Policy Type: 'C' for individual insurance contracts, 'G' and 'F' for group contracts, where 'F' represents the individual contract inside the company policy 'G'.

- Product: as known, only the motor line of business is taken into consideration in this project, although, inside that sector, *Tranquilidade* as a wide range of different products so that the contract meets the costumer's needs. Besides the usual individual motor insurance, several fleet products are available, products for classic/collection cars and others.

- Property Finance Type: Financial status of the vehicle, 'Y' for vehicles that are still being paid to a financial institution or with a leasing contract, and 'N' if the vehicle is paid for.

- Number of Claims: number of claims that already happened in each policy.

- Agent Number: the agent that celebrated the insurance contract with the costumer.

- Client Number: the policyholder. This is a link variable, that allows the connection with the client's information table.

- Designated Driver Number: usually it matches the policyholder but it can happen that the usual driver is not the same as the client.

3.2. CLAIM DATA

After collecting the information related to the policy and using the claim number to link to the claims table, the following 7 variables were added to the dataset:

- Date of Event: date in which the claim took place.

- Time of Event: hour at which the claim event took place.

- Reported Date: date at which the policyholder reported the accident to the insurance company.

- Accident Description: text-free variable with a small description of what happened in the claim event.

- Location Type: refers to the type of location where the claim event happened, it has four different labels: urban area, high way, road or others.

- Is Third Party Insured: ‘Y’ if the person who complaint is the one that has an insurance contract with *Tranquilidade*; ‘N’ if the complainer is not a client of the company.
- Participant: The ID number of the person that was driving the vehicle when the claim event took place. Can match the client or not.

3.3. CLIENT DATA

Using the Client Number, a link with the clients table was possible. Here, some “personal” information about the client was available and 5 more features were added to the dataset:

- Sex of the Client: ‘M’ if client is male, ‘F’ if female, ‘Z’ if unknown.
- Country of the Client: country of the client.
- County of the Client: municipality of the client
- Client Type: ‘C’ if business client, ‘P’ if particular client.
- Business Type: information available only for collective clients: small, medium or big companies.

3.4. INSURED OBJECT DATA

With the claim number, information about the insured object was easily collected (8 variables):

- Object Number: ID number of the vehicle.
- Object Type: if the vehicle is of type light or heavy
- Is Insured Object: if the vehicle is insured in the company or not.
- Year: year of the car.
- License Plate: license plate of the vehicle.
- Mark: brand of the car.
- Model: model of the car.
- Vehicle Category: group at which the vehicle belongs to; if the vehicle is light, heavy, of public transportation, of goods transportation, an emergency vehicle, or other.

Through the object number, it was possible to gather more information but only for the vehicles insured in the company. This information will be later studied since it can have too many null values to be relevant for the model, but for the time being 10 features joined the dataset.

- Vehicle Category Level 2: the same as the vehicle category but with more detail.
- Statistical Code: again, essentially the same as the vehicle category but with even more detail than *Vehicle Category Level 2*, for example, if the vehicle is of heavy type for passenger transportation or if is of heavy type for goods transportation.
- Risk Category: since only the Automobile line of business is being taken care of in this project, there is only one risk category 'E', which means the risk of the vehicles is classified according with the Statistical Code referred above. Since for every claim the risk category is the same, this variable is not relevant and it will be dropped.
- License Plate Type: category of the license plate of the vehicle. 'D' if the license plate belongs to a car of the diplomatic corps; 'N' for the Portuguese license plates; 'R' if the vehicle is a trailer; 'E' for license plates belonging to other countries; 'K' if there is no license plate; 'M' for engine number.
- Vehicle Type: other classification for the vehicle, 'A' if automobile, 'B' for motorbikes, 'C' for trucks, 'D' for bicycles, 'E' for buses, 'F' for trailers, 'G' for others.
- Driver's License Date: date at which the driver got his driver's license.
- Birth Date: birth date of the client.
- Driver's License Type: category of the driver's license of the client. 'A' for motorbikes; 'B' for light vehicles; 'C' for heavy vehicles; 'D' for public transportation vehicles.
- Transportation Type: refers to the category of the dangerous goods that the vehicle transports. 'T00' if there are no dangerous goods being transported; 'MP*x*' if dangerous materials of category *x* are being transported (*x* between 1 and 5, where 1 are the less dangerous and 5 the most dangerous materials)
- Vehicle Use Type: the category of the driver of the vehicle. 'N' for normal; 'M' for women; 'R' for retired.
- Vehicle's power: type of power of the vehicle. 'CV' and 'HP' for horse power; 'CC' for cylinder power; 'KW' for kilowatt.
- Own Damage Insurance: 'Y' if the insurance contract also coverages the damage of the client's own vehicle, 'N' otherwise.

3.5. FRAUD HISTORY DATA

As most insurance companies, the fraud information was quite unusable since all the variables, even though they existed, were simply empty. Only three variables was actually fulfilled:

- Participant Fraud: Number of frauds the participant was involved.
- Has Fraud: target variable. ‘Y’ if proven fraud, ‘N’ if legitime or not proven.
- Has Investigation: if the claim was investigated (‘Y’) in order to find evidence of fraud or not (‘N’).

As previously mentioned, after joining the different data collected, the dataset was complete and ready to be exported in a *csv* format, to therefore be imported in Python. In PyCharm the data will be studied and treated, as we will see in the next chapter.

4. DATA PREPROCESSING

In this chapter, all the necessary processes to prepare the dataset for the models will be described. By preprocessing the data, the dataset becomes more accurate, consistent and complete. This step is very important for the Machine Learning models, since the quality and the usefulness of the information that can be extracted from it, directly affects the performance of the models.

To begin with, an undersampling procedure will take place. Afterwards, some new variables will be derived from the existing ones, all of them will be treated and cleaned or dropped in case of inconsistency, and then the missing values in the dataset will be found and dealt with. After these steps, some graphs will be provided in order to have an idea of the features' importance, which ones are more eager to fraud and if some of them shall be aggregated or clustered. In order to have the input features ready to be fed into the model, a feature encoding procedure will be applied for the categorical variables and finally, a research for outliers will happen.

Before any of the referred steps, a 'problem' had to be solved. The target variable, fraud or not, was linked to the claim event. By joining all the information described in chapter two, the records were 'duplicated'. For example, if an event happened where two cars were involved, the same event will appear twice in the dataset, and if those two drivers had body injuries besides the damage in the vehicles, each insured object will have two coverages activated (body and material), which means that the vehicles will have two records each and, therefore, the claim event will have four records. Hence, the same claim event will be classified as fraudulent four times instead of just one.

To surpass the above described, the coverages (variable *type*) were transformed into several flags. The variable type was filled with the coverage that was activated. In *Tranquilidade* some different coverages mean the same, just with different names for the different products. Therefore, this variable was grouped in fourteen main types of coverage: crash, material damages, body injuries, natural catastrophes, fire or explosion, earthquakes, theft, glass breakage, replacement vehicle, other occupants' protection, third party, vandalism, legal protection and extra. After grouping it, the variable *type* was replaced by the respective fourteen binary variables for each main type of coverage in order to have only one record per object, this procedure is illustrated in figure 4.

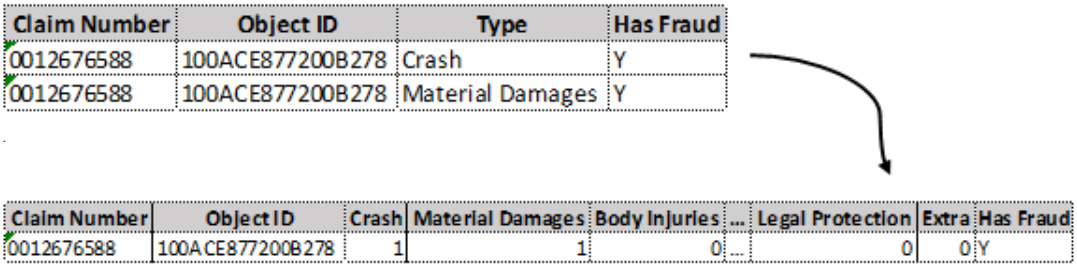
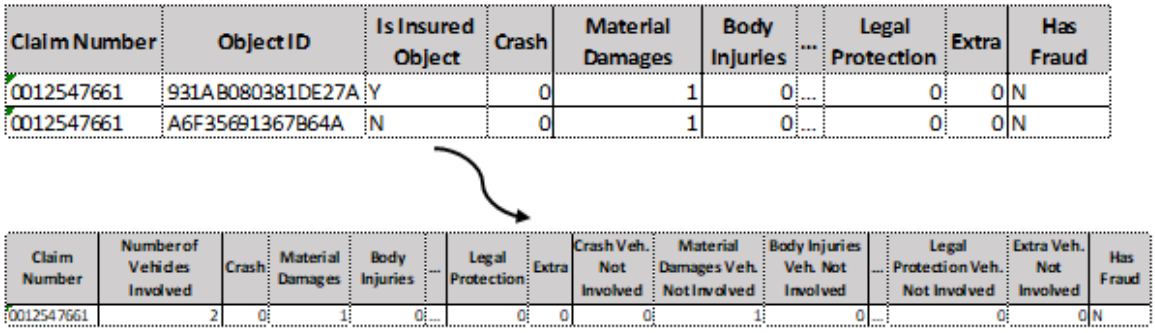


Figure 4 – Dealing with Duplicated Insured Objects (source: author)

Another two relevant variables linked to variable *type* were the *coverage initial date* and the *coverage last updated*, these two can give us a clear indication if the client decided to add more coverages to his insurance policy after signing the contract. Thus, two flags were created: one if the client made any changes in any coverage of the contract and other if this change was made less than a year ago.

Still, the claim event would have as many records as objects involved so that also had to be dealt with. First, only the vehicles that were insured in the company were considered (*isinsuredobject*='Y'), which immediately led to have only one record per claim event. This decision was due to the fact that most of the information about the insured object was only fulfilled for those vehicles. Although, the coverages that were activated were also fulfilled for the vehicles that were not *Tranquilidade*'s responsibility, hence fourteen more binary variables related to the activated coverages of the second vehicle were added to the dataset. Besides these variables, the variable *Number of Vehicles Involved* was also calculated by counting the number of vehicles that took part of the claim event. Figure 5 illustrates the procedure described. With these procedures, the dataset had only one line per claim event which made the target variable accurately linked again.



Claim Number	Object ID	Is Insured Object	Crash	Material Damages	Body Injuries	...	Legal Protection	Extra	Has Fraud
0012547661	931AB080381DE27A	Y	0	1	0	...	0	0	N
0012547661	A6F35691367B64A	N	0	1	0	...	0	0	N

Claim Number	Number of Vehicles Involved	Crash	Material Damages	Body Injuries	...	Legal Protection	Extra	Crash Veh. Not Involved	Material Damages Veh. Not Involved	Body Injuries Veh. Not Involved	...	Legal Protection Veh. Not Involved	Extra Veh. Not Involved	Has Fraud
0012547661	2	0	1	0	...	0	0	0	1	0	...	0	0	N

Figure 5 – Dealing with more than One Object Involved (source: author)

4.1. UNDERSAMPLING

Fraud has a problem that is common to every insurance company in any country in the world: an unbalanced dataset. A fraud investigation is a costly process, which leads the companies to only investigate the claims where there is strong evidence that some doubtful activity took place and if the cost of the claim is worth it when compared to investigation cost. From the small percentage of claims that are actually investigated, only a small part of those have been proven and classified as fraudulent claims by the investigators.

The situation described leads to a very unbalanced dataset since most of the records will be classified as "not fraudulent". In *Tranquilidade*'s case, only 1% of the dataset has the fraud label equal to 'Y'. Claims from January 1st of 2016 until September 30th of 2020 were extracted. It reflected in almost 1 million records and only around 8 thousand of those were

classified as frauds. Which meant, that 99% of the dataset had a label of '0' in the target variable, as reflected in figure 6.

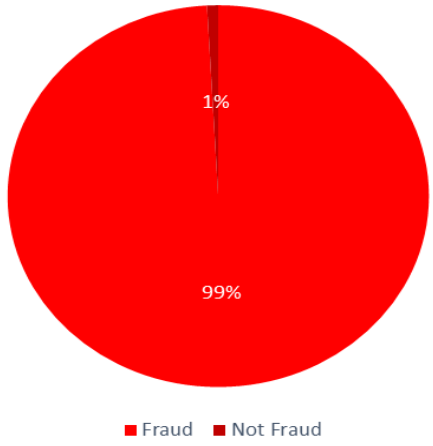


Figure 6 – Original Target Class Weights (source: author)

Although, not all the claims are relevant for this analysis. It only makes sense to consider the claims that were investigated, otherwise, claims considered legit may be wrongly classified since there was not an expert investigating it. To exceed this, only the claims that had the variable *Has Fraud Investigation* with label 'Y' should be considered. Of the 1 million claims mentioned above, 58.820 of them were indeed investigated and considered for the project, which changed the original weights of the pie chart to the ones in figure 7, with 12% of the dataset with a positive label in the target variable.

Figure 7 – Target Class Weights considering only Investigated Claims (source: auhor)

Even so, this dataset could not be used since it would still be very disproportionate, the model would simply learn that most of the records are classified as '0'. Even if it classified the whole set as "not fraudulent", the model accuracy would still be around 90%. This accuracy is 'false' since the model is being considered accurate without predicting any fraud at all. In contrast, the precision would be quite low since is a metric that quantifies only the number of correct positive predictions, which is obviously a problem.

To surpass the problem mentioned above, an undersampling procedure was applied to this last dataset, where only the investigated claims were considered, in order to have a dataset that had a 50-50 percent chance of a claim being fraudulent or not, as shown in figure 8. In python, with *RandomUnderSampler* package, random samples of the majority class (not fraudulent) were randomly picked and removed from the data in order to get 8 thousand fraudulent and 8 thousand legitimate records. Thus, the dataset went from 1 million records to 60 thousand, and after applying the undersampling to the final 16 thousand records. The already small dataset became even smaller, but at least balanced.

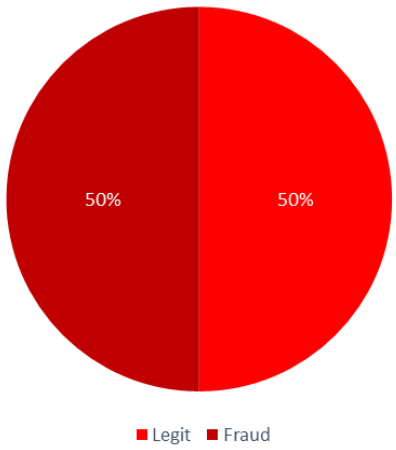


Figure 8 – Target Class Weights after Undersampling (source: author)

4.2. CREATION OF NEW VARIABLES

Some variables by themselves don't provide any relevant information, but combined, cleaned or treated, can improve the model. In this sub-section, the input features will be treated but only later their relevancy to the several machine learning models will be studied. In *table 2* it is possible to see a resume of the new variables created.

In the original dataset, several ID variables were found: *policy number*, *client number*, *agent number* and *license plate* (already explained in *chapter 2*). These variables, without any treatment, are not of any use to the model so they will be dropped. However, before dropping them, some information needs to be extracted. The number of previous frauds that each

policy/client/agent/vehicle had at the time of the claim were counted before dropping the variables. The referred variables were all dropped expect for *client number*, that before being discarded was used with the variables *participant ID* and *designated driver*. With them, two new binary variables were created, with label '1' if the participant and the client were not the same (*Participant dif Client*) and if the designated driver and the client didn't match (*Driver dif Client*).

The dataset was also rich in date type variables, with respect to both claim and policy. With the *date of loss* and the *reported date* of the claim, the number of days between the claim event and the reported date to the insurance company was calculated, then the number of days was grouped in intervals of 10% with respect to the number of claims. Also, the day of the week at which the claim event took place was extracted from the variable *date of loss*. *Time of loss* could have been of interest as well, for example, a pattern could be found that most frauds tend to happen during the night. Although, most records were missing which made this variable less usable and therefore dropped.

The variable *policy seniority* was created from the number of years retrieved from the difference between the *begin date* and the *period begin date* of the policy. Again, this variable was grouped in equal intervals of 10% with respect to the number of claims. Then, the variable *begin date* was no longer necessary and therefore dropped. Before doing the same with the *period begin date*, by obtaining the difference between this variable and the *date of loss* of the claim, the *number of days without claims* was calculated and grouped in intervals as well.

Number of years of driver's license and *vehicle's age* at the time of the claim event were derived from the difference between the *date of loss* and the *year of the driver's license*, and between the *date of loss* and the *year of the vehicle*, respectively. The *client's age* was also calculated from the amount of years between the *date of loss* and the *date of birth* of the client.

Original Variables	New Variables
Policy Number	Number of Frauds of the Policy
Client Number	Number of Frauds of the Client
Agent Number	Number of Frauds of the Agent
License Plate	Number of Frauds of the Vehicle
Client Number	Participant Dif Client
Participant ID	
Client Number	Driver Dif Client
Designated Driver	
Date of Loss	Number of Days to Report the Claim
Reported Date	
Date of Loss	Day of the Week of Claim
Begin Date of the Policy	Policy Seniority
Period Begin Date of the Policy	
Period Begin Date of the Policy	Number of Days Without Claims
Date of Loss	
Year of the Driver's License	Number of Years of Driver's License
Date of Loss	
Year of the Vehicle	Vehicle's Age
Date of Loss	
Date of Birth of the Client	Client's Age
Date of Loss	

Table 2 – Variables Correspondence (source: author)

Finally, *description of the claim* was taken care of. This variable is of text free, where the claim handler describes what happened during the claim event. Some basic text mining techniques were applied, first by lower casing all the characters, then by looking for some key words and creating flags for each of them. Hence, *description of the claim* gave place to thirty-six new binary variables: case 10, case 11, case 12, case13, case 14, case 15, case 16, case 17, case 20, case 21, case 30, case 31, case 40, case 50, case 51, case 52, case 53, case 60, case 61, case 62, case 63, case 64, case 70, case 71, case 72, parking, minor crashes, chain crashes, rear, crash, windows, assistance, recede, vandalism, fire, theft and run over. The *case x* variables may not mean anything to someone out of the insurance field but there is a national correspondence table created by the *Portuguese Insurance Association (APS – Associação Portuguesa de Seguradores)*. In this table is possible to find the description of each case (illustration in *appendix*):

- Case 10: vehicles circulating in the same direction, vehicle X with his rear hit.
- Case 11: vehicles circulating in the same direction but in different roads, vehicle Y changes its way invading vehicle X path.

- Case 12: vehicles circulating in the same direction but in different roads, hit without proof of path changing.
- Case 13: vehicles circulating in the same direction but in different roads, vehicle X was stopped.
- Case 14: vehicles circulating in the same direction but in different roads, vehicle Y hits vehicles X when turning left in a road crossing or junction.
- Case 15: vehicles circulating in the same direction but in different roads, vehicle Y hits vehicles X when turning left to leave the road or enter a parking spot.
- Case 16: vehicles circulating in the same direction but in different roads, vehicle Y overtakes a row of vehicles using the road in the contrary direction with all the necessary cautions and the vehicle X is turning left on a road crossing or junction.
- Case 17: vehicles circulating in the same direction but in different roads, vehicle Y overtakes a row of vehicles using the road in the contrary direction and the vehicle X is turning left to leave the road or enter a parking spot.
- Case 20: vehicles circulating in different directions but on the same road, vehicle Y hits vehicle X while crossing the road axis.
- Case 21: vehicles circulating in different directions but on the same road, vehicle Y and vehicle X both crossing the road axis.
- Case 30: vehicles coming from different roads, vehicle Y hits vehicle X while crossing the road axis.
- Case 31: vehicles coming from different roads, vehicle Y and vehicle X both crossing the road axis.
- Case 40: vehicle X hit while totally stopped or parked.
- Case 50: vehicle X didn’t respect traffic signs and has hit while vehicle Y overtook it.
- Case 51: while doing a reverse or making a u-turn, vehicle Y hit vehicle X.
- Case 52: starting to drive after being stopped or parked, vehicle Y hit vehicle X.
- Case 53: vehicle Y opens the door invading the road and gets hit by vehicle X.
- Case 60: chain shock, where each vehicle gets hit only by the vehicle that was immediately behind.
- Case 61: chain shock, where vehicle C hits vehicle B that hits vehicle A a second time, since it had hit it before.

