INTELLIGENT RECOMMENDER SYSTEM FOR CAR INSURANCE PLANS

Recommending coverage packages with the focus on flexibility and personalization

TAMÁS PÓTSA

Work project carried out under the supervision of:

Professor Rongjiao Ji

16/12/2022

GROUP PART

Acknowledgments: To professor Rongjiao Ji for the incessant support provided throughout this thesis and to José Vieira and Gonçalo Roque for all the help, availability, and valuable input.

Abstract: Acknowledging the success of personalized recommendations as support to promote sales within a business, this paper proposes the development of a recommender system to answer Fidelidade's problem of depersonalization in the auto insurance sector. To build a model able to consider historical data from the customer and the car to recommend the best auto insurance package, a thorough data cleaning and model hypertunning were made to ensure that the three main objectives: predictability (accuracy when predicting), explainability (explaining to each customer the reason to recommend a certain product) and flexibility (proposing different coverages combinations) were satisfied.

Keywords: Recommendation System; Business Analytics; Data Science; Car Insurance; Data-driven decision making.

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

GROUP PART

GROUP PART

1. Introduction

1.1. Company background

Fidelidade Group is a multinational insurance company that operates in 12 countries through 5 companies: Fidelidade, Multicare, Fidelidade Assistência, Via Directa, and Companhia Portuguesa de Reinsurance.

According to the company's 2021 Management Report, Grupo Fidelidade has 29.1% of the market share, revenues of 4.9 billion euros in 2021 (with revenues coming from international markets accounting for more than 1 billion euros) and a customer base greater than 8.3 million (Portugal representing 27%). Additionally, the company is the market leader in Portugal, Bolivia, and Cape Verde. (Fidelidade 2021)

Regarding its products, one can say that they are an ecosystem of integrated insurance services in the areas of mobility, health, home, savings, and elderly care. All these services are delivered to consumers through the company's robust and diverse distribution channels (online and offline), which are a key component of its omnichannel strategy, which entails utilizing several channels in an aligned and complementary manner to facilitate the customer's purchase journey. Among all available services, auto insurance was the one selected as the focus of this work project.

1.2. The auto insurance

Auto insurance products have demonstrated, over the previous few years, the relevancy of their existence in the insurance companies' portfolios. This is a consequence of car insurance being mandatory by law for the European Union (EU) members. The number of potential clients is directly related to the number of existing cars in Portugal. In 2020, there were 541 cars for one thousand people resulting in more than 1 car per family (Expresso 2022), and in 2021, more than 5 million passenger cars were in circulation (Pordata 2021), resulting in revenues of 1.8 billion euros for the auto insurance market (Económico 2022). The same applies to Fidelidade: in 2021,

more than 33% of the non-life revenue came from the automobile side (Fidelidade 2021). Since car users in Portugal are increasing over the years, Fidelidade should work towards capturing these new potential clients. In fact, given the importance this segment has in the company's activity, Fidelidade needs to be constantly improving and differentiating itself from its competitors.

An analysis of the 18 largest Portuguese auto insurance brands was conducted by the Consumer Guidance Institute Portugal (CGIP). This study took into consideration four different pillars to evaluate the auto insurance offer: (1) the number of different offers, (2) the price, (3) the website including the simulator, and (4) the customer service. This analysis reveals that Fidelidade stands out in all its offers (barebone, medium, and premium offers), and is included in the top three positions of the ranking for the criteria (1), (2), and (4). However, when it comes to criteria number (3), the company is not included in the top 3 rankings, showing space for additional improvement (CGIP 2021).

1.3. Problem

As stated previously, Fidelidade shows a disadvantage regarding the website's ability to simulate auto insurance packages. In fact, when buying auto insurance online, the information required for the customer to get a recommendation is very limited, being only necessary to fill in the customer's age, claim's historical data, and the car's plate. With this information, the customer obtains three out of the five pre-made insurance packages: auto 1, auto 2, auto 3, auto 4, and autoestima. In addition, he has the possibility, depending on the insurance that was recommended, to add some extra coverages (as seen in Table 17 of the appendix).

Below is the detailed list of auto insurance plans with a description of their respective protection needs:

- **Auto 1** - For those who are exclusively concerned with complying with the law and choose to buy online, motivated only by the price.
- **Auto 2** - For those who, in addition to legal compliance, prefer personal protection, both for the driver and its occupants.
- **Auto 3** - For those looking for protection from unforeseen events, unrelated to their behavior, in damage to the vehicle.
- **Auto 4** - For those who value extensive protection, with full vehicle damage guarantees and a high level of service.
- **AUTOESTIMA** - Innovative solution for those who want insurance for their damage and have a car up to 20 years old.

The absence of a detailed and personalized recommendation simulator is more and more impacting Fidelidade's capacity to keep up with its insurgent new competitors. In fact, during the past few years, digital and highly personalized features have been given considerable focus when developing insurance products. Indeed, some insurtechs are already entering the market with 100% digital offers and hyper-personalized products. This is the case of Metromile, an auto insurance company selling pay-per-mile products, that is charging the customer in accordance with the amount of use he gives the car. Big players such as Fidelidade need to improve and develop some already existing tools to compete against these new entries.

1.4. Solution

This work project intends to support Fidelidade in its strategy: to find assertive sales channels to provide solutions that are tailored to each client's unique needs (Fidelidade 2021) by creating a machine learning recommender system, helping Fidelidade to promote a better customer experience and being able to meet customers' needs.

Several models were built and tested to attain the best possible results, content-based models (models where historical data is considered and where it is possible to explain to the

client the reason to recommend such insurance package) were submitted to some hypertunning and the most important features were retrieved. This type of model was used to recommend the five current insurance packages available in Fidelidade, but it was also used to recommend some new insurance plans more personalized. (These models will be referred to in more detail in the result section). A new technique was also used, a mix between clusters and collaborative filtering (a model that uses ratings already given by users to recommend items to other users), to try to recommend the right insurance package to the customer (this technique is described in more detail in the results section).

Three objectives were pre-defined to measure the success of this work project:

i) Predictability measures the accuracy of the recommendations made, this means predicting well the insurance package that best fits the customer, this metric will be monitored using a confusion matrix that is explained in detail in the Models section.

ii) Flexibility represents how well the model can recommend more personalized insurance packages to the customer, instead of having five predefined insurance packages more packages with extra coverages were created.

iii) Explainability, this latter represents the difficulty level of explaining to the client the reason he was recommended with such insurance.

The coming sections will describe in more detail each step followed to achieve the best possible model, never forgetting the three pre-defined goals. In the data chapter, data will be explored and thoroughly cleaned to be fed, later, into the model. This chapter will be followed by the models section, where a theoretical explanation of machine learning recommender models and a summary of the evaluation metrics that will be used to measure the models' performance will be presented. The results chapter will expose three individual work projects, each one of these latter is focused on one of the three project objectives previously mentioned, and, in the

end, to conclude there will be a chapter where the work project results are summarized, and some limitations and next steps will also be presented.

2. Data

2.1. Exploratory data analysis

Throughout the project, a thorough analysis was performed to better understand the data and some potential relationship between this latter and insurance packages recommended to the client. Starting with an exploratory data analysis (generally used to analyze and investigate data sets and summarize their main characteristics of it, employing data visualization methods (IBM 2020) resulting in some visualizations and some interesting findings.

2.1.1. Data description

Before cleaning the database there was information about 248046 client-car pairs and 105 features. It is necessary to refer to the fact that there is a one-to-many relationship between customers and cars meaning that there are some clients with more than one car, but there are no cars belonging to more than one customer.

It is also important to refer to the fact that from 105 features, 104 are referring to historical data either about the client or the car and 1 feature is the insurance package bought by each client. This latter is going to be the target variable used for every model.

The goal of our project is to recommend insurance packages to customers based on data about themselves and their cars. This means that the variable that stores the type of insurance package a customer chooses for his or her car is the one where its value depends on the changes of all the other variables. That makes “DSC_OPCAO_COB” the target variable, and all the other features (except any feature that stores a unique ID) independent variables.

To have a better understanding of how the independent variables affect the target variable an Exploratory Data Analysis was performed. This latter will start by looking at the big picture followed by a more detailed analysis. Understanding the data is essential for conducting

meaningful and in-depth research. Before cleaning the data, there were 55 categorical (or nominal) variables and 50 numerical (or interval). A categorical variable has two or more categories, but there is no intrinsic ordering to them, while a numerical variable is one where there is a clear ordering between the values, and the intervals between them are equally spaced (UCLA 2021). Moreover, more than 3M missing values were found in the database, being necessary to deal with them before feeding the model. The missing values cleaning section will be discussed in more detail in the Data cleaning chapter.

2.1.2. Data Analysis

The following section will conduct a detailed analysis of the dataset, possible relations between them, and patterns hidden, that can help better understand further model results, will be shown. This analysis will be divided into three different sections, one related to the target variable, another related to the customer analysis, and the last section related to the car analysis.

2.1.2.1. Target variable

As above mentioned, the target variable, the insurance packages bought by each customer, must be given special prominence, being the first variable to be analysed. This variable is composed of five different categories (Auto 1, Auto 2, Auto 3, Auto 4, and Autoestima) seen in more detail in Table 17. By plotting the number of customers that bought each product, the following graph was obtained.