- Case 62: chain shock, where each vehicle gets hit only by the vehicle that was immediately behind, through projection done by the last vehicle.
- Case 63: chain shock, where each vehicle is responsible for the damage of the front vehicle.
- Case 64: chain shock with more than three vehicles, where each vehicle is responsible for the damage of the front vehicle.
- Case 70: vehicle Y (motorbike) hits vehicle X while overtaking it outside the road way.
- Case 71: vehicle Y (motorbike) hits vehicle X while overtaking it from the right side.
- Case 72: vehicle Y (motorbike) hits the open door of vehicle X while overtaking it from the right side.

4.3. GRAPHICAL ANALYSIS

The graphical view makes data easier to understand and information easily compared. To start some simple double axis bar graphs were constructed for each of the variables, where it is possible to see the number of claims and the fraud percentage for each class in each variable. The information that was chosen to be seen in the graphs can give a hint of the relevancy of each variable to the model, since it makes the correlation with the target variable easily seen. All the graphs are available in the *appendix*.

For example, by looking at the graph of figure 9 it is possible to see that the probability of having a fraud almost doesn't change according to the sex of client.

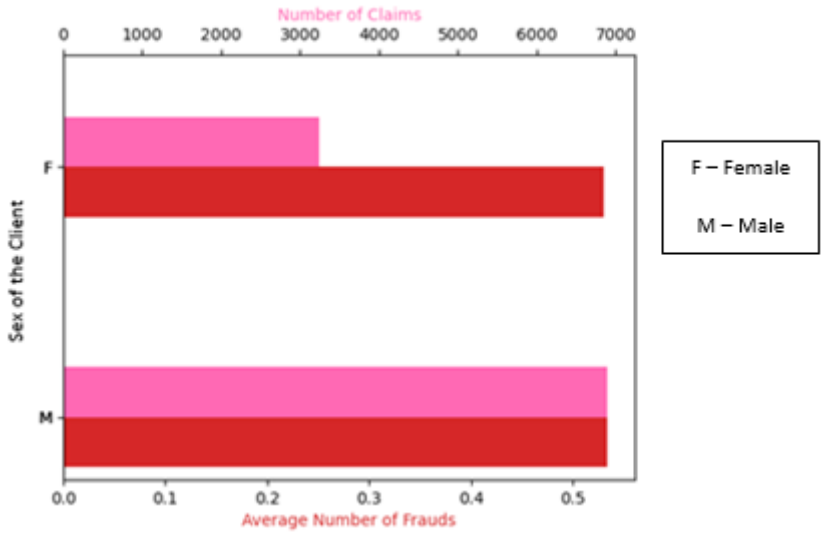


Figure 9 – Number of Claims and Average Number of Frauds by Client's Sex (source: author)

On the other hand, for example, it is possible to check that the variable *location type* will likely be relevant to the model since there is a higher fraud probability for the claim events that happened on a highway and on a road than for the ones that occurred in an urban area or other locations, as shown in figure 10. There is still a third group of variables where it is not possible to draw any conclusions since most of the records belong to just one of the classes. Even if the fraud probability is highly different, there is not enough claims in each class to consider it relevant, like the *type of driver's license* that is possible to see in figure 11.

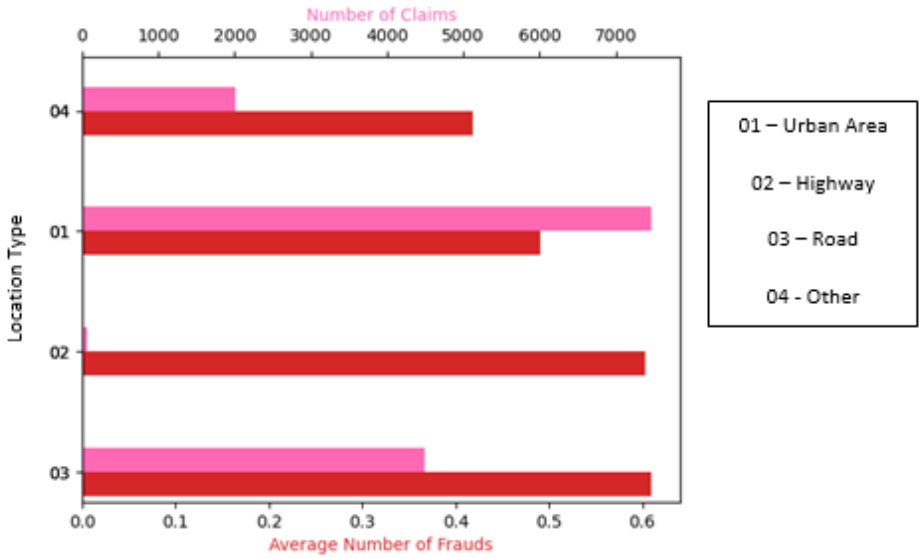


Figure 10 – Number of Claims and Average Number of Frauds according to Location (source: author)

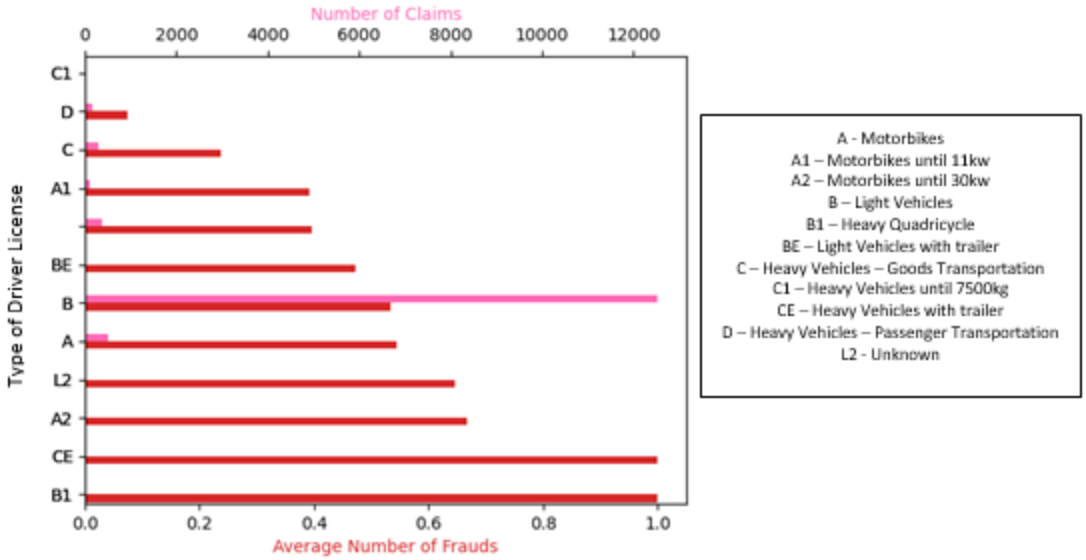


Figure 11 – Number of Claims and Average Number of Frauds according to the Type of Driver's License (source: author)

Usually the bar graphs are enough to give a perspective of the correlation of each variable with the target variable, except for two variables: the *mark* of the vehicle and the *county* of client. Here the bar graphs had too many different classes to be understandable at naked eye, hence, a scatter plot was the second obvious choice for the graphical visualization of both *mark* and *county*. It was straightforward that it would be of interest to group both these variables into clusters. To simplify, these two variables were grouped according to the average number of frauds.

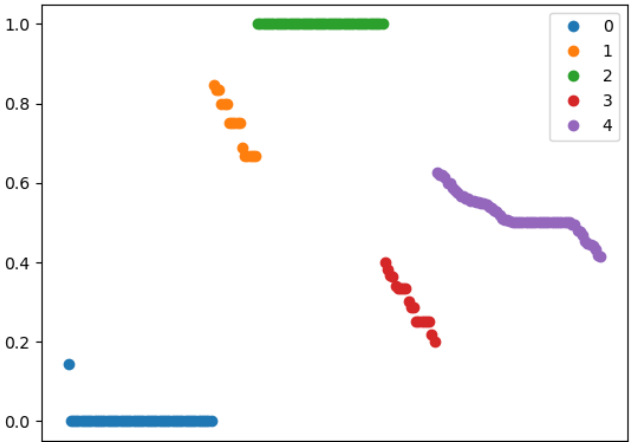


Figure 12 – Mark Scatter Plot (source: author)

In figure 12, is possible to see that the variable *mark* was grouped into five clusters and the same number of clusters was chosen for *county*, as it is obvious in figure 13. For the

variable *mark*, the brands that were classified with the label '0' were the ones less eager to fraud, followed by the brands with the label '3', label '4', label '1' and the ones with label '2' being the ones with higher average number of frauds. By grouping *county*, the clients with county label '2' were the ones with a lower average number of frauds, with label '1', label '3', label '0' coming after and label '4' the ones with higher fraud probability.

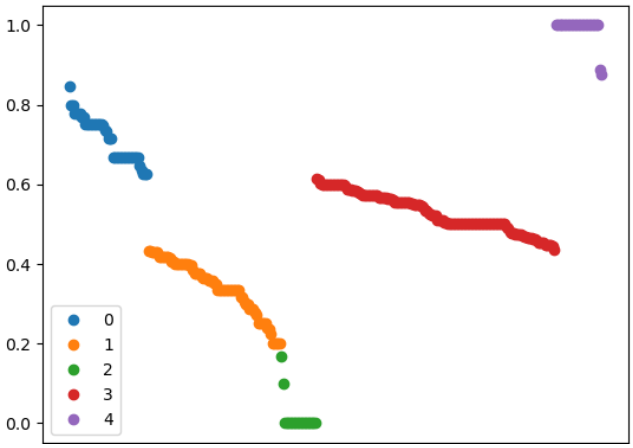


Figure 13 – County Scatter Plot (source: author)

4.4. TREATING THE NULLS

Some of the variables in the dataset had missing records. In the case that the percentage of empty records was above 15%, those variables were immediately dropped, which was verified for the variables *sex of the client* and *sex of the insured*. These two features are equivalent but the first came from the client information data and the second from the insured object information data. From the graphical analysis, it was possible to see that the classes of the variable sex (any of them) didn't have a significant difference in the fraud probability, hence, both the *sex of the insured* and *sex of the client* were no longer part of the dataset.

The *years of driver's license* and the *age of the client* were removed from the dataset as well, due to their high percentage of missing records. Also, the fourteen binary variables created to represent the coverages that were activated for the vehicle involved that was not insured in the company were dropped. Since 25% of the claims only had the insured vehicle involved in claim event, 25% of the information related to the second vehicle was missing (check *table 3*).

The remaining variables that still had missing values but below the 15% limit, were all categorical and a simple replacing procedure was applied: the empty records were replaced by the most common class of the respective variables.

Variable	Missing Values %
Sex of the Client	27.72
Vehicle Category	1.63
Statistical Code	0.18
Sex of the Insured	26.54
Driver's License Type	2.73
License Plate Type	0.38
Vehicle Type	4.81
Transportation Type	2.60
Vehicle Use Type	10.54
Vehicle's Power	10.04
Vehicle Category Level 2	0.18
Own Damage Insurance	0.18
Years of Driver's License	25.55
Vehicle's Age	0.38
Age of the Client	25.34
Brand of the Vehicle	0.21

Dropped Variables

Table 3 – Missing Values Percentage (source: author)

4.5. Label Encoder

The machine learning models don't accept non-numerical variables, which leads to the need of replacing their format to make them ready for the model.

Hence, each target label of each of the categorical variables was replaced by a number, with a value between 0 and the number of classes minus 1.

4.6. Outlier Detection

In Machine Learning, it is common to come across situations where outliers are present in the dataset. These outliers are extreme values or values that do not follow the pattern in the data, odd values that diverge from all other.

For the outlier detection, z-score was the chosen concept. Also known as standard score, it helps to understand if a data value is greater or smaller than the mean and how far away it is from it.

$$z - score = \frac{x - mean}{standard\ deviation}$$

A z-score of 3 was applied, meaning that we are looking for data records that are 3 points away from the mean, which in most literature is far away enough to be considered an outlier.

Z-score classifies 0.014% of the dataset as outliers so it is was decided not to remove them from the data, since it was not relevant enough to disturb the learning process of the models.

4.7. CORRELATION

Correlation shows the strength of a relationship between two variables and is expressed numerically by the correlation coefficient, with values that range between -1 and 1. This implies that as one variable moves, either up or down, the other variable moves in the same direction if the coefficient is positive. A negative correlation means that two variables move in opposite directions, while a zero correlation implies no linear relationship at all.

The correlation matrix was drawn (full matrix in the *appendix*) to have a hint of the relevancy of the variables to the model by checking the dependency between all the variables with the target variable, in figure 14 the variables with highest correlation can be seen.

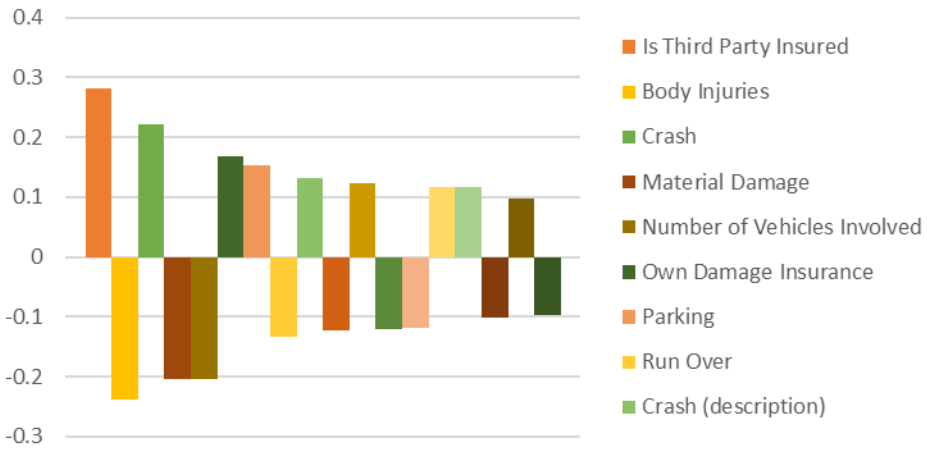


Figure 14 – Correlation with Target Variable

5. MODELLIZATION

Before feeding the data into the models, the dataset was split into two sets. With 70% of the records, the train set was built and the remaining 30% for the test set. Accuracy and error estimates on the training data are not good indicators of the performance of the model on future data, since this data will most likely not be exactly the same as the data used for training. The metrics used on the training set are essentially used to measure the overfitting degree of the classifier.

For this reason, the test set with 30% of the records of the original dataset was also created. The model doesn't learn anything from these samples, it is a completely independent set, which makes the performance metrics more qualified to define the performance of the model. For both sets, stratified samples were used to make sure that each class is represented with approximately equal proportions of fraudulent and not fraudulent claims in both subsets, since there was no advantage in going back to unbalanced datasets.

The original dataset had multiple features with different degree of magnitude, range and units. Even though for classifiers as Decision Trees or Random Forests this is not an obstacle, some of the Machine Learning algorithms are highly sensitive to these features. To surpass the situation described, a scaling technique known as standardization was used. The values are centered around the mean with a unit standard deviation:

$$X' = \frac{X - \mu}{\sigma}$$

After standardizing both subsets, they are ready to be fed into the models. Three machine learning metrics will be used to measure the classifiers: accuracy, recall and precision. Accuracy represents the percentage of correct predictions the model was able to retrieve. Recall (also referred as sensitivity) gives the fraction of actual positives that were correctly identified, while precision shows the proportion of positive identifications that were actually correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives and *FN* = False Negatives.

5.1. LOGISTIC REGRESSION

Logistic Regression is a very popular artificial intelligence approach for classification problems, it predicts the probability that a record belongs to one of the two possible classes. Linear Regression was not used since it works only for continuous variables and uses Ordinary Least Squares as error minimization technique. In the Logistic Regression, Ordinary Least Squares is not a good estimator since it performs well in average observations but not on "describing" the data points, it doesn't make sense to have probabilities below 0 or above 1.

With Logistic Regression the output of the model is a probability between 0 and 1 that reflects the likelihood of the event occurrence. The input of the model are the linearly scaled input features, on these features is then applied a nonlinear transformation to make sure that the range of the output is in the [0,1] interval:

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

To obtain the vector of parameters, Maximum Likelihood Estimation is applied to find the $\hat{\beta}$ vector of estimates that maximizes the likelihood of observing the sample, by an iterative algorithm that starts with arbitrary values. Since the estimated equation looks like $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_k x_{ki}$, from the value of the observation $\hat{y}_i$ the estimation probability is obtained:

$$\hat{P}(y_i = 1|X_i) = \frac{e^{\hat{y}_i}}{1 + e^{\hat{y}_i}}$$

After importing the *LogisticRegression* classifier from the *Linear Model Sklearn* package, the model was trained with the training set and the information learned from the data was stored. What was learned was applied to the test set in order to predict the labels of the target variable. To measure the performance of the model, accuracy, recall and precision were calculated. As it is possible to see in figure 15, for any of the performance metrics, the values are around 72%. An accuracy of 71,35% means that the model is predicting 71,35% of the overall data correctly. Also, with recall, it is possible to know that the classifier correctly identifies 71,99% of all frauds and with precision, when the model predicts a fraud, is correct 72,59% of the time.

Metrics	Training Set	Test set
Accuracy	72.75%	71.35%
Recall	71.91%	71.99%
Precision	74.65%	72.59%

Figure 15 – Performance of Logistic Regression Classifier (source: author)

5.2. NEURAL NETWORK

An Artificial Neural Network is based on the human brain, a collection of 'neurons' is created and connected, allowing them to communicate with each other and therefore the network is asked to solve a problem. It attempts to do it over and over, each time adjusting and strengthening the connections that lead to the best solutions and diminishing the ones that lead to failure. In the Neural Network, each neuron will receive a set of weighted (w_i) inputs (x_i) from the neurons of the previous layer and these are added together to form the activation value:

$$z = \sum_{i=1}^{n} w_i x_i + w_0$$

The activation value is then 'squashed' by an activation function to therefore retrieve the output value $y = a(z)$, its responsible for transforming the inputs in outputs. When hidden layers are present, the activation functions are needed to introduce nonlinearity in the network. The training algorithm adjusts the weights in the network in order that the inputs that help achieve good results are increased, and weakened otherwise. It starts with the weights set randomly and then small adjustments are made until a good result is achieved. Not only the weights are adjusted during the training, the error signals which 'back propagate' from the output to the input neurons also have small changes in order to reduce it.