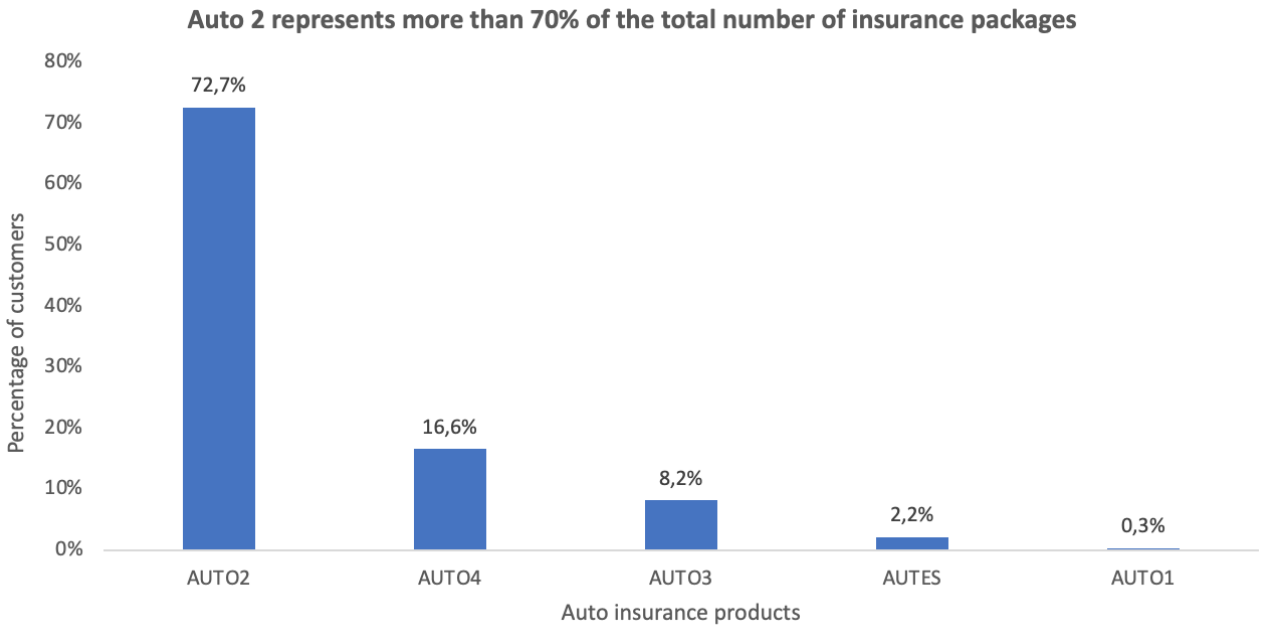


Figure 1 - Percentage of customers that bought each auto product. The x-axis represents the five different auto insurance packages, the y-axis represents the percentage of customers.

As seen in **Error! Reference source not found.** the target variable does not follow a uniform distribution, this is consequence of having an unbalanced database. Most of the customers bought Auto 2 packages (around 72,7%), contrasting with 0.3% of the customers that bought Auto 1. When dealing with such cases it is necessary to take some measures to prevent the model from wrongly predicting the target variable. A trade-off between personalization and prediction power is identifiable. To find the best ratio for it some hyper-tunning was done in the modeling phase.

2.1.2.2. Customer analysis

In the following section, a detailed analysis will be shown, it is important to have in mind that all the variables that will be presented were the ones that showed, during the EDA, to be more interesting even if in the modelling phase they do not appear as being the ones with highest importance score for the model.

Typical exploratory data analysis is divided into numerical and categorical variables

analysis but, since the main purpose of this chapter is to identify and understand the data, this section was divided into customer and car analysis, both categories including numerical and categorical variables.

Starting by understanding the typical auto customer, the variable 'QTD_IDADE' proved to be of great importance, its values are dispersed between 8 and 113, being the mean equal to 49 years old. In Figure 2, it is possible to observe that the data distribution is similar to a normal distribution being that more than 50% of the customers on the database are aged between 36 and 65.