In this specific situation a multi-layer perceptron is being used, which means that besides having the input layer (where each neuron gets only one input) and the output layer (each neuron directly goes to the outside), there is also some hidden layers that connect the input and output layers. The hidden layers are responsible for classifying the characteristics. For the

implementation of the Neural Network, *MLPClassifier* from the *Neural Network Sklearn* package was used, with the following parameters:

- activatation = 'relu', rectified linear unit function as activation function for the hidden layers:

$$f(x) = \max(0, x)$$

- alpha = 1, regularization term parameter
- hidden_layer_sizes = (100, 50, 50), number of neurons in each of the three layers
- learning_rate = 'constant', constant learning rate given by learning_rate_init parameter
- learning_rate_init = 0.01, the initial learning rate used
- solver = 'adam', refers to a stochastic gradient-based optimize used for weight optimization

After training the classifier with the training data, storing what was learned and applying to the test set, the performance results obtained are the ones described in figure 16. The accuracy is quite close to the one obtained with the Logistic Regression Classifier. Although, the number of frauds correctly identified is a bit higher, with a 74,57% value. By looking at precision, it is possible to see that it slightly reduced, when the classifier labels a record as fraud, it is correct 70,20% of the time.

Metrics	Training Set	Test set
Accuracy	74.39%	71.66%
Recall	76.49%	74.57%
Precision	73.29%	70.20%

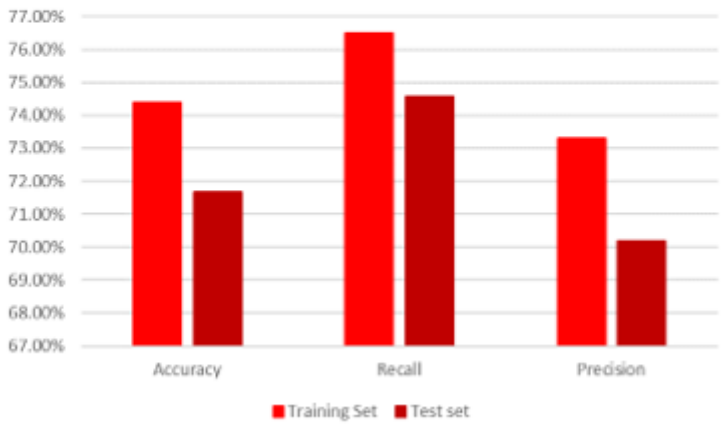


Figure 16 – Performance of Neural Network Classifier (source: author)

5.3. DECISION TREES

Decision Trees models split the data multiple times according to certain cutoff values in the input features and, through the splitting, different subsets of the dataset are created. This algorithm is easily understandable since the split rules used are easy to interpret. The intermediation subsets are the internal or split nodes, while the final subsets are the terminal or leaf nodes. To predict the outcome in each leaf node, the average outcome of the training data in the node is used.

Trees can be used for classification or regression, here the classification model is the appropriate one. The goal is to discriminate between classes by obtaining leaves as pure as possible. For each input variable the dataset can be divided into several partitions so the model will evaluate each one and establish the best partition. For the selected partition, the process is repeated and stops when a stopping criterion is reached, which means that all samples for a given node belong to the same class, there are no remaining attributes for further splitting or that there are no samples left. In this case, the chosen metric for the attribute selection was entropy, therefore the attribute selected is the one that has more information gain:

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Hence, if we choose the attribute X to split the D in n partitions, then the information needed to classify D is:

$$Entropy_X(D) = \sum_{j=1}^{n} \frac{|D_j|}{|D|} Entropy(D_j)$$

And the information gained by branching on attribute X is:

$$Gain(X) = Entropy(D) - Entropy_X(D)$$

DecisionTrees from the *Tree Sklearn* package was used to apply the Decision Tree Classifier. For this classifier, the parameters used were the ones below:

- cpp = 0.0007, complexity parameter used for Minimal Cost-Complexity Pruning, the subtree with the largest cost complexity that is smaller than 0.0007 will be chosen
- criterion = ‘entropy’, information gain to measure the split quality
- max_features = ‘sqrt’, the number of features to consider when looking for the best split:

$$\text{max\_}features = \sqrt{n\_features}$$

- min_impurity_decrease = 0, A node will be split if this split induces a decrease of the impurity greater than or equal to 0
- min_samples_leaf = 0.05, 5% of the sample is the minimum number to be at a leaf node
- min_weight_fraction_leaf = 0, the minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node
- splitter = 'best', the best split was the strategy used to choose the split at each node

As usual, accuracy, recall and precision were calculated to evaluate the performance of the model as it is possible to see in figure 17. The metrics decreased significantly with this classifier, for any of the metrics the values are always around 64% which is almost 10 percentage points below the previous two models.

Metrics	Training Set	Test set
Accuracy	63.84%	63.37%
Recall	63.44%	64.35%
Precision	65.70%	64.82%

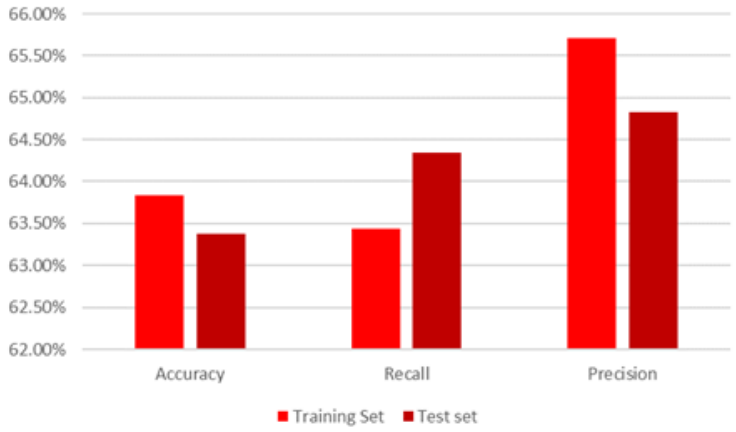


Figure 17 – Performance of Decision Tree Classifier (source: author)

With this classifier, it was also possible to know which variables were more relevant for the learning process of the model. As referred in figure 18, the most important features are the binary variables with value '1' if the coverages Crash and/or Material Damages were activated. Responsible for 23%, the variable *Is Third Party Insured* was also important, followed by the *Location Type*, the *Vehicle Category Level 2*, the *Number of Days Between the Event and the Reported Date* and the *Participant Different from Client* variables.

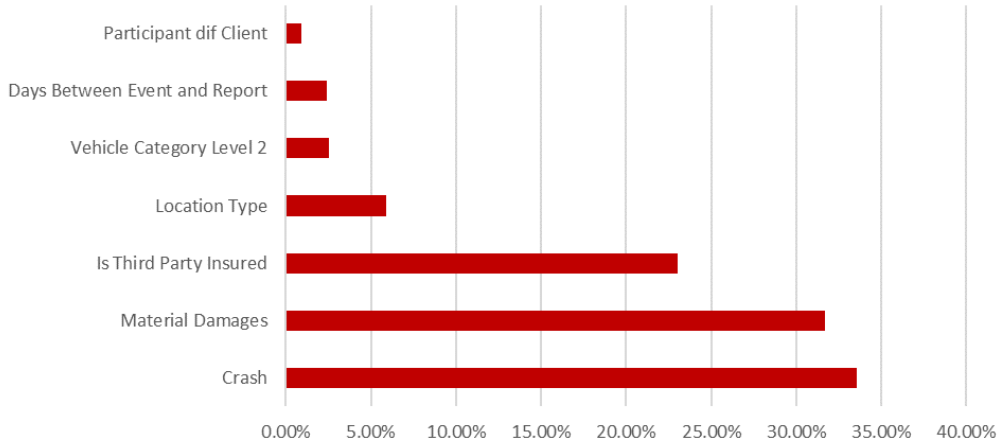


Figure 18 – Features Importance of Decision Tree Classifier (source: author)

5.4. RANDOM FOREST

A Random Forest, like the name implies, consists in a large number of individual Decision Trees that operates as an ensemble. Each individual tree in the Random Forest outputs a class prediction and the model's output is found by majority voting. By common sense it is possible to understand that a large number of relatively uncorrelated trees will outperform a single one, and this is why Random Forest is considered an ensemble of Decision Trees. The goal of ensemble methods is to combine the predictions of several base estimators built (Decision Trees in this case) with a given learning algorithm in order to improve it over a single estimator, it combines several machine learning techniques into one predictive model in order to decrease variance. The Decision Trees outputs are independent and if the classifiers have the same individual accuracy, then the majority voting is guaranteed to improve on the individual performance.

Since the goal of a Random Forest is to decrease variance, the ensemble is made of classifiers built on bootstrap replicates of the training set and the classifiers outputs are combined by plurality vote. From the training set, *t* bootstrap samples are created which, like *Bagging*, induces diversity but instead of samples with replacement from the original training set (bootstrap sampling) to create a new training set of length *N*, the size of the bootstrap samples is the same as the size of the training set. For each split of each node, in each individual Decision Tree, *x* attributes are randomly chosen out of the total *X* attributes and then one attribute out of *x* is chosen to split the node.

From several literatures, it was found that 25-50 Decision Trees are enough to get the ensemble error to level off and are known to run more efficiently on large datasets (which is not the case in this project) and can take care of a large number of input features.

With *RandomForestClassifier* from the *Ensemble Sklearn* package, a Random Forest was applied to the training set in order to learn from the data and that information was then stored. The paramenters chosen for the Random Forest Classifier were:

- oob_score = True, using out-of-bag samples to estimate the generalization accuracy
- criterion = 'entropy', information gain to measure the split quality
- min_samples_leaf = 0.05, 5% of the sample is the minimum number to be at a leaf node
- n_estatimators = 20, 20 trees in the forest

Afterwards, the information learned was applied to the test set and the performance metrics obtained are the ones in figure 19. As expected, given that Random Forest is an ensemble of Decision Trees, all the measures were improved when compared to the *DecisionTreeClassifier*. Although, the metrics are all still quite below the Logistic Regression and the Neural Networks classifiers.

Metrics	Training Set	Test set
Accuracy	65.67%	64.39%
Recall	70.07%	68.42%
Precision	65.81%	64.61%

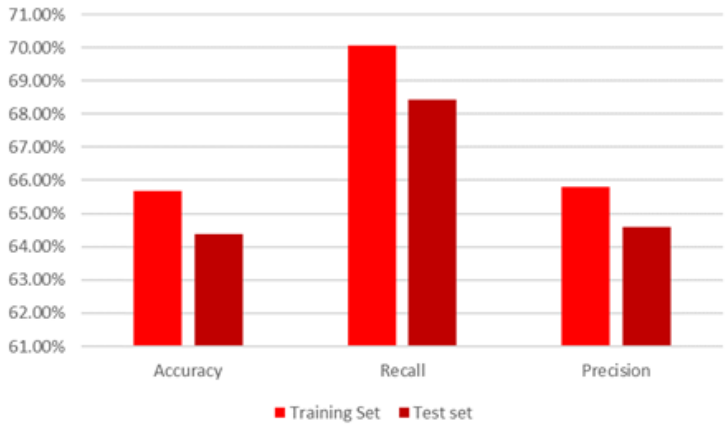


Figure 19 – Performance of Random Forest Classifier (source: author)

Such as the Decision Tree classifier, with the Random Forest model it is also possible to know the features' importance. In figure 20 is possible to see which variables are more relevant for the prediction of the target variable and the respective percentage of its contribution.

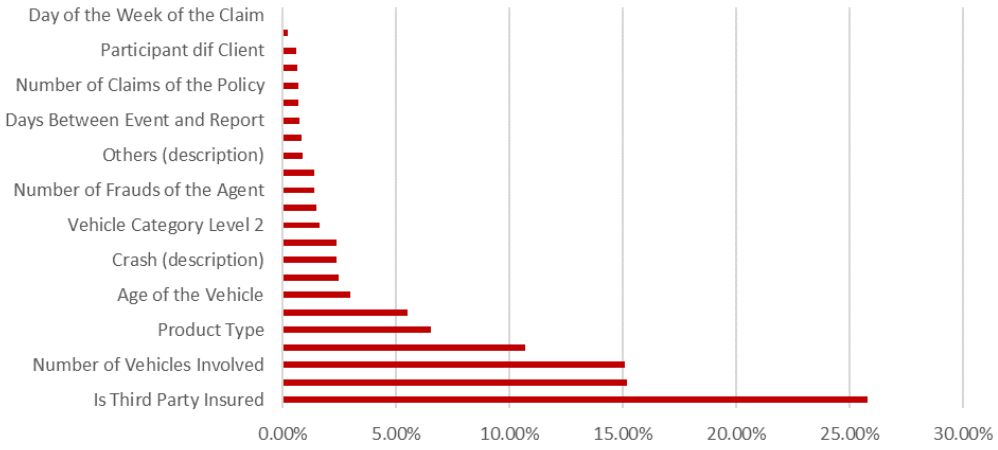


Figure 20 – Features Importance of Random Forest Classifier (source: author)

5.5. GRADIENT BOOSTING

The idea of boosting implies that a weak learning algorithm can be modified to perform better. The 'boosting' of an algorithm is mainly done by focusing on reducing bias, since the base models that are often considered for boosting are low in variance and high in bias. Differently from bagging, different models in boosting can't be done in parallel.

Boosting is an iterative procedure where new models are influenced by performance of previously built ones. The goal of the new model is to become an expert in the records that were wrongly classified by the previous models. At each iteration, the previous misclassified examples get a larger weight in order to have more importance in the new model and later the classifiers are combined by an accuracy weighted vote.

While in the Random Forest algorithm the trees created are always full sized, each tree is worth the same and the order in which the trees are built is ignored, in the Gradient Boosting initially only several single leaves are created (which represent the initial guess for the weights) and then a controlled sized tree is constructed from errors made in the previous tree and keeps building trees until it reaches a threshold or fails to improve the fit. To apply the Gradient Boosting classifier to the dataset, *GradientBoostingClassifier* also from the *Ensemble Sklearn* package was imported and the chosen parameters the following:

- criterion = 'friedman_mse', information gain to measure the split quality
- learning_rate = 0.1, learning rate shrinks the contribution of each tree by 10%
- loss = 'deviance', the loss function to be optimized, 'deviance' refers to deviance (= logistic regression) for classification with probabilistic outputs
- min_samples_leaf = 1, one is the minimum number to be at a leaf node

- n_estimators = 400, 400 boosting stages to perform

After learning from the train set and applying the information learned to the test set, the metrics outputs were the ones in figure 21. An accuracy of 73,52% was reached, meaning that the model is predicting 73,52% of the overall data correctly. The recall and precision metrics had values of 74,34% and 74,72%, respectively. Hence, like the Logistic Regression and the Neural Network models, the Gradient Boosting classifier behaves quite well.

Metrics	Training Set	Test set
Accuracy	78.41%	73.52%
Recall	78.97%	74.34%
Precision	79.46%	74.72%

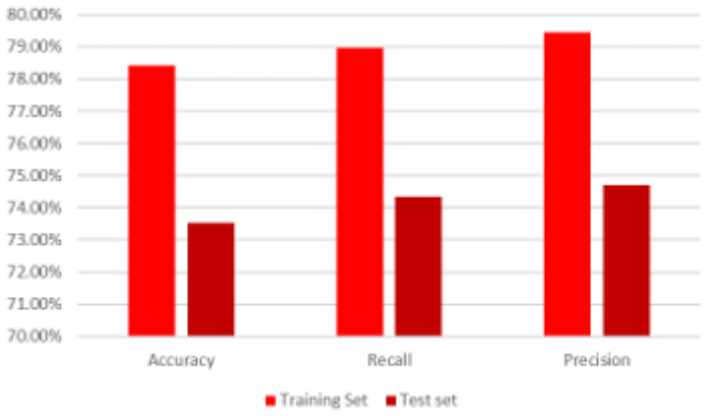


Figure 21 – Performance of Gradient Boosting Classifier (source: author)

With figure 22 is possible to understand the contribution of each variable to the learning process of the Gradient Boosting model. The more relevant variables were also of importance on the previous models so it is possible to conclude that independently of the model, even though the contribution percentage might be different, the most importance features are often the same regardless of the chosen classifier.

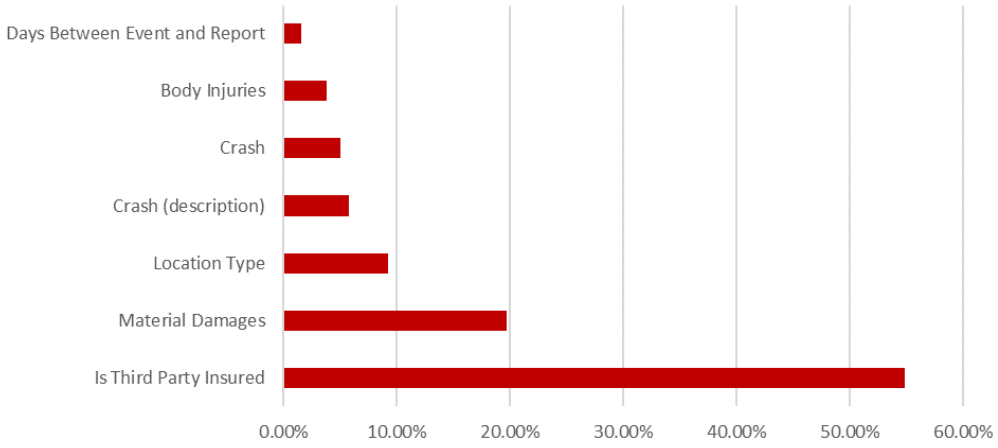


Figure 22 – Features Importance of Gradient Boosting Classifier (source: author)

6. BEST FIT

As mentioned in the previous chapter, the metrics used on the training set are essentially used to measure the overfitting degree of the model so from here onwards, only the performance metrics on the test set will be considered.

In figure 23, a resume of the performance of each model is represented. Overall, the Logistic Regression, the Neural Network and the Gradient Boost classifiers are the ones that behave better. Even though, the recall in the Neural Network is higher than in the Gradient Boost, the later was considered the best model since it performs better in general and was the one chosen to implement in the company.

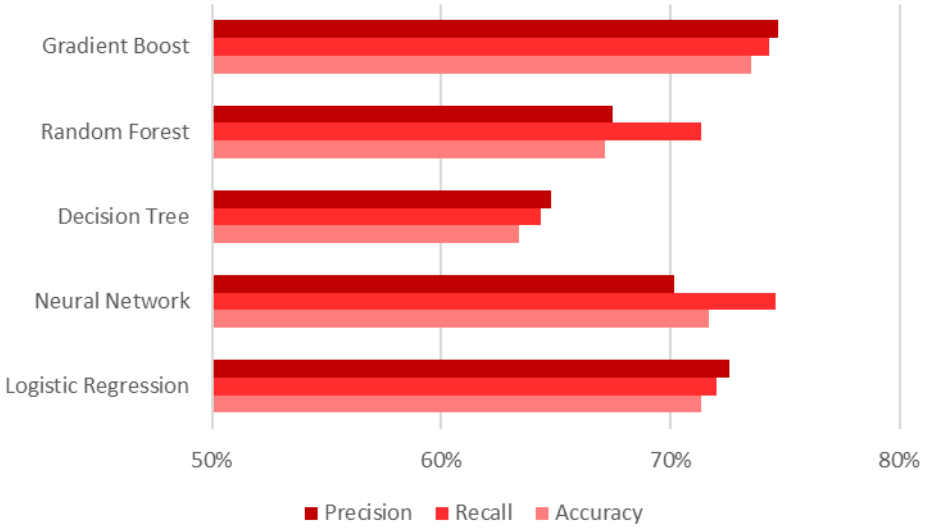


Figure 23 – Performance Metrics by Classifier (source: author)

After choosing the model, the input variables were chosen. The choice was made by looking upon the variables that were considered relevant in the previous chapter. All the variables that were important for the Decision Trees, Random Forest and Gradient Boosting classifiers were kept. Since some of those variables, like body injuries and material damages, were binary variables derived from the variable *type*, all the binary variables that came from *type* were kept. The same reasoning was made with the variables extracted from the *description of the claim*. Meaning, for example, that not just the binary variable with value '1' if the word 'crash' was present in the claim description was kept, but all the flag variables that derived from the *description of the claim* were.