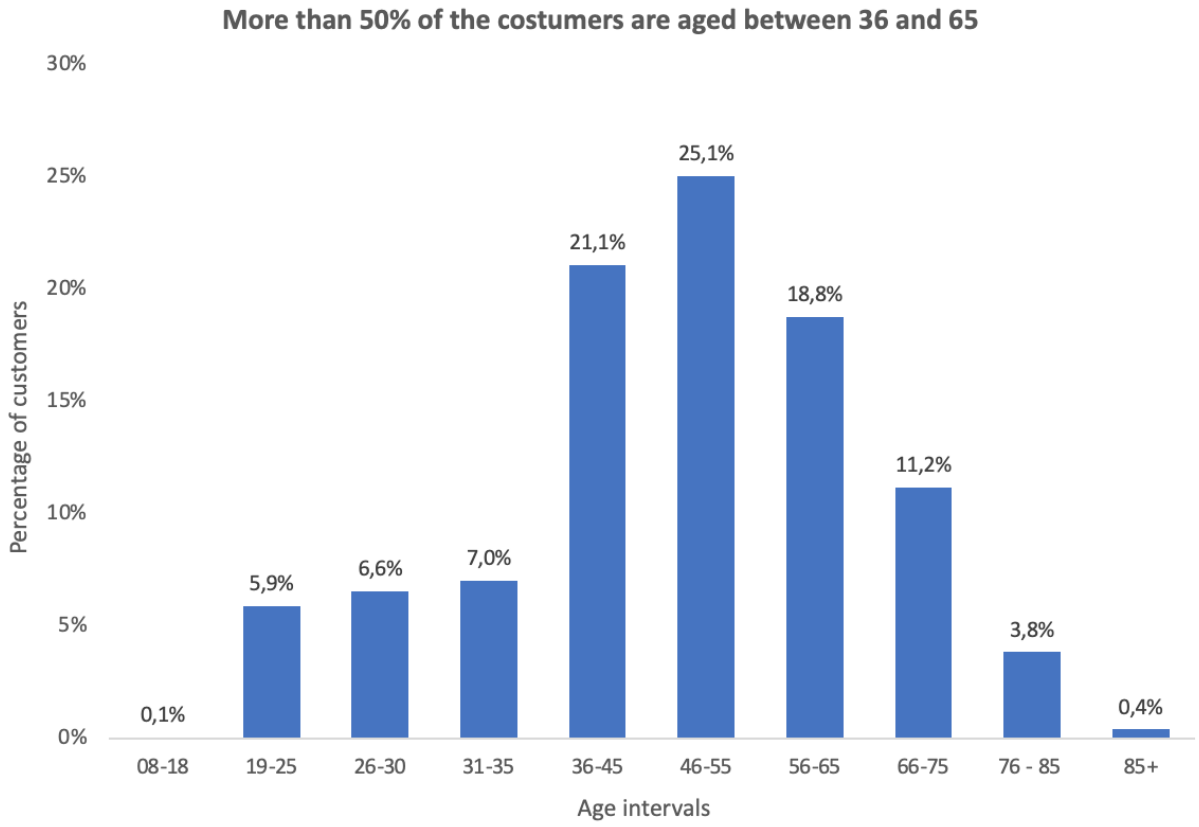


Figure 2 - Percentage of customers for each age interval. On the x-axis the age is represented and in the y-axis the percentage of customers.

Another important feature regarding the customer is the 'QTD_ANOS_ANTIG_CLIENTE', representing the number of years since customers are in Fidelidade. Values for this variable range between 0 and 95, being the mean equal to 15 years. In Figure 3, one can observe that 50% of the customers are in Fidelidade for more than 15 years.

It is interesting to link this information to the leadership position of Fidelidade company in the insurance sector in Portugal. It is known that Fidelidade is seen as a friendly brand, accompanying the client at every stage of his life, this strategy can be seen in this graph.

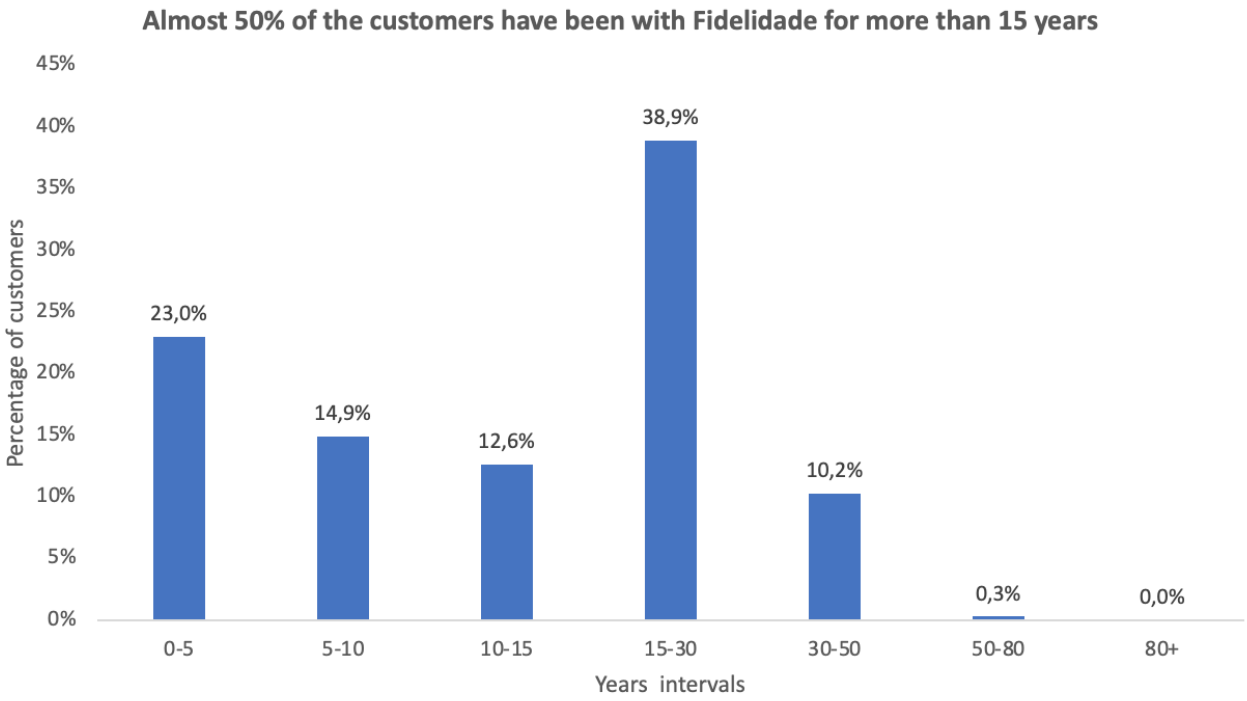


Figure 3 - Percentage of customers by years interval. On the x-axis, the number of years of seniority is represented and on the y-axis the percentage of customers.

Still, on the customer side, some variables bring relevant information about the customer this is the case of: 'QTD_SIN_CRIADOS_UMES', 'QTD_SIN_CRIADOS_U3M' and 'QTD_SIN_CRIADOS_U6M'. These variables represent the number of claims in the last month, three months, and six months respectively. Currently, Fidelidade simulator is using this information to help recommend auto insurance products to its customers. This variable affects directly the premium paid for a product, the higher the number of claims the higher the premium. By looking at Table 1, it is possible to see that most clients present in this database did not have a claim in the last 6 months. Lack of claims can reveal a variety of characteristics about the client, but perhaps most specifically that he may be a safe driver, making him the kind

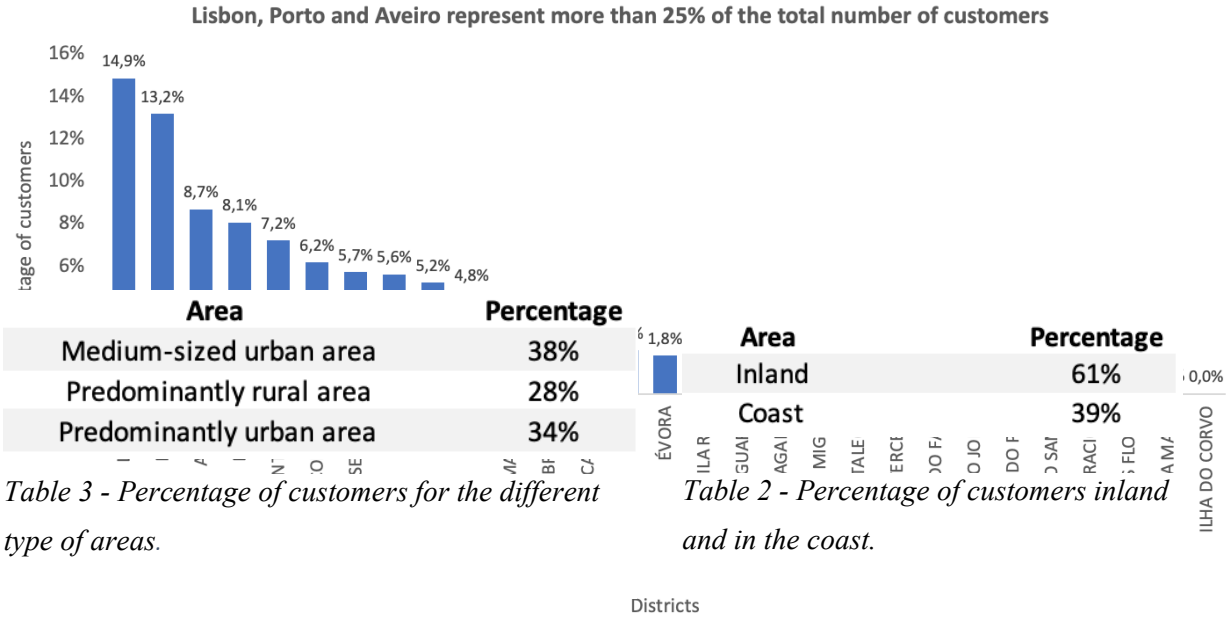
of client Fidelidade values.

Regarding geographical information about the customer, several features on the database are interesting to look into more detail, this is the case of 'DSC_DISTRITO',

Table 1 - Percentage of customers by number of claims.

0	14%
1+	86%

'DSC_INTERIOR_LITORAL', 'DSC_RURAL_URBANO'. These three variables represent respectively the customer district, if it is inland or coastal and rural or urban. By looking at **Error! Reference source not found.**, it is possible to conclude that more than 25% of the total number of customers live in Lisbon, Porto, and Aveiro while only 4% of the auto customers belong to the islands (Açores and Madeira). These numbers are not a surprise since the number of people living in Lisbon, Porto, and Aveiro is much higher than the population of the islands.



Area	Percentage
Medium-sized urban area	38%
Predominantly rural area	28%
Predominantly urban area	34%

Table 3 - Percentage of customers for the different type of areas.

Area	Percentage
Inland	61%
Coast	39%

Table 2 - Percentage of customers inland and in the coast.

Figure 4 - Percentage of customers by district. On the x-axis districts are represented and on the y-axis the percentage of customers.

Still in the geographical analysis, one can see that 61% of the customer live in an inland area of the country while 39% live on the coast (Table 3). Regarding the type of area, 38% live in a medium-sized urban area, 28% in a predominantly rural area, and 34% predominantly urban

area Table 2).

At this stage, a typical auto customer lives in Lisbon, Porto, or Aveiro. Is between 35 and 65 years old and is a client of Fidelidade for more than 15 years. Regarding his professional activity, there are more than 53% of customers from whom the information is not available, from the other 47% it is known that 22% are currently working, 10% have a proper company, 2% are retired and the rest either a student or unemployed.