6.1. BAGGING

Since the Gradient Boosting was chosen as the final classifier, there was still an attempt of improving this classifier by using an ensemble. Bagging (Bootstrap Aggregating), already mentioned before, is an ensemble meta-estimator that fits the base classifiers, each on random subsets of the original dataset and then aggregates their individual predictions with an averaging process to form a final prediction. Is a meta-algorithm that combines homogeneous weak learners in parallel, learns independently from each and combines them, typically used to reduce variance by introducing randomization into its construction procedure and then making an ensemble out of it. The goal is to produce an ensemble model that is more robust than the individual models composing it.

By bagging the Gradient Boosting classifier, all the performance metrics were slightly improved, as it is possible to see in figure 24. Hence, Bagging Gradient Boost stepped up to be the model that will be implemented.

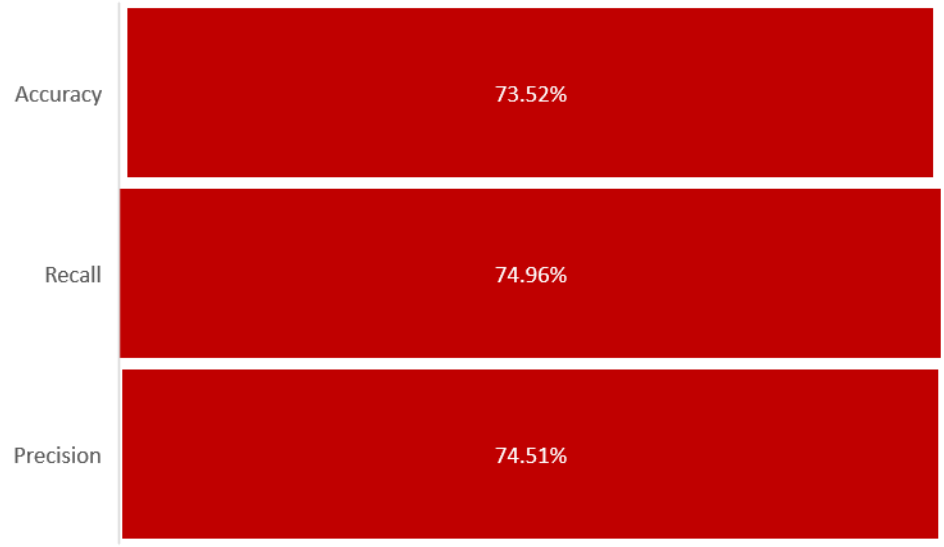


Figure 24 – Performance Metrics for Bagging Neural Network Classifier (source: author)

6.2. NEW SAMPLE

The performance measures, as the name suggests, give a numeric description of the model performance. Although, to actually realize what kind of improvements is the model giving to the company, the best model (Bagging Gradient Boost) was applied to a new and independent sample. The referred sample was composed by 2.072 investigated claim events that occurred between October 1st of 2020 and April 30st of 2021 (*reminder:* train set was composed by the investigated claim events that occurred between January 1st of 2016 and September 30th of

2020). To this new dataset no undersampling procedure was applied, the goal was for the model to retrieve a fraud probability for all the claims present in the dataset.

From the 2.072 investigated claims, 638 had the target variables with label '1', hence a fraud percentage of 30,79%. The records of the dataset were divided into five intervals: percentile 25%, percentile 50%, percentile 75%, percentile 90% and percentile of 100% as it is possible to see in *table 4*.

Percentile	Inf.	Sup.	0	1	Total	Fraud %
Percentile 25	0.0%	57.1%	412	106	518	20.5%
Percentile 50	57.1%	68.4%	368	150	518	29.0%
Percentile 75	68.4%	76.7%	363	155	518	29.9%
Percentile 90	76.7%	84.9%	185	125	310	40.3%
Percentile 100	84.9%	100.0%	106	102	208	49.0%
			1434	**638**	**2072**	**30.79%**

Table 4 – Fraud Probability Intervals (source: author)

By drawing the graph correspondent to the table presented before, present in figure 25, it is possible to see that the fraud detection rate increases in the last three intervals, being more significant in the last two of them. It is safe to say that not only the fraud detection gets increased from 30% to 49%, but the number of claims investigation can be notably decreased as well. The goal of the company is not just to detect more frauds but also spend less money in the fraud investigation. It is more efficient to investigate only the claims belonging to the last two intervals, which means that the company will go from 2.072 investigated claims to around 518, leading to a remarkable cost reduce.

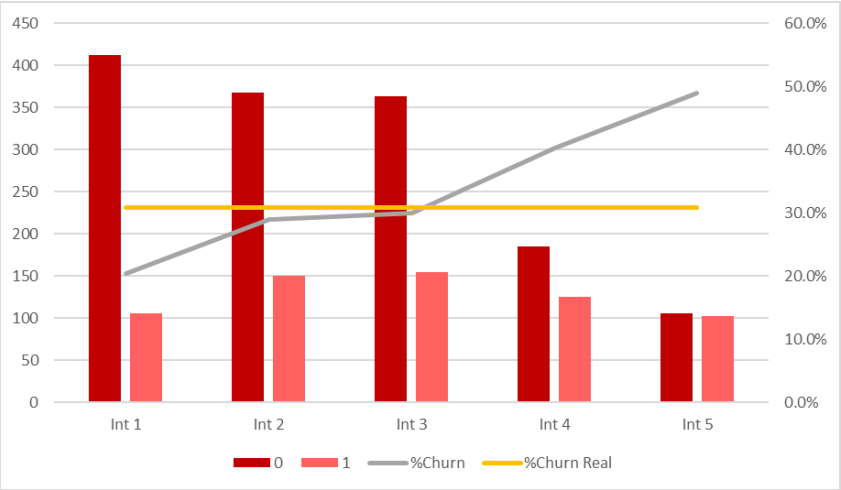


Figure 25 – Model applied to a New and Independent Sample (source: author)

With the results provided, the project was very appealing to the claims department and the necessary procedures to implement the model are now in place. Although, as it will be discussed in the next chapter, this phase of the project is not yet ready and it will take around an year to be, since *Tranquilidade* never had a model running in real time and doesn't have the necessary infrastructure in place to do it immediately.

7. IMPLEMENTATION

Until now *Tranquilidade* doesn't have the necessary infrastructure to run a model in real time, although there is already a project in place to get the company ready to implement the Fraud model and all the models that will take place in the future.

The goal is to have a technical solution for the analytics team to be able to build, deploy, test, maintain and govern analytic models made in R and in Python in an autonomous way. The objective is to create an infrastructure to develop and train models in R or Python, deploy these models on a server that will allow to add, maintain and govern them using a User Interface configuration and to have a mechanism to test the execution of the models. Also, the IT team will ensure that the analytics team is autonomous in all the previous points.

An infrastructure that will enable the recurring batch extraction, execution and delivery of the data analysis to business users will also be created. To ensure the previous described two solutions were presented: online and batch. With the online approach, presented in figure 26, the Application Programming Interface (API) will validate the input fields' format and generate a CSV file with the input fields. This file will be saved on a remote machine and run in Python. The output will be returned to the business user and the result will be saved also in a CSV file and saved in a database that will be available in PostgreSQL, as well as the input and execution time.

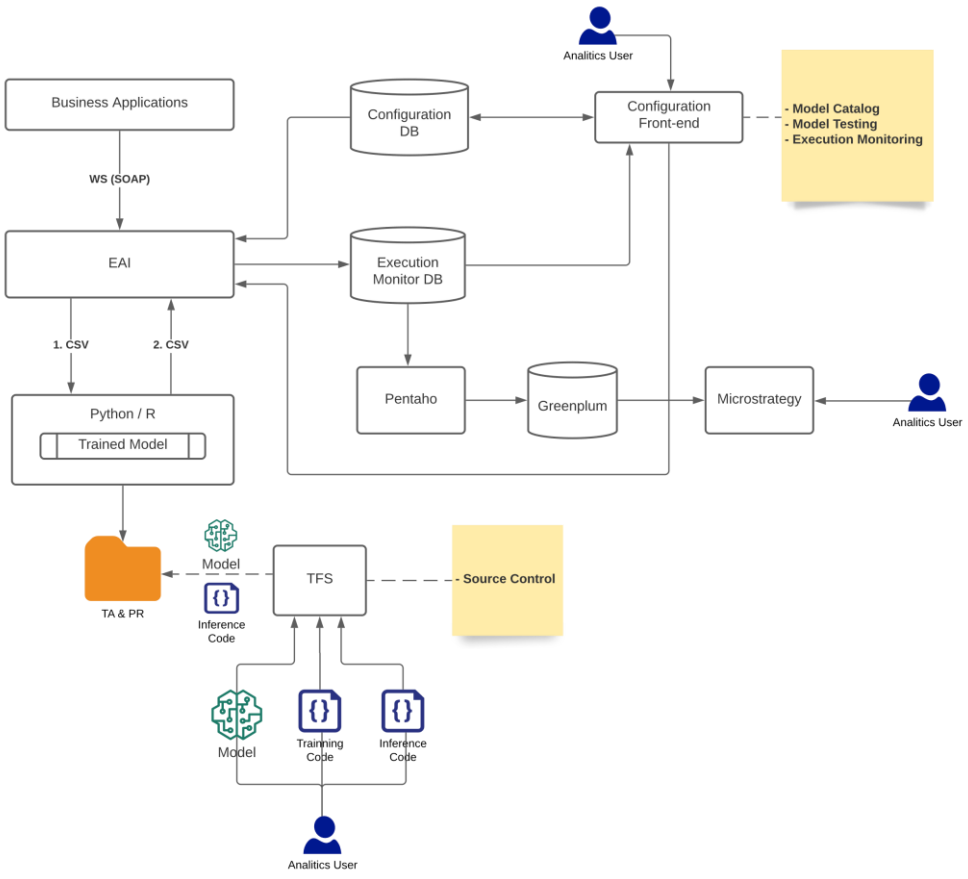


Figure 26 – Online Implementation Approach (source: Tranquilidade)

The batch proposal, referred in figure 27, lays in the business analytics platform already used in *Tranquilidade*: *MicroStrategy*. To do so, a CSV file will be delivered in a shared folder and the model will be executed every time a new file is created. The output CSV file will be generated and an Extract Transform and Load (ETL) process will provide and save the results in PostgreSQL. The business user can make a self-service report using MicroStrategy, where all the input and output information will be stored and available. It is not yet decided which of the propositions will be carried forward, costs and effort will be evaluate but there is a preference for the online approach.

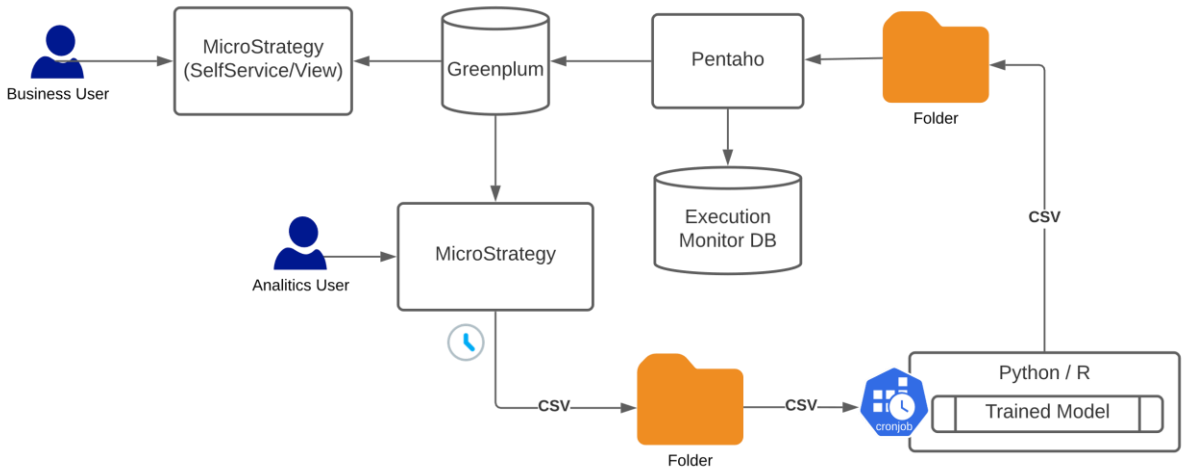


Figure 27 – Batch Implementation Approach (source: Tranquilidade)

In the meantime, the model is run ‘by hand’. Every week the Claims Department provides a file with a list of the claims that occurred during the previous one. The model is applied to this file by the analytics team and returned with an extra column in the file, where the result of the classifier is displayed. This situation will only remain until a better approach, as discussed before, is ready and implemented, in order for the process to be fully automatic and for the output to be available in real time.

8. CONCLUSION AND FUTURE WORKS

The goal of this project was fulfilled and very well received by the interested department. The claims department was pleased with the results since before the company was investigating more than 1000 claims to accomplish a 30% fraud detection rate and, after the implementation of the model, the amount of claims investigated was decreased and at the same time, it was also possible to increase the percentage of detected frauds. Now, by investigating around 500 claims, 50% of the frauds are detected. Therefore, the model is better at prioritizing the claims and is able to find more frauds which leads to less investigation costs and less unfair indemnifications paid.

Although, there is still room for improvements. In the short-term future, fraud models will be implemented for the other Lines of Business: House, Workers Compensation and Health. Since a lot of variables are similar is likely that, even though some of the input variables will differ, many will remain the same. Therefore, it will be easier to create a model for the other insurance fields now that the one for the motor insurance is constructed, reuse of experience and knowledge will be an advantage.

Still in the Automobile Line of Business, there is still one upgrade in mind. The goal will be to improve the current model to be cost-sensitive. The classifier implemented in this project aims to minimize the misclassifications and, of course that leads to a minimization of the costs in fraud investigations and in unfair compensations paid but, in terms of costs, improvements can be made. It's even safe to say that it can be better for the company to allow more misclassifications at the benefit of lower total costs.

The now implemented fraud model, assumes that all misclassifications errors carry the same cost, the cost-sensitive model would consider costs that vary by type of classification. The regular fraud model implies the costs in *table 5* and for the true positive cases, the investigation cost might be superior to the claim cost, which leads to a greater cost for the company than to simply pay the indemnification amount.

Predicted \ Real	Fraud	Legitimate
Fraud	True Positive Cost = Investigation	False Positive Cost = Investigation + Claim
Legitimate	False Negative Cost = Claim	True Negative Cost = Claim

Table 5 – Regular Model Costs (source: author)

In the future, in order that the costs are taken into consideration, the company is inclined to an interesting approach that would be to train a regular model (the one in this project) but classify each claim when making predictions according to the expected costs. In this scenario, the costs on the training set are not needed, which is what makes this improvement quite simple.

Every time a claim event occurs and enters the company's system, an initial provision is set based on the type of coverage that is activated. Which means that for each claim that opens, it is possible to know the average cost that is expected to have regarding that claim. It is also possible to study the average cost of a fraud investigation. Hence, it is only worth to investigate the claims that have an expected cost higher than the expected investigation cost. In order to apply this approach, it is only necessary to know the initial provision for each type of claim and the expected investigation cost. Afterwards, a second and final variable will be created that will have the label '1' if the model predicted the claim as fraudulent and if the expected cost is greater than the expected investigation cost, and the label '0' otherwise.

The expected cost regarding the type of claim is already set in *Tranquilidade* but the expected fraud investigation cost is not. It is still to be decided if the expected investigation cost will be a simple average of the previous costs, if it will vary according to the type of claim or if a machine learning model will also be implemented to predict it. Effort and costs of implementation will be discussed and weighted. With this approach, less frauds will be investigated and, consequently, less frauds will be detected but the main goal is to reduce the fraud related costs even if that means to detect less frauds. Although, the priority is, for now, to implement this regular model to the other insurance fields of the company.

9. BIBLIOGRAPHY

Bhandari, A. (2020). Feature Scaling for Machine Learning: Understanding the Difference Between Normalization and Standardization. *Analytics Vidhya.*

Benedek, B., László, E. (2019). Identifying Key Fraud Indicators in the Automobile Insurance Industry Using SQL Server Analysis Services. *Studia Universitatis Babes-Bolyai Oeconomica, 64* (2), 53-71.

Chen, J. (2020). Insurance Fraud. *Investopedia*.

Clarke, M. (1990). The control of insurance fraud: a comparative view. *British Journal of Criminology, 30*, 1–23.

Derrig, R. A. (2002). Insurance fraud. *The Journal of Risk and Insurance, 69*, 271–287.

Dionne, G., Giuliano, F., Picard, P. (2009). Optimal Auditing with Scoring: Theory and Application to Insurance Fraud. *Management Science, 55*, 58-70.

Fraud Detection: How Machine Learning Systems Help Reveal Scams in Fintech, Healthcare, and eCommerce. *Altexsoft*.

Galeotti, M., Rabitti, G., Vannucci, E. (2020). An Evolutionary Approach to Fraud Management. *European Journal of Operational Research*, 284, 1167-1177.

Garde, M. D. (2017). Fraud Management in Insurance Claims. *The Journal of Insurance Institute of India.*

Gepp, A., Wilson, J. H., Kumar, K. & Bhattacharya, S. (2012). A Comparative Analysis of Decision Trees Vis-a-vis Other Computational Data Mining Techniques in Automotive Insurance Fraud Detection. *Journal of Data Science*, 537-561.

Insurance Fraud Handbook (2019). *Association of Certified Fraud Examiners.*

Kovalenko, O. (2020). How to Use AI and Machine Learning in Fraud Detection. *SDP-Group.*

Maio, L. S. C. G. da C. (2013). Fraude nos Seguros: A Tolerância à Fraude no Seguro Automóvel. *Universidade do Porto.*

McCarthy, R., Ceccucci, W., McCarthy, M., Halawi, L. (2019). Alpha Insurance: A Predictive Analytics Case to Analyze Automobile Insurance Fraud using SAS Enterprise Miner. *Information Systems Education Journal*.

Molnar C. (2021). Interpretable Machine Learning – A Guide for Making Black Box Models Explainable.

Moser, R. (2019). Fraud Detection with Cost-Sensitive Machine Learning. *Towards Data Science.*

Niemi, H. (1995). Insurance fraud.

Sharma, S. (2019). Artificial Intelligence in Insurance Sector. *The Journal of Insurance Institute of India.*

Subudhi, S., Panigrahi, S. (2020). Use of Optimize Fuzzy C-Means Clustering and Supervised Classifiers for Automobile Insurance Fraud Detection. *Journal of King Saud University – Computer and Information Sciences, 32,* 568-575.

The State of Insurance Fraud Technology (2019). *Coalition Against Insurance Fraud.*

Viaene, S., Dedene, G. (2004). Insurance Fraud: Issues and Challenges. *The Geneva Papers on Risk and Insurance, 29,* 313-333.

Yan, C., Li, Y. (2015). The Identification Algorithm and Model Construction of Automobile Insurance Fraud Based on Data Mining. *Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control.*

Yan, C., Li, M., Liu, W. (2019). Improved ELM Optimimization Model for Automobile Insurance Fraud Identification based on AFSA. *Intelligent Data Analysis, 23,* 67-85.

Yan, C., Li, M., Liu, W., Qi, M. (2020). Improved Adaptive Genetic Algorithm for the Vehicle Insurance Fraud Identification Model based on a BP Neural Network. *Theoretical Computer Science, 817,* 12-23.

10.APPENDIX

10.1. CLAIM DESCRIPTION

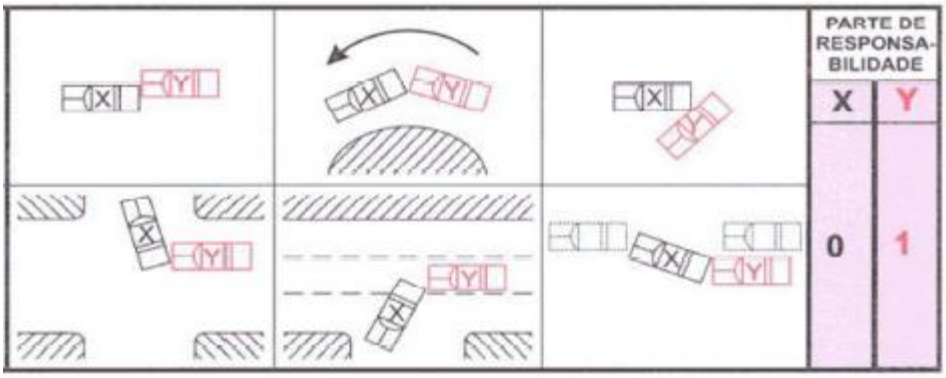


Figure 1 – Case 10 (source: *APS*)

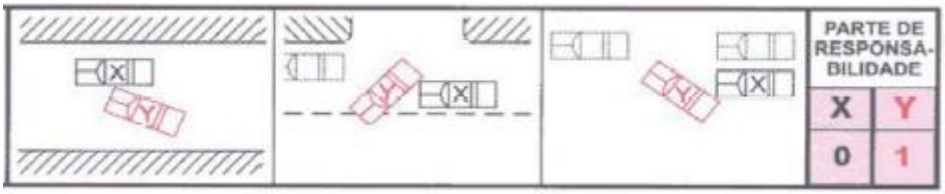


Figure 2 – Case 11 (source: *APS*)

Figure 3 – Case 12 (source: *APS*)

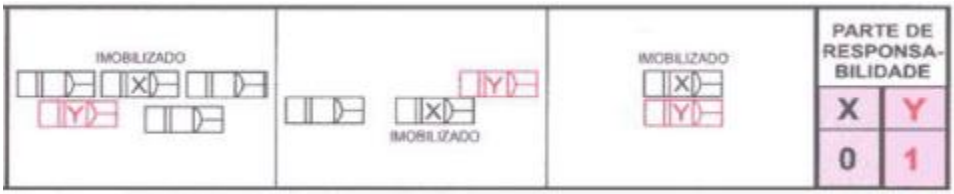


Figure 4 – Case 13 (source: *APS*)

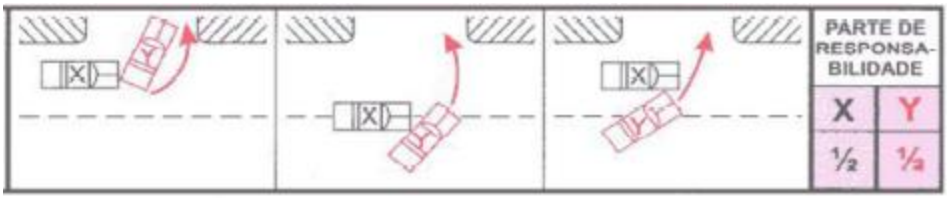


Figure 5 – Case 14 (source: *APS*)

Figure 6 – Case 15 (source: *APS*)

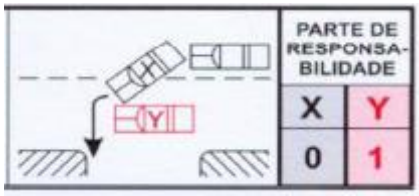


Figure 7 – Case 16 (source: *APS*)

Figure 8 – Case 17 (source: *APS*)

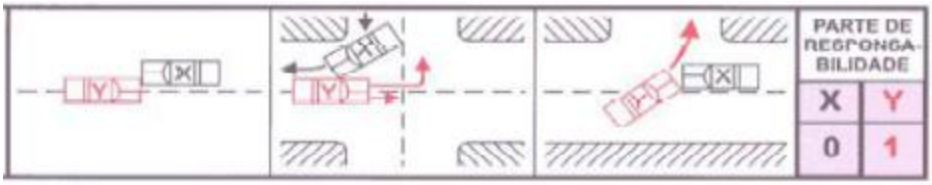


Figure 9 – Case 20 (source: *APS*)

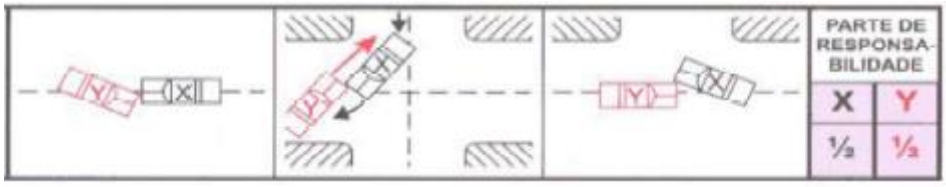


Figure 10 – Case 21 (source: *APS*)

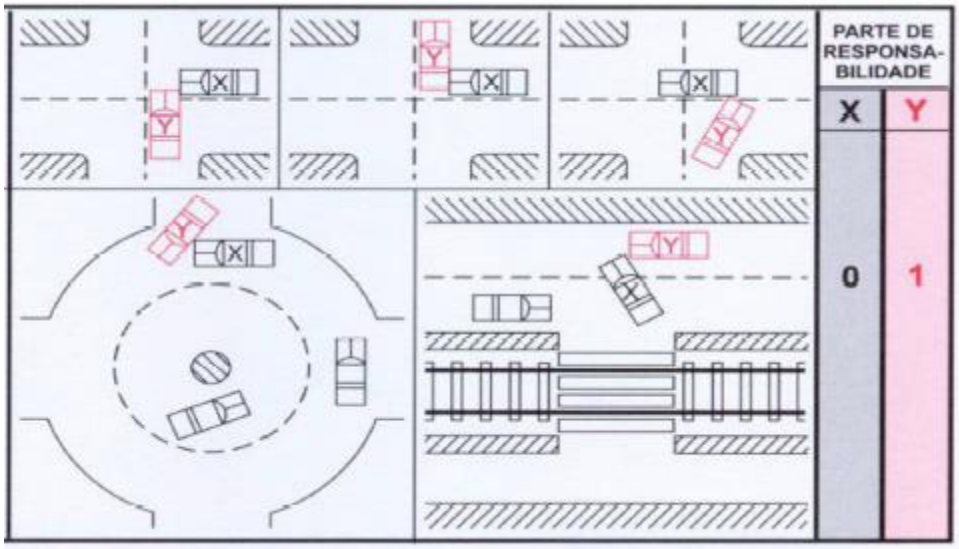


Figure 11 – Case 30 (source: *APS*)

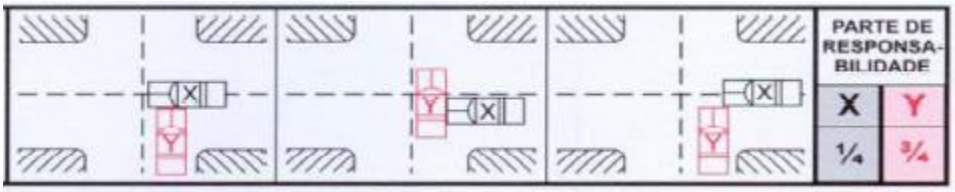


Figure 12 – Case 31 (source: *APS*)

PARTE DE RESPONSA-BILIDADE	
X	Y
0	1

Figure 13 – Case 40 (source: *APS*)

PARTE DE RESPONSA-BILIDADE	
X	Y
0	1

Figure 14 – Case 50/51/52/53 (source: *APS*)

Figure 15 – Case 60 (source: *APS*)

Figure 16 – Case 61 (source: *APS*)

Figure 17 – Case 62 (source: *APS*)

Figure 18 – Case 63 (source: *APS*)

Figure 19 – Case 64 (source: *APS*)

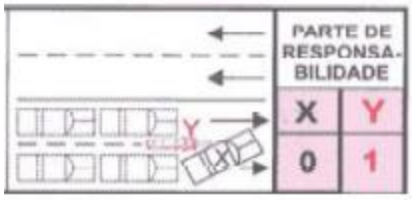


Figure 20 – Case 70 (source: *APS*)

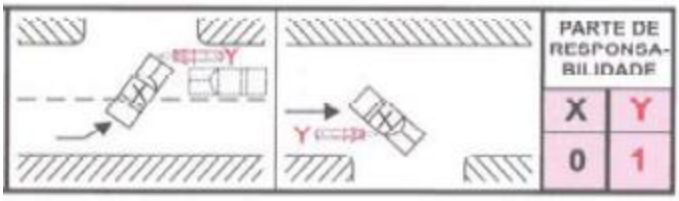


Figure 21 – Case 71 (source: *APS*)

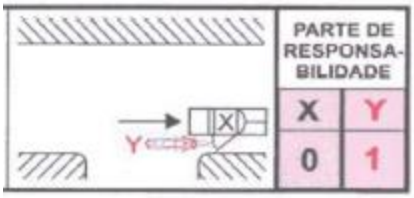


Figure 22 – Case 72 (source: *APS*)

10.2. GRAPHICAL ANALYSIS

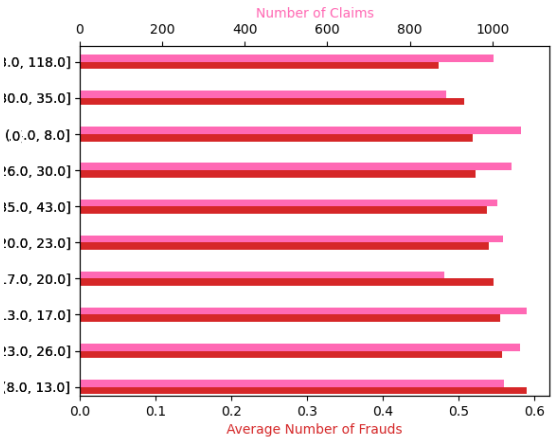


Figure 23 – Number of Claims and Average Number of Frauds by Years of Driver's License (source: author)

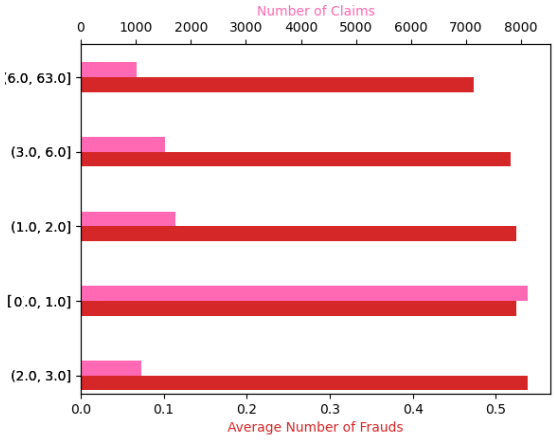


Figure 24 – Number of Claims and Average Number of Frauds by Policy Seniority (source: author)

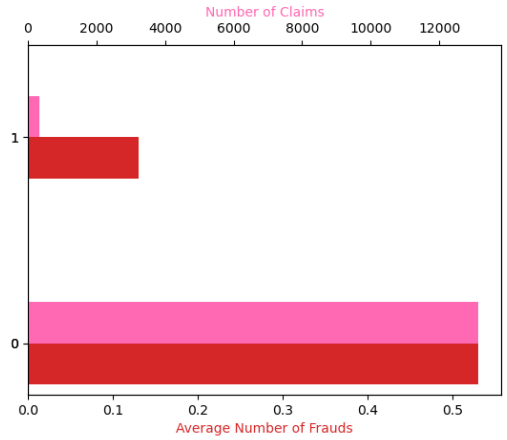


Figure 25 – Number of Claims and Average Number of Frauds by Run Over Description Flag (source: author)

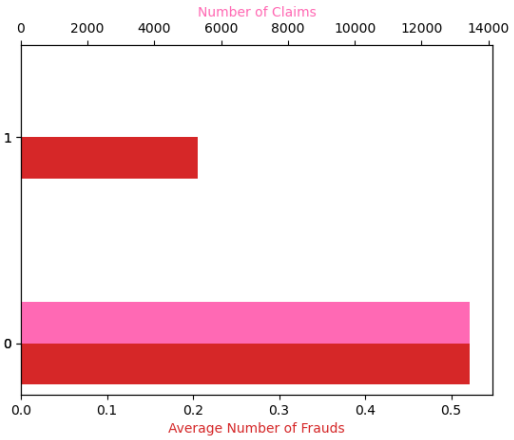


Figure 26 – Number of Claims and Average Number of Frauds by Chain Crash Description Flag (source: author)

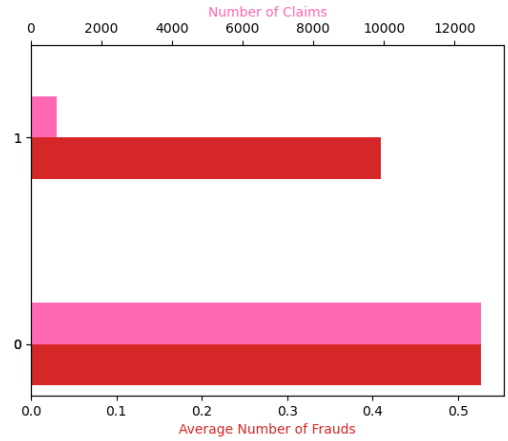


Figure 27 – Number of Claims and Average Number of Frauds by Assistance Description Flag (source: author)

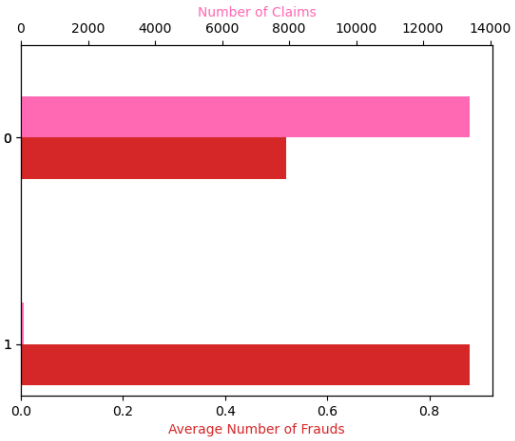


Figure 28 – Number of Claims and Average Number of Frauds by Minor Crash Description Flag (source: author)

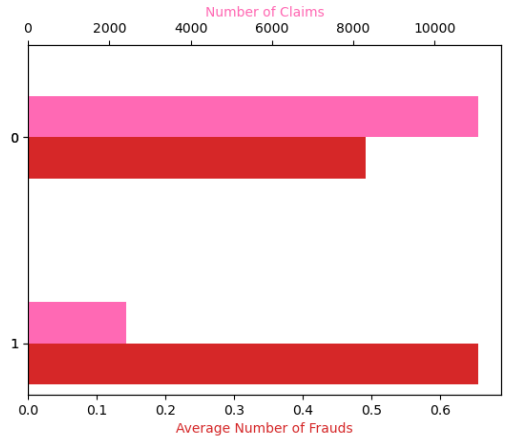


Figure 29 – Number of Claims and Average Number of Frauds by Crash Description Flag (source: author)

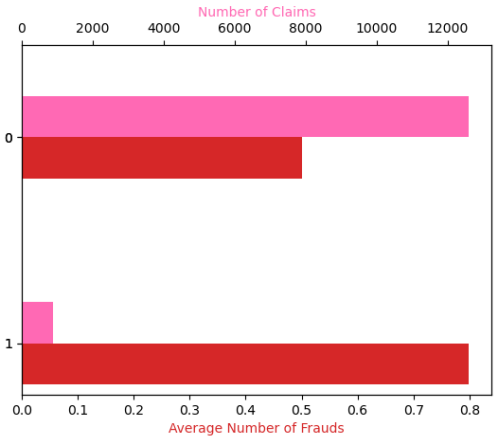


Figure 30 – Number of Claims and Average Number of Frauds by Parking Description Flag (source: author)

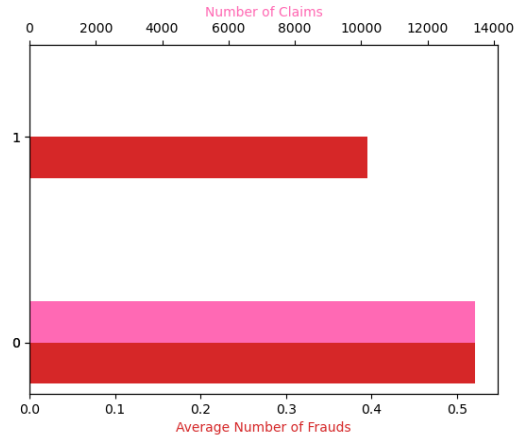


Figure 31 – Number of Claims and Average Number of Frauds by Fire Description Flag (source: author)

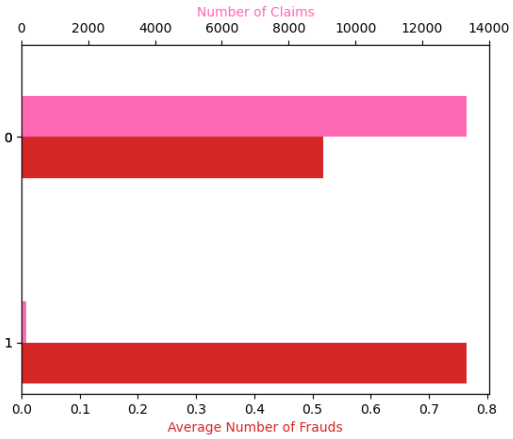


Figure 32 – Number of Claims and Average Number of Frauds by Recede Description Flag (source: author)

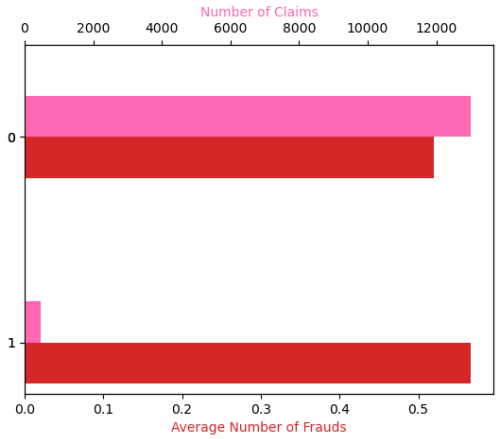


Figure 33 – Number of Claims and Average Number of Frauds by Theft Description Flag (source: author)

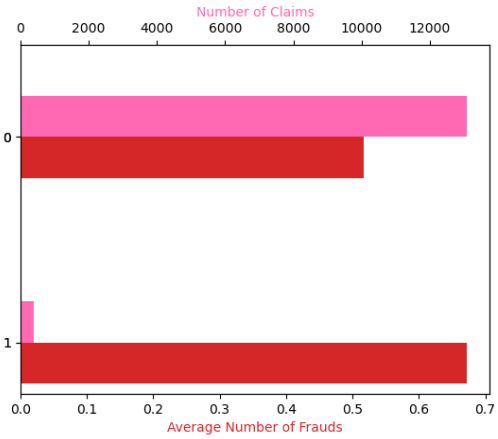


Figure 34 – Number of Claims and Average Number of Frauds by Rear Description Flag (source: author)