There is still important information to be included in the customer analysis. The market department from Fidelidade built a variable (code with three characters) that includes three characteristics from the customer: age, income level (1- Low, 2-Medium, 3-High), and the channel the customer used to buy the auto insurance. From this variable, it is possible to conclude that almost 30% of the customers that bought auto insurance are adults, with medium income levels, and went to an agent to buy their insurance (A2A). It is also interesting to note that the second segment with the highest percentage is A1P meaning adults with low-income levels who bought their insurance in person.

2.1.2.3. Car analysis

Regarding the car data side, several features stand out when looking at the 105 features available. It is important to refer that several features came through the Eurotax database, this latter is accessed by inserting the car plate on the form available on the company's website, currently, Fidelidade is using them to recommend insurance packages to customers. These variables are interesting to be analysed since they bring detailed information about each car available on the dataset.

Starting by analysing the top models used by auto customers in Fidelidade, there were 90 different models, but to be able to take insights from the graph only the top 10 models are shown in Figure 5. It is possible to conclude that, from the graph, Renault is the one with a higher percentage representing 12,5% of the auto customers in Fidelidade. This latter is followed by Peugeot, Opel, and Volkswagen the three of them with percentages between 9,7% and 7,7%.

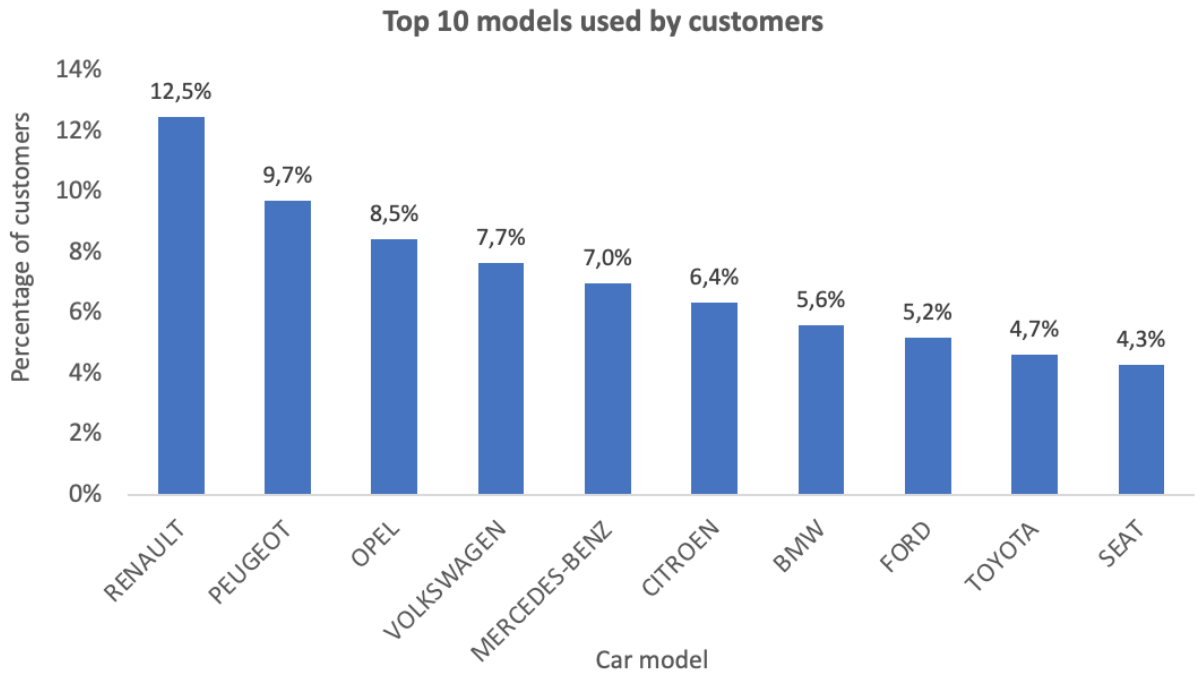


Figure 5 - Ten models with high customer percentage. On the x-axis the car model is represented and on the y-axis the percentage of customers.

By analysing three other variables 'EUROTAX_FUEL', 'TIPO_VEICULO', DSC_TIPO_UTILIZACAO_VEIC, meaning respectively the car fuel, the type of vehicle (light, commercial or motorcycles), and type of car utilization (bought or leasing). It was possible to observe that 62.8% of the cars run on diesel, it was also interesting to note that 99% were bought not leasing and 84.2% were light vehicles (as seen in Figure 18 and Figure 19 in the appendix).

Another car feature that was explored was the horsepower of each car, for this variable, bins were created based on 10 quantiles. Meaning that, by looking at Figure 6, the first bar represents the number of cars that has equal or less horsepower than 10% of all cars in the database, and the last bar represents the cars that had equal or more horsepower in them than 90% of all cars.