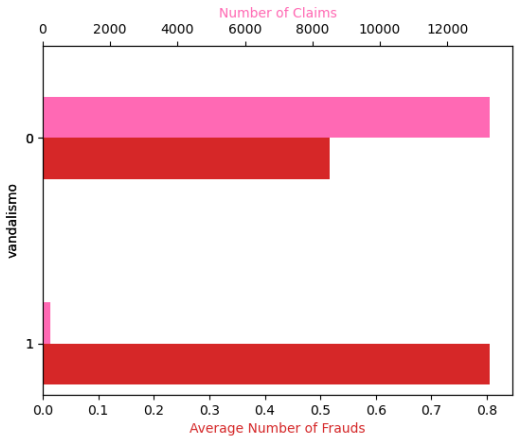


Figure 35 – Number of Claims and Average Number of Frauds by Vandalism Description Flag (source: author)

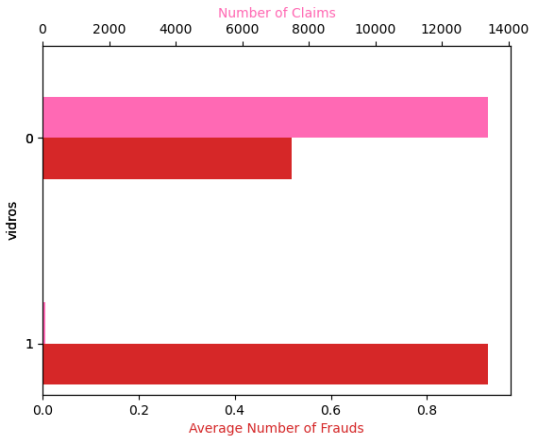


Figure 36 – Number of Claims and Average Number of Frauds by Windows Description Flag (source: author)

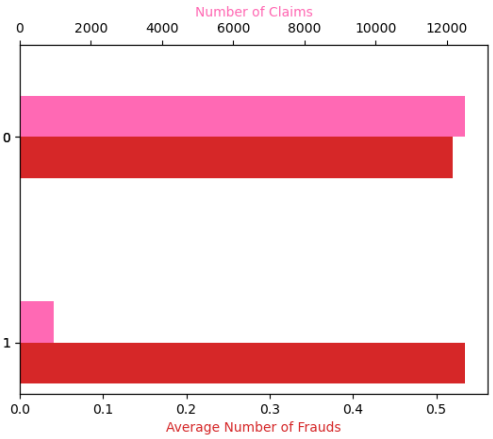


Figure 37 – Number of Claims and Average Number of Frauds by Case 10 Description Flag (source: author)

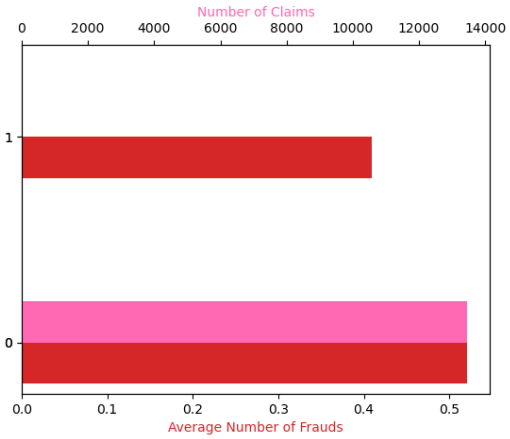


Figure 38 – Number of Claims and Average Number of Frauds by Case 12 Description Flag (source: author)

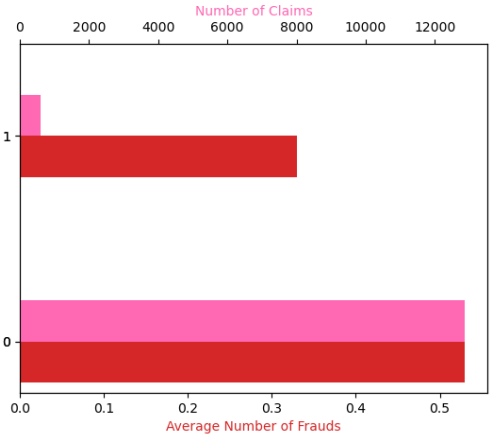


Figure 39 – Number of Claims and Average Number of Frauds by Case 11 Description Flag (source: author)

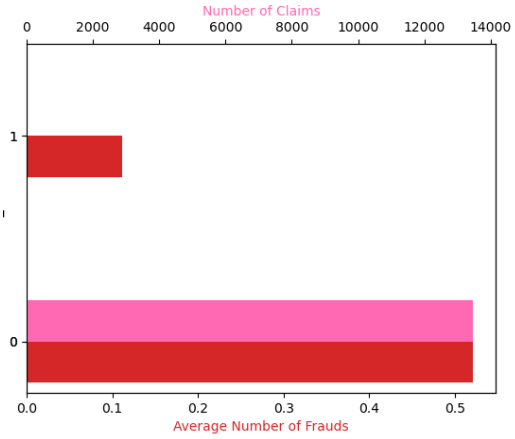


Figure 40 – Number of Claims and Average Number of Frauds by Case 13 Description Flag (source: author)

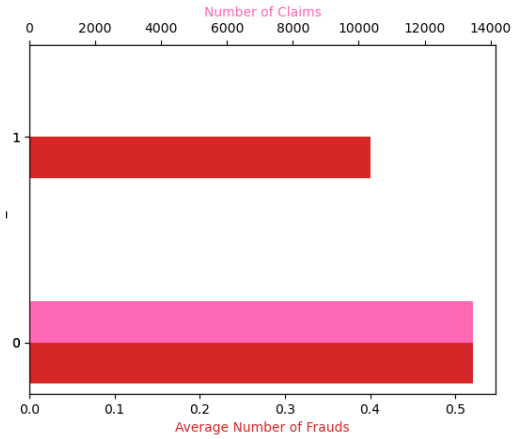


Figure 41 – Number of Claims and Average Number of Frauds by Case 14 Description Flag (source: author)

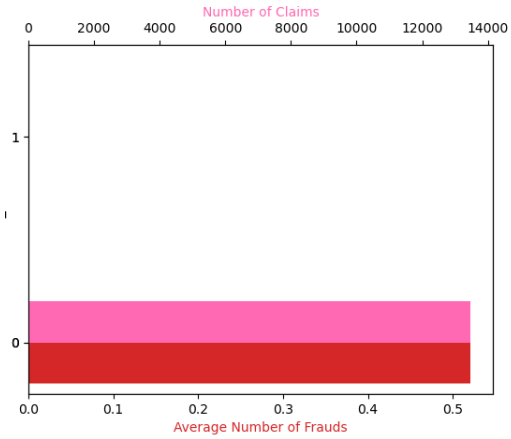


Figure 42 – Number of Claims and Average Number of Frauds by Case 16 Description Flag (source: author)

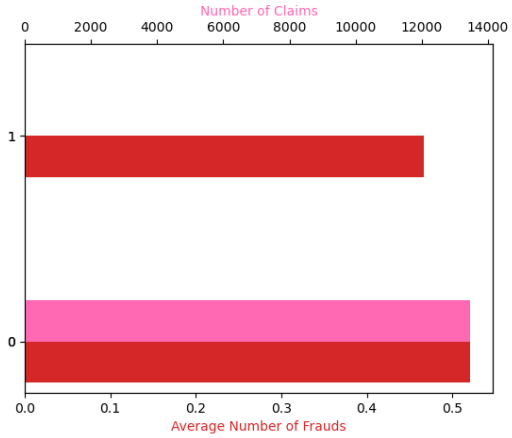


Figure 43 – Number of Claims and Average Number of Frauds by Case 15 Description Flag (source: author)

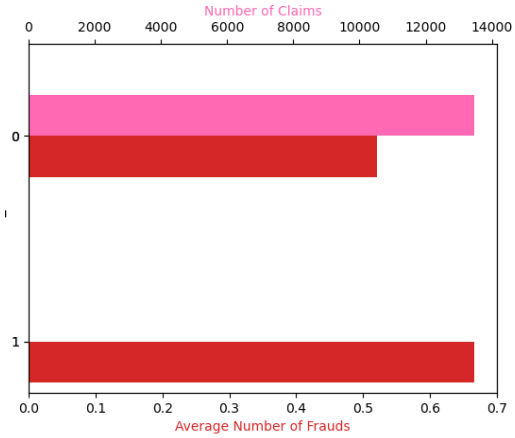


Figure 44 – Number of Claims and Average Number of Frauds by Case 17 Description Flag (source: author)

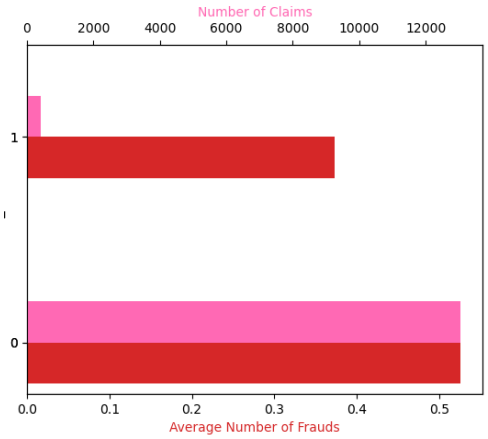


Figure 45 – Number of Claims and Average Number of Frauds by Case 20 Description Flag (source: author)

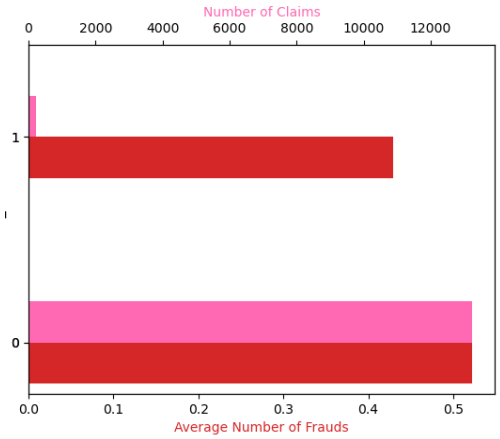


Figure 46 – Number of Claims and Average Number of Frauds by Case 30 Description Flag (source: author)

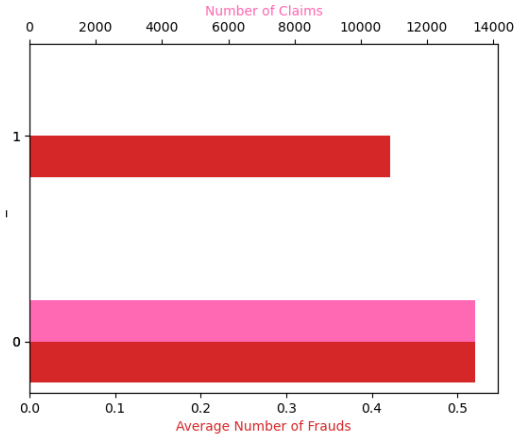


Figure 47 – Number of Claims and Average Number of Frauds by Case 21 Description Flag (source: author)

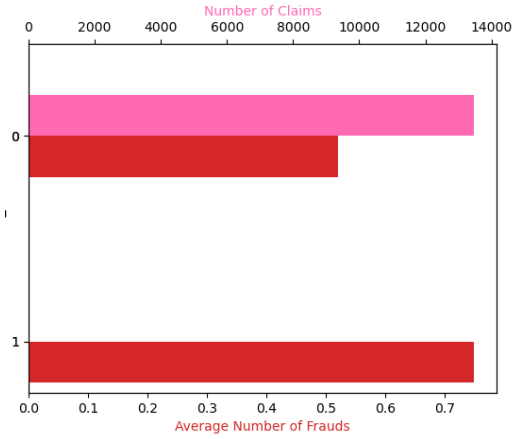


Figure 48 – Number of Claims and Average Number of Frauds by Case 31 Description Flag (source: author)

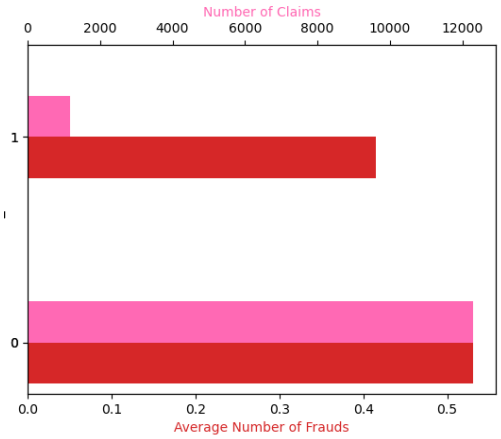


Figure 49 – Number of Claims and Average Number of Frauds by Case 40 Description Flag (source: author)

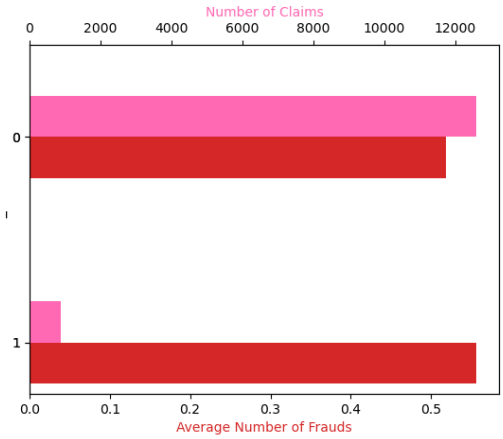


Figure 50 – Number of Claims and Average Number of Frauds by Case 51 Description Flag (source: author)

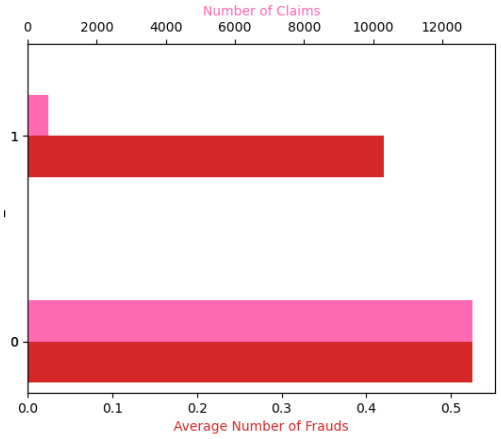


Figure 51 – Number of Claims and Average Number of Frauds by Case 50 Description Flag (source: author)

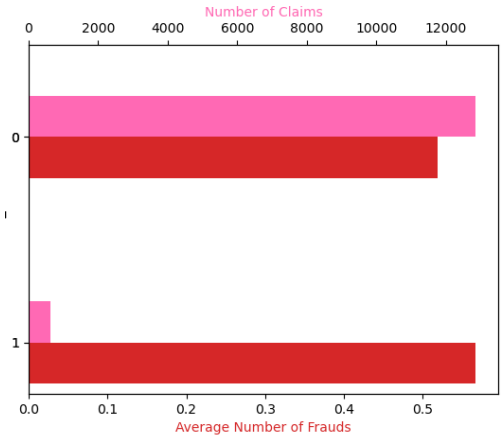


Figure 52 – Number of Claims and Average Number of Frauds by Case 52 Description Flag (source: author)

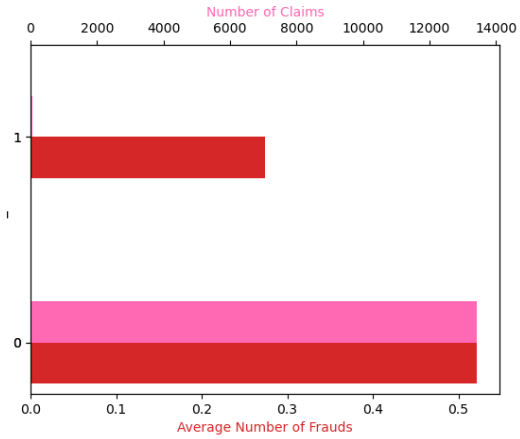


Figure 53 – Number of Claims and Average Number of Frauds by Case 53 Description Flag (source: author)

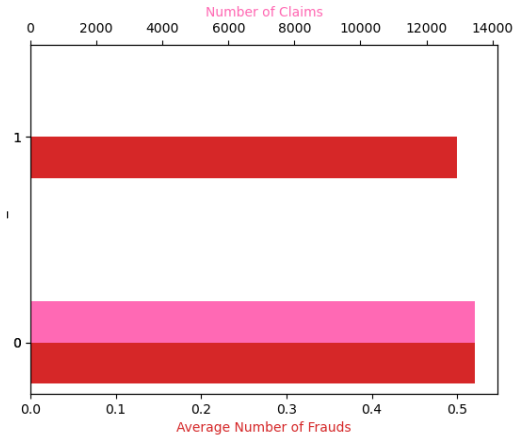


Figure 54 – Number of Claims and Average Number of Frauds by Case 61 Description Flag (source: author)

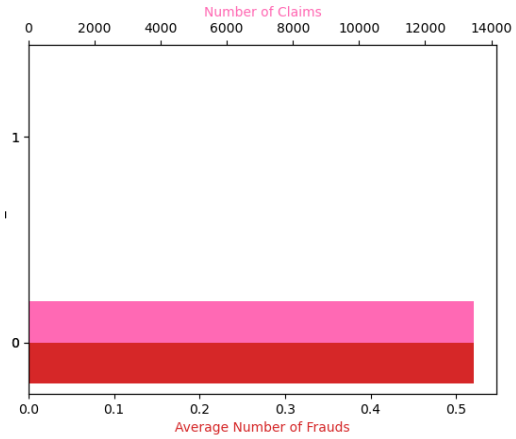


Figure 55 – Number of Claims and Average Number of Frauds by Case 60 Description Flag (source: author)

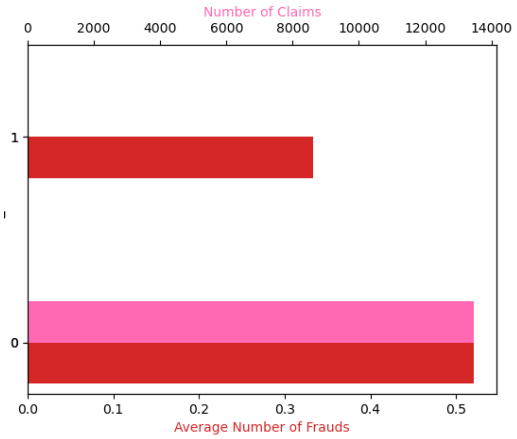


Figure 56 – Number of Claims and Average Number of Frauds by Case 62 Description Flag (source: author)

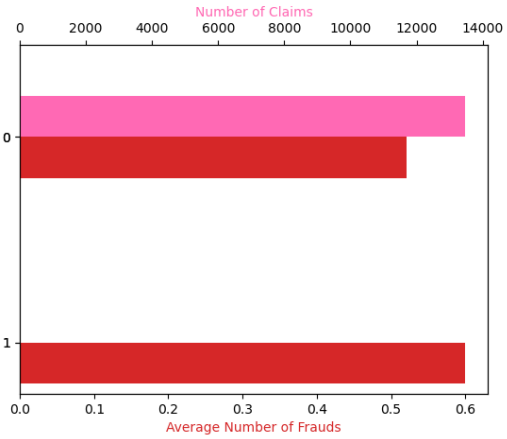


Figure 57 – Number of Claims and Average Number of Frauds by Case 63 Description Flag (source: author)

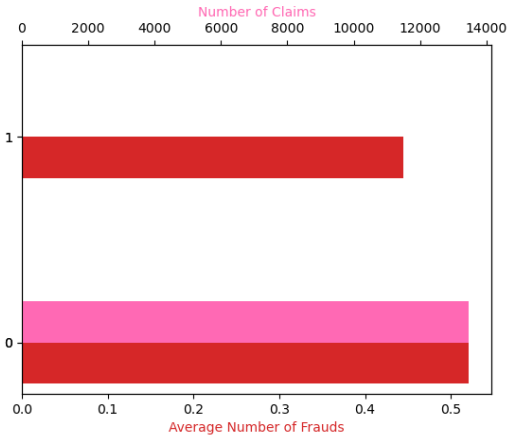


Figure 58 – Number of Claims and Average Number of Frauds by Case 70 Description Flag (source: author)

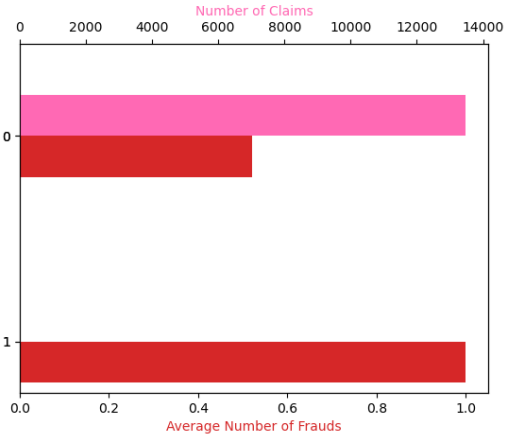


Figure 59 – Number of Claims and Average Number of Frauds by Case 64 Description Flag (source: author)

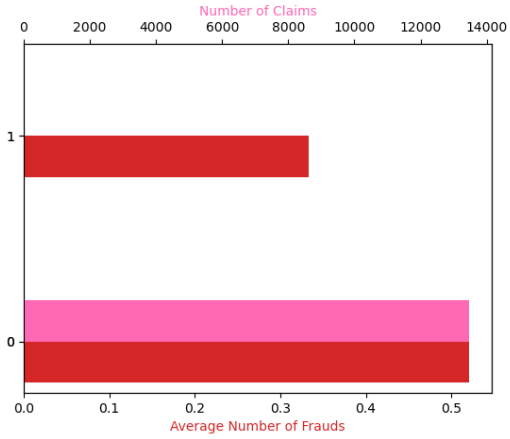


Figure 60 – Number of Claims and Average Number of Frauds by Case 71 Description Flag (source: author)

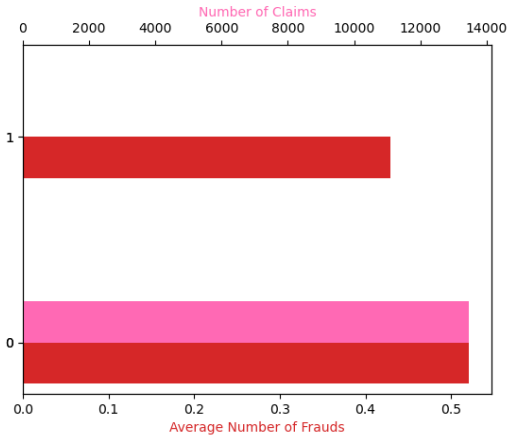


Figure 61 – Number of Claims and Average Number of Frauds by Case 72 Description Flag (source: author)

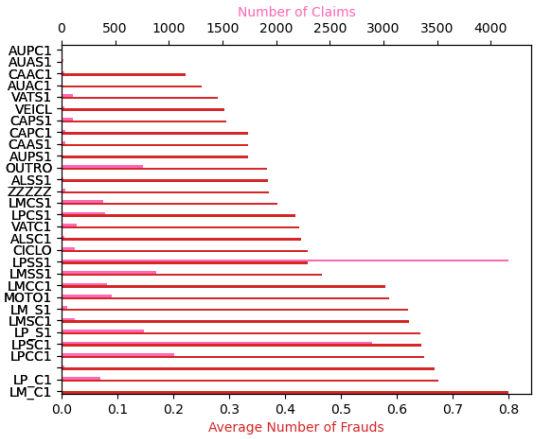


Figure 62 – Number of Claims and Average Number of Frauds by Vehicle Category Level 2 (source: author)

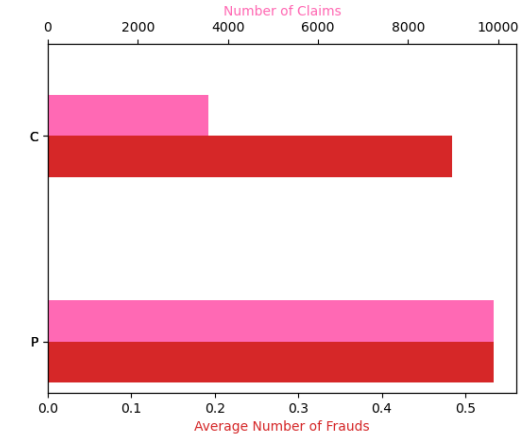


Figure 63 – Number of Claims and Average Number of Frauds by Client Type (source: author)

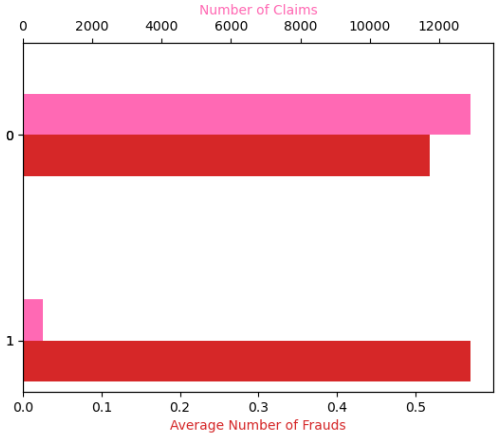


Figure 64 – Number of Claims and Average Number of Frauds by Driver Dif Client Flag (source: author)

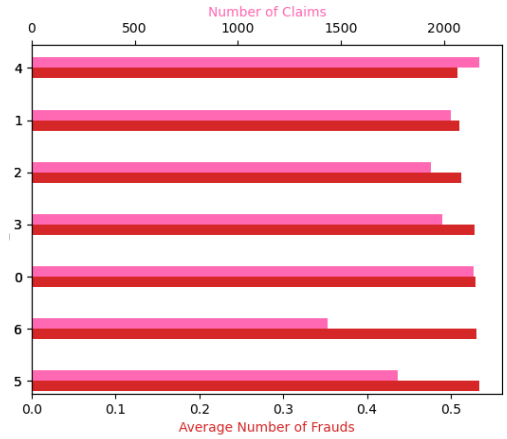


Figure 65 – Number of Claims and Average Number of Frauds by Day of the Week of Claim (source: author)

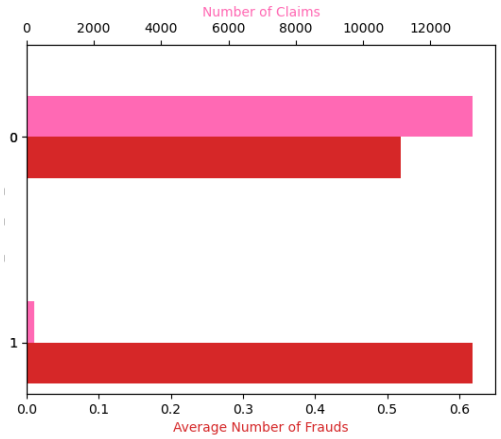


Figure 66 – Number of Claims and Average Number of Frauds by Change in Contract Year Flag (source: author)

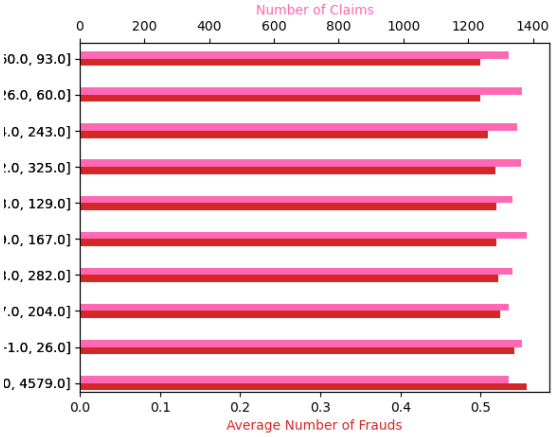


Figure 67 – Number of Claims and Average Number of Frauds by Number of Days Without Claims (source: author)

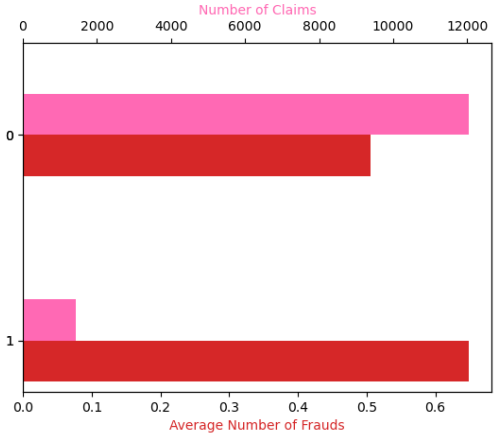


Figure 68 – Number of Claims and Average Number of Frauds by Change in Contract Flag (source: author)

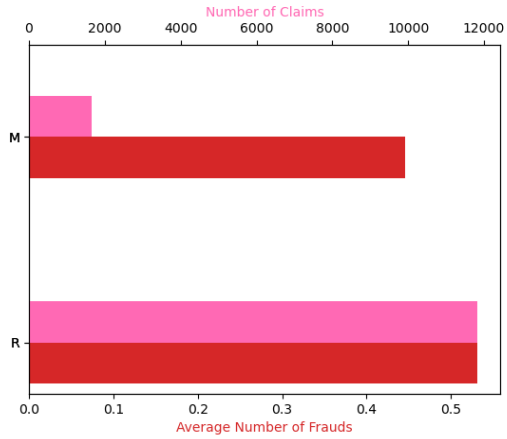


Figure 69 – Number of Claims and Average Number of Frauds by Business Type (source: author)

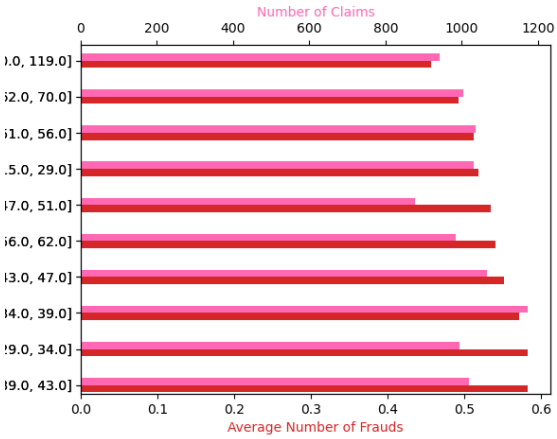


Figure 70 – Number of Claims and Average Number of Frauds by Age of the Client (source: author)

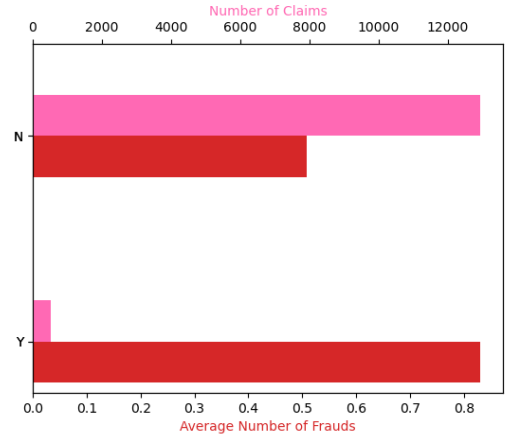


Figure 71 – Number of Claims and Average Number of Frauds by Participant Fraud (source: author)

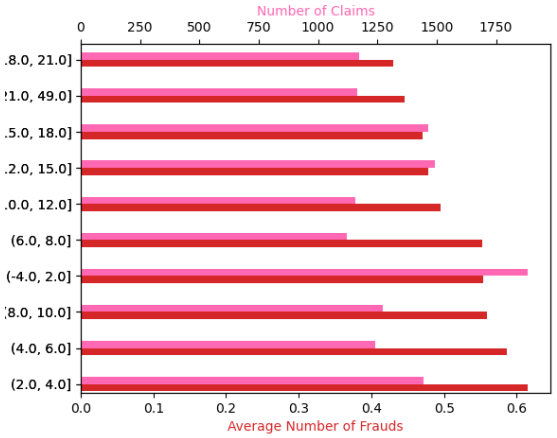


Figure 72 – Number of Claims and Average Number of Frauds by Vehicle's Age (source: author)

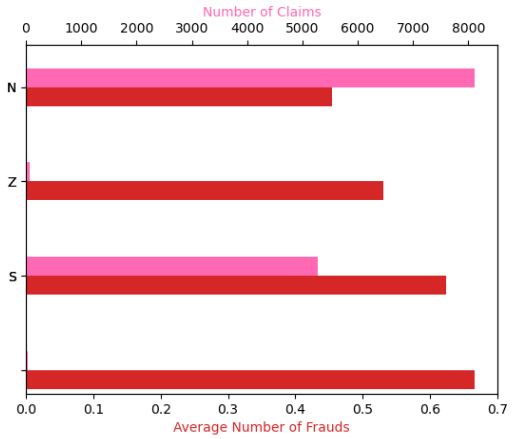


Figure 73 – Number of Claims and Average Number of Frauds by Own Damage Insurance (source: author)

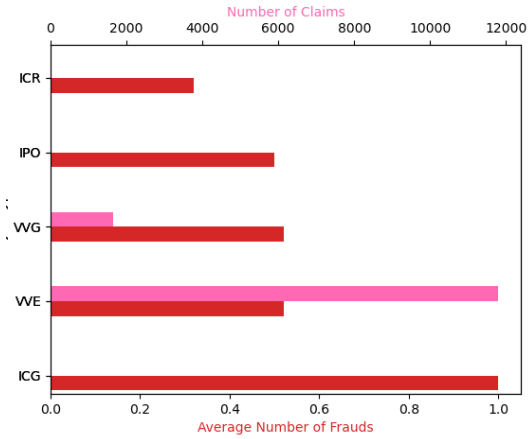


Figure 74 – Number of Claims and Average Number of Frauds by Object Type (source: author)

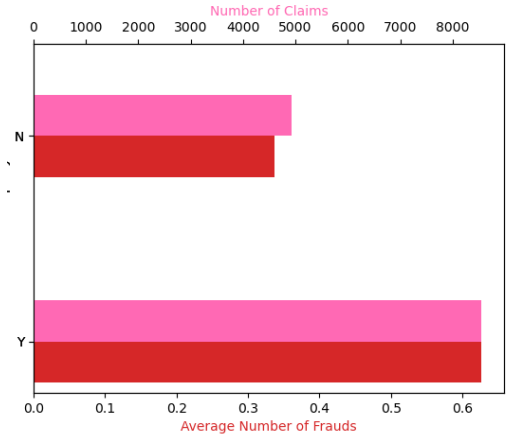


Figure 75 – Number of Claims and Average Number of Frauds by Is Third Party Insured (source: author)

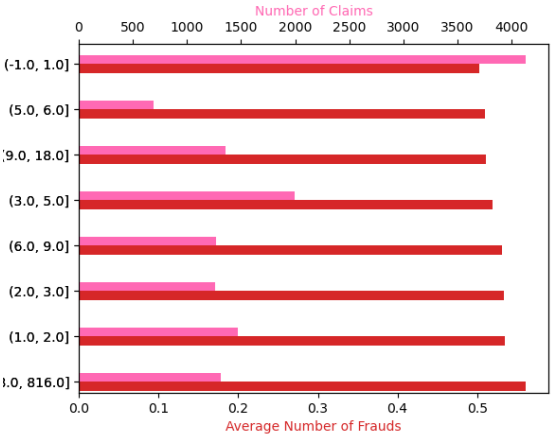


Figure 76 – Number of Claims and Average Number of Frauds by Number of Days to Report a Claim (source: author)

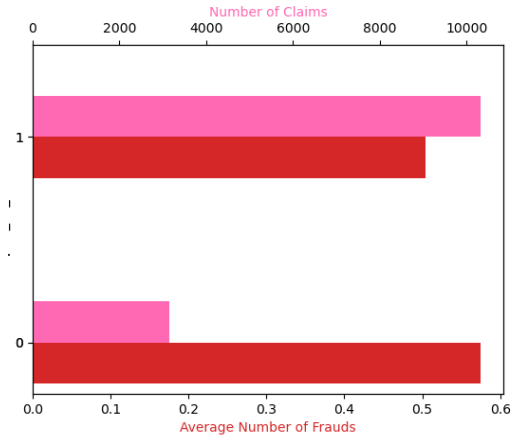


Figure 77 – Number of Claims and Average Number of Frauds by Participant Dif Client Flag (source: author)

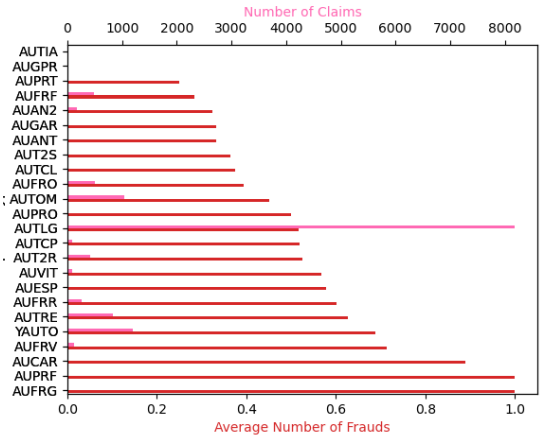


Figure 78 – Number of Claims and Average Number of Frauds by Product (source: author)

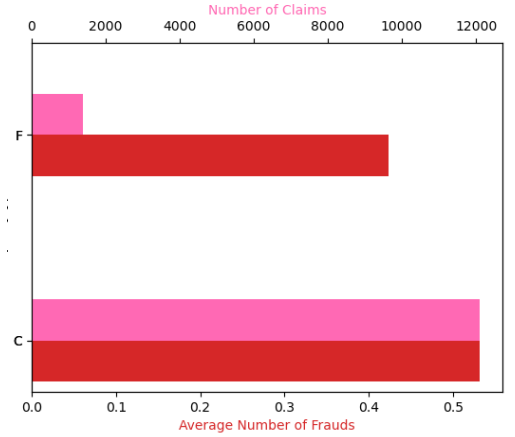


Figure 79 – Number of Claims and Average Number of Frauds by Policy Type (source: author)

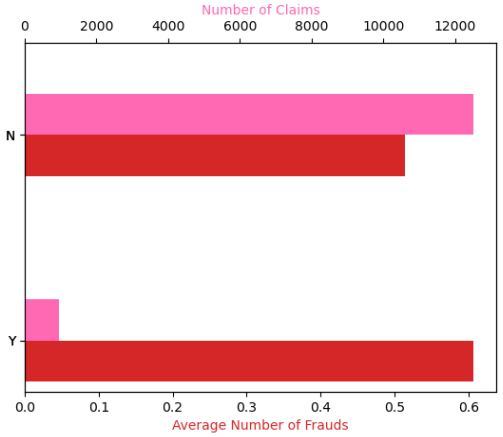


Figure 80 – Number of Claims and Average Number of Frauds by Property Finance Type (source: author)

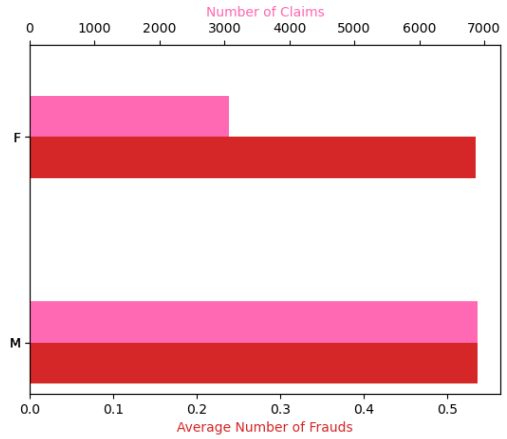


Figure 81 – Number of Claims and Average Number of Frauds by Sex of the Client (source: author)