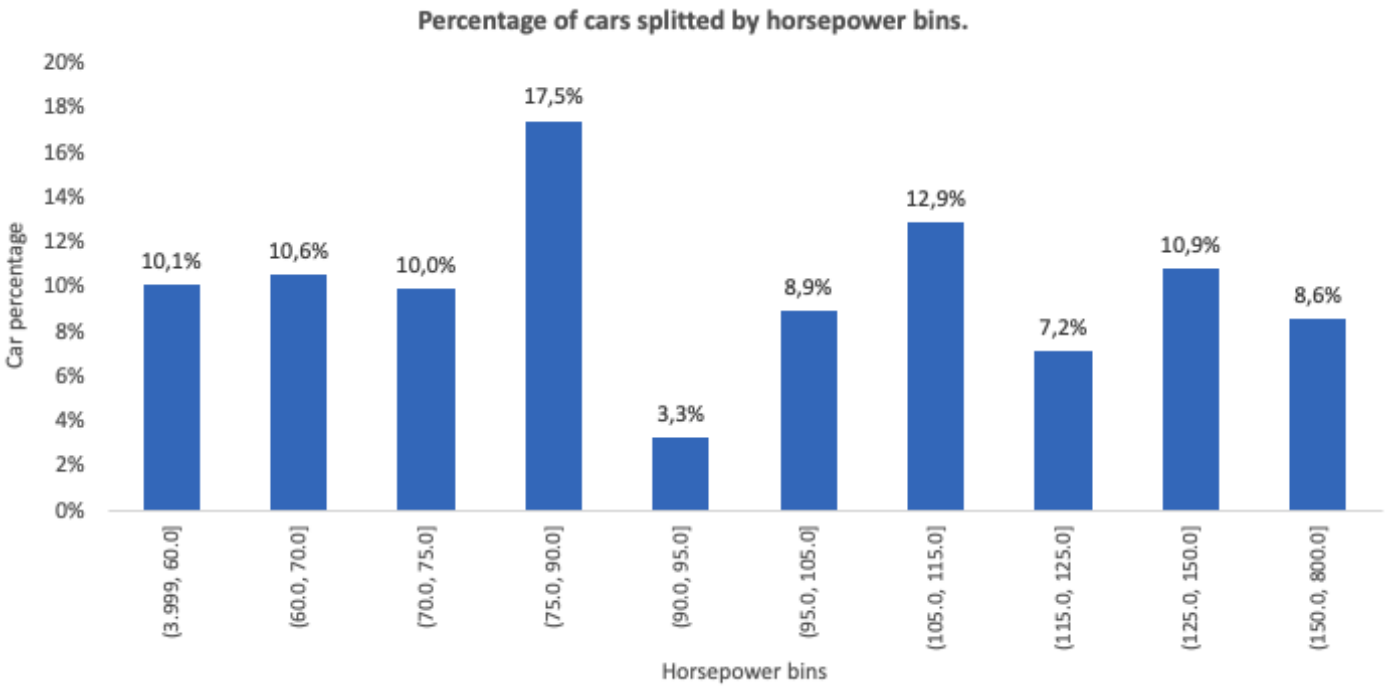


Figure 6 - Car percentage by horsepower bins. On the x-axis horsepower bins are represented and on the y-axis the car percentage.

In Figure 6, one can observe that 17.5% of the cars have a horsepower between 75 and 90, and also that more than 50% have a horsepower below 105, meaning that most cars in the database tend to have low horsepower.

All information stated above is used to understand the type of car that is most frequent in the database.

2.2. Data cleaning

The following section will be focused on how the missing values were handled. In this

section, some techniques used to clean the data and to convert several datatypes into others are also going to be explained in more detail.

The first step towards cleaning up missing values was to analyse and understand if the percentage of this latter was or was not significant. In fact, when exploring the database, several features had high percentages of missing values, incompatible with the proper functioning of the model. Faced with this situation there are three possible solutions: fill the null values with a new value, drop the rows (this happens when the percentage of null values is not significant) meaning that the database will lose car-client pair observations or drop columns (this happens when the percentage of null values is very significant, in the work project this boundary was set in the 15% of missing values) losing features.

This being said, the data cleaning step started by filling several variables with meaningful data. For categorical, most missing values were filled with 'NODATA' or 'Desconhecido' this was the case of seven variables (EUROTAX_TRACCAO, DSC_RURAL_URBANO, DSC_INTERIOR_LITORAL, DSC_DISTRITO, COD_SEGMENTO, COD_ESTADO_CLIENTE, and DSC_SITUACAO_PROF) while for numerical data some missing values were filled with the mean value and some others with 0 when this latter did not have already a meaning. That was the case of the variables related to the number of claims that had 30% of missing values (QTD_SIN_CRIADOS_UMES, QTD_SIN_CRIADOS_U3M, QTD_SIN_CRIADOS_U6M) where the value zero already had a meaning, here the strategy was to fill the missing values with the mean value this was also applied in the case of TAX_CRIME_CONDUCAO_ALCOOL_DICO (criminal rate related to driving under the effect of alcohol by geography) that had 18% of missing values. At the end of this cleaning phase, all the variables had less than 6% of missing values, meaning that the option of dropping the number of rows would not result in a big loss for the dataset.

The second step of the data cleaning phase was correcting the datatypes of several features. The biggest challenge occurred when trying to convert categorical variables into a float, this happened due to the presence of several different number separators such as commas and dots. This step was executed on twelve different variables, most of which were related to the value of the policy or with the capitals selected for several coverages. The method used to understand which categorical variables should be converted into numerical ones was to look at the variables that had a low number of unique values.