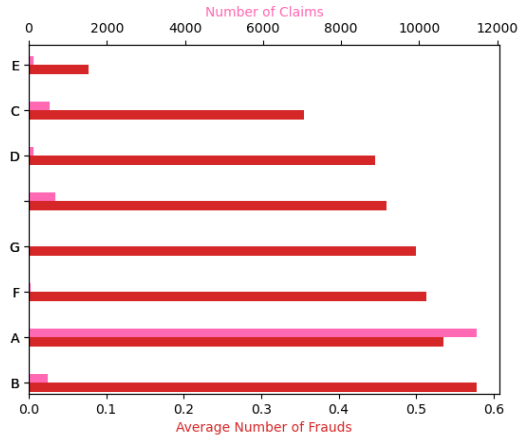


Figure 82 – Number of Claims and Average Number of Frauds by Vehicle Type (source: author)

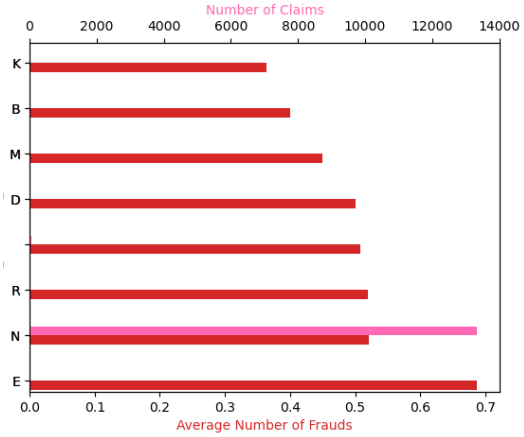


Figure 83 – Number of Claims and Average Number of Frauds by License Plate Type (source: author)

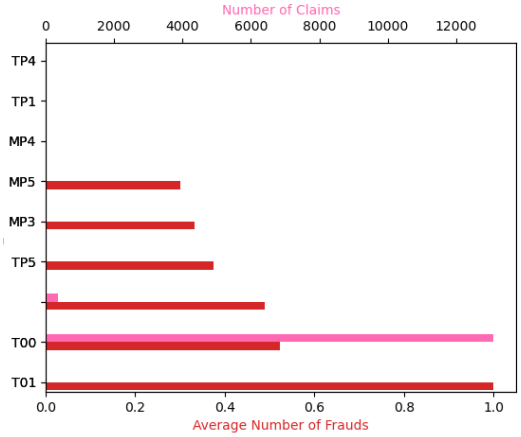


Figure 84 – Number of Claims and Average Number of Frauds by Transportation Type (source: author)

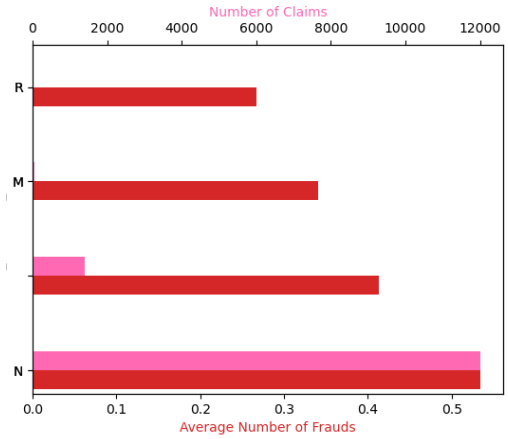


Figure 85 – Number of Claims and Average Number of Frauds by Vehicle's Use Type (source: author)

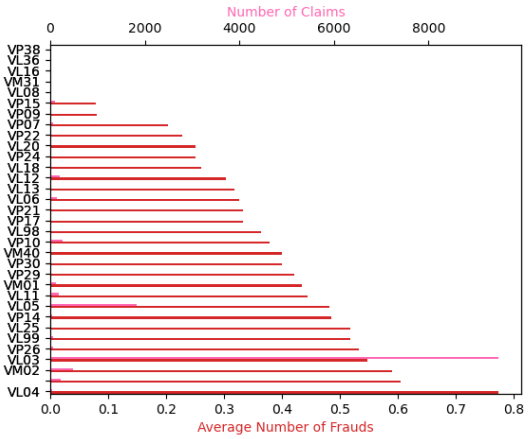


Figure 86 – Number of Claims and Average Number of Frauds by Vehicle Category (source: author)

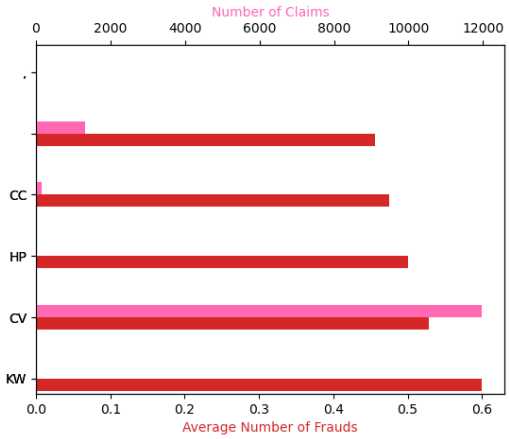


Figure 87 – Number of Claims and Average Number of Frauds by Vehicle's Power (source: author)

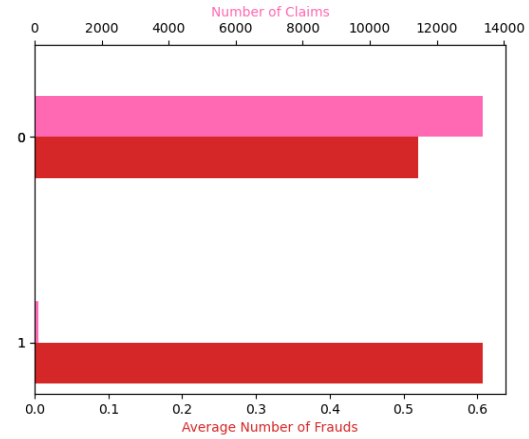


Figure 88 – Number of Claims and Average Number of Frauds by Replacement Vehicle Activated Flag (source: author)

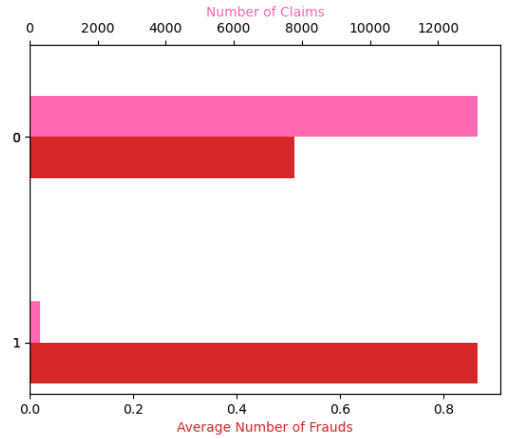


Figure 89 – Number of Claims and Average Number of Frauds by Vandalism Activated Flag (source: author)

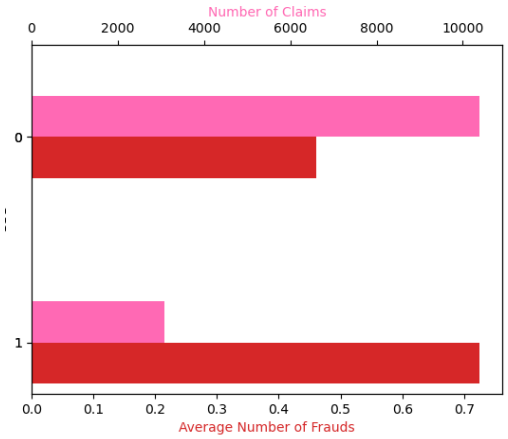


Figure 90 – Number of Claims and Average Number of Frauds by Crash Activated Flag (source: author)

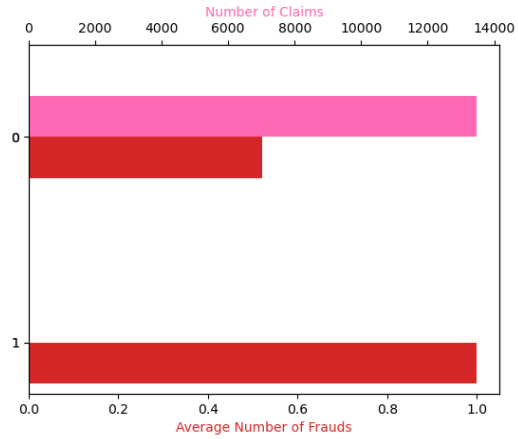


Figure 91 – Number of Claims and Average Number of Frauds by Natural Catastrophes Activated Flag (source: author)

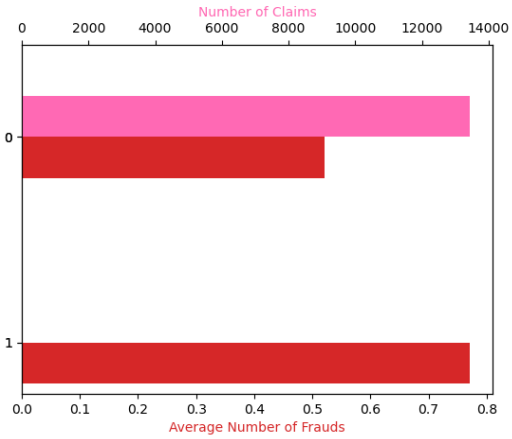


Figure 92 – Number of Claims and Average Number of Frauds by Earthquakes Activated Flag (source: author)

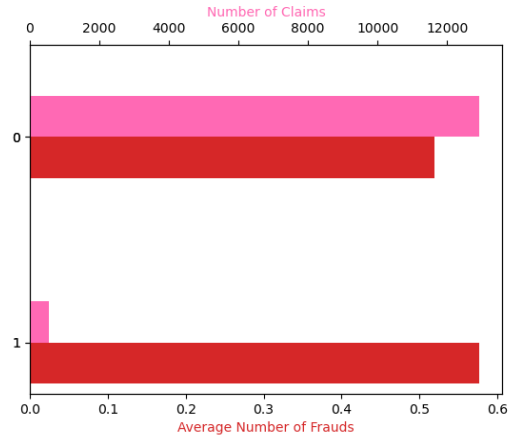


Figure 93 – Number of Claims and Average Number of Frauds by Theft Activated Flag (source: author)

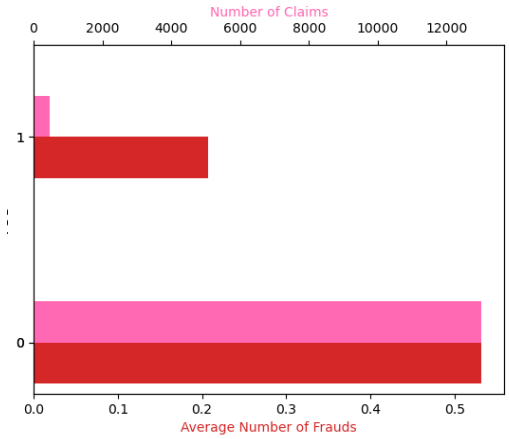


Figure 94 – Number of Claims and Average Number of Frauds by Other Ocupants' Protection Activated Flag (source: author)

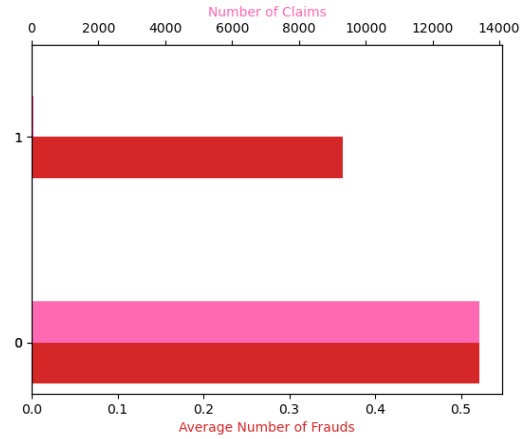


Figure 95 – Number of Claims and Average Number of Frauds by Fire/Explosion Activated Flag (source: author)

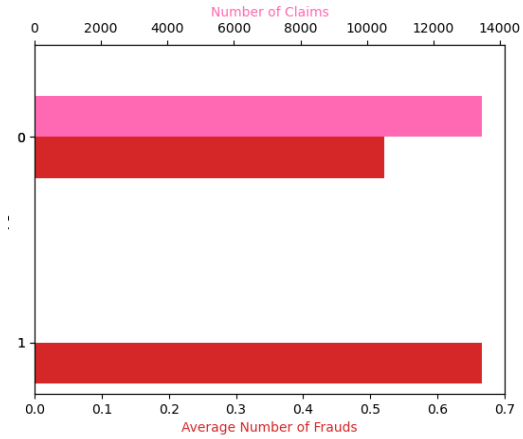


Figure 96 – Number of Claims and Average Number of Frauds by Legal Protection Activated Flag (source: author)

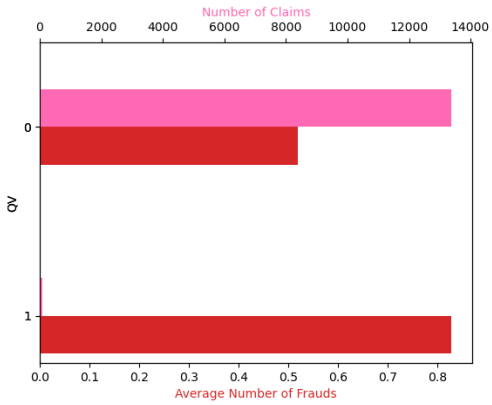


Figure 97 – Number of Claims and Average Number of Frauds by Glass Breakage Activated Flag (source: author)

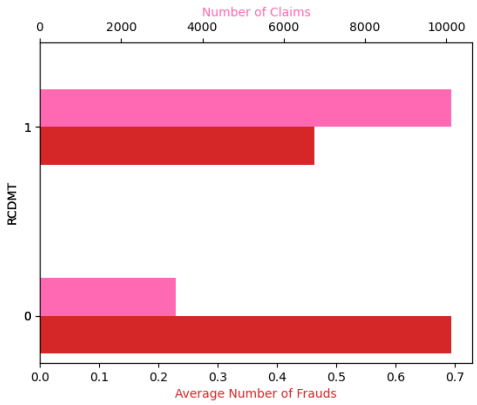


Figure 98 – Number of Claims and Average Number of Frauds by Material Damage Activated Flag (source: author)

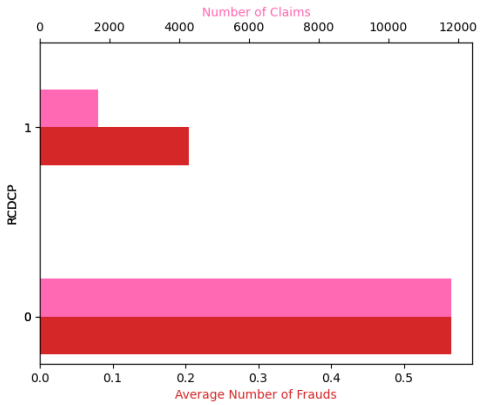


Figure 99 – Number of Claims and Average Number of Frauds by Body Injury Activated Flag (source: author)

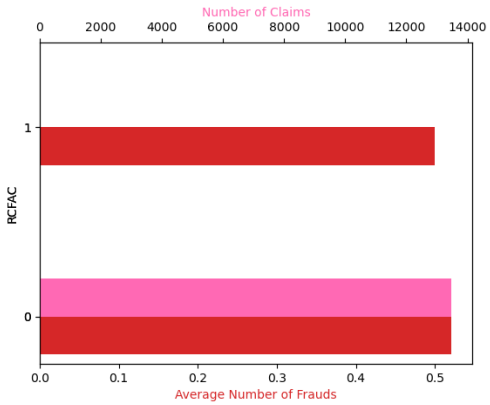


Figure 100 – Number of Claims and Average Number of Frauds by Extra Activated Flag (source: author)

10.3. CORRELATION

Correlation	Target Variable
Policy Type	-0.0703
Product	0.1169
Property Finance Type	0.0529
Number of Claims	0.0287
Location Type	0.0267
Is Third Party Insured	0.2822
Participant Fraud	0.1224
Client Type	0.0449
Business Type	0.0623
Object Type	0.0119
Vehicle Category	-0.0969
Statistical Code	-0.1226
Driver's License Type	-0.0891
License Plate Type	-0.0021
Vehicle Type	-0.1013
Transportation Type	0.0079
Vehicle Use Type	0.0048
Vehicle's Power	0.0104
Vehicle Category Level 2	0.0203
Own Damage Insurance	0.1679
Target Variable	1.0000
Crash Coverage Activated Flag	0.2214
Material Damage Coverage Activated Flag	-0.2042
Body Injury Activated Flag	-0.2385
Theft Coverage Activated Flag	0.0251
Glass Breakage Coverage Activated Flag	0.0605
Other Ocupants' Protection Coverage Activated Flag	-0.1203
Vandalism Coverage Activated Flag	0.1176
Replacement Vehicle Coverage Activated Flag	0.0235
Fire/Explosion Coverage Activated Flag	-0.0058
Legal Protection Coverage Activated Flag	0.0230
Earthquakes Coverage Activated Flag	0.0167
Extra Coverage Activated Flag	0.0081
Natural Catastrophes Coverage Activated Flag	0.0081
Change in Contract Flag	0.0982
Change in Contract Year Flag	0.0285
Number of Vehicles Involved	-0.2030
Participant Dif Client	-0.0669
Driver Dif Client	0.0200
Number of Days to Report a Claim	0.0143
Policy Seniority	-0.0249
Number of Days Without Claims	0.0196
Day of the Week of Claim	0.0012
Vehicle's Age	-0.1187
Number of Frauds of the Policy	0.0393
Number of Frauds of the Agent	0.0855
Number of Frauds of the Client	0.0444
Number of Frauds of the Vehicle	-0.0025
Case 10 Description Flag	0.0118
Case 11 Description Flag	-0.0891
Case 12 Description Flag	-0.0116
Case 13 Description Flag	-0.0188
Case 14 Description Flag	0.0115
Case 15 Description Flag	-0.0159
Case 16 Description Flag	-0.0152
Case 17 Description Flag	-0.0006
Case 20 Description Flag	-0.0635
Case 21 Description Flag	-0.0090
Case 30 Description Flag	-0.0189
Case 31 Description Flag	0.0031
Case 40 Description Flag	-0.0778
Case 50 Description Flag	-0.0326
Case 51 Description Flag	0.0155
Case 52 Description Flag	0.0085
Case 53 Description Flag	-0.0300
Case 60 Description Flag	-0.0088
Case 61 Description Flag	0.0115
Case 62 Description Flag	-0.0005
Case 63 Description Flag	0.0031
Case 64 Description Flag	-0.0006
Case 70 Description Flag	0.0061
Case 71 Description Flag	-0.0091
Case 72 Description Flag	-0.0094
Parking Description Flag	0.1526
Minor Crash Description Flag	0.0586
Chain Crash Description Flag	-0.0253
Rear Description Flag	0.0515
Crash Description Flag	0.1309
Windows Description Flag	0.0688
Assistance Description Flag	-0.0581
Recede Description Flag	0.0408
Vandalism Description Flag	0.0861
Fire Description Flag	-0.0090
Theft Description Flag	0.0157
Run Over Description Flag	-0.1343
Others Description Flag	-0.0381
Mark	0.0205
County	0.0359

Figure 101 – Correlation (source: author)