There was also the need to standardize some variables, i.e. some variables had some values in upper case that were considered different from the ones in lower case. This was the case of the DSC_SITUACAO_PROF, this variable changed from having 15 unique values to 8.

At the end of this data cleaning phase, the data set had 59 features and 247k rows, meaning that only 7,5% of the initial information was lost.

3. Models

The purpose of our project, as mentioned above, is to build a recommender system for Fidelidade auto insurance. Currently, they have a model where the user solely needs to input basic information about himself and the car and the actual system will recommend three of the already existing insurance packages (Auto 1, Auto 2, Auto 3, Auto 4, and Autoestima). Fidelidade wants to design a more personalized offer and recommend it to the customer to retain them as soon as the simulation started.

Before moving on to the modelling part, it is necessary to understand the type of models that exist and how they can be useful in achieving the objective that has been proposed.

There are three types of recommender systems models: collaborative filtering, content-based, and hybrid models. It is important to have in mind that even though several models need fewer data, the result and the accuracy of the latter is directly related to the quantity and quality of the data.

3.1. Collaborative Filtering

Starting with the collaborative filtering model: the name speaks for itself it is a process of filtering (evaluating) items using the opinions of others. People are accustomed to seeking recommendations, advice, and opinions wherever they go. In reality, this kind of model is just a formalization of behaviour already existent. People turn to those close to them and whom they trust for advice first, whether at work, school, or home. When they want to see a new movie, if they know that person X typically has a similar taste, they will ask him for a recommendation. Similarly, if person Y is the expert on books, they will ask him for recommendations anytime they need a new book.

Instead of asking others for recommendations, the model suggests movies, books, hotels, and music based on other users who share the same interests. This type of strategy makes the search for suggestions much more efficient. The type of model that is behind these platforms can be divided into two different algorithms: user-based and item-based nearest neighbour.

3.1.1. User-based nearest neighbour algorithm

The user-based nearest neighbour algorithm is based on the similarity among users, this means that if the users are similar to each other (this similarity can be based on the rating each other gave to the movie, song or book…) then they might like the same items. Assuming that user n is similar to user u, one can say that n is a neighbour of u. This type of algorithm is used in Amazon "customers who bought this item also bought this" and used in Netflix "users who watched this movie also watched this one".

This type of model is based on the similarity between users. This metric should be computed considering not only the rating that users gave to the item but also considering that not every user rates in the same way (there are some users for whom four out of five is a perfect movie and some others where four is just a normal movie), therefore the similarity between users should be calculated using Pearson correlation. This formula ranges from 1 for users that have a perfect agreement to -1 for users with perfect disagreement and takes into consideration this difference between ratings.

This algorithm is very useful and widely used nowadays. However, several practical challenges are important to be recalled. Since the information used to build the model is given by the user it is very common to deal with sparse information resulting sometimes in skewed correlations. Another problem with the user-based nearest neighbour algorithm is that it requires previous information about the user, it is very difficult to recommend to a new user a movie not knowing which movies he has seen and enjoyed. The scalability is also an issue when talking about this algorithm, when computing a large number of combinations between users and items this could lead to a problem regarding the duration of the computation.

3.1.2. Item-based nearest neighbour algorithm

The item-based nearest neighbour algorithm is very similar to the user-based one, but instead of computing the similarity between users, the similarity is calculated between items (i.e in the case of movies the similarity is calculated between movies and not between users). Instead of recommending items to a user using the similarity between users, items will be recommended based on the similarity between them. An example could be user u likes item i_1 and item i_1 is similar to items i_2 and i_3 therefore items $i_{2 \text{ and } 3}$ will be recommended to user u. The most popular similarity metric is the adjusted-cosine similarity, which is computed using all users who have rated both items i and j. The formula can be seen below where $\overline{R_u}$ is the average of the u-th user's ratings. This metric takes into account the difference in the rating scale between different users by subtracting the user average rating from the co-rated pair. The adjusted cosine similarity metric is calculated alongside the columns.

$$ajusted\ cosine\ similarity(i,j) = \frac{\sum_{u \epsilon U}^{n}(R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \epsilon U}^{n}(R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \epsilon U}^{n}(R_{u,j} - \bar{R}_u)^2}}$$

There are two more similarity metrics that can be used to calculate the similarity between items such as cosine-based similarity and correlation-based similarity this latter is the same as the Pearson correlation metric used as a similarity in the user-based nearest neighbour algorithm.

To calculate the cosine-based metric the two items are considered as being two vectors and that way the cosine is computed, this value is used as being the similarity between two items.

$$cosine(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}} \frac{1}{\sqrt{\sum_1^n b_i^2}}$$

where $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

As seen previously, there are several possible ways to compute the similarity metrics, therefore, in order to be able to understand better which type of metric to use depending on the different scenarios, there are some cues that can help to decide:

- Pearson correlation should be used if the data is subject to user-bias or different rating scales

- Cosine should be used if the data is sparse

- Adjusted cosine should be used for an item-based approach to adjust for the user-bias

About the implementation of this model, it follows the same steps as the user-based algorithm, the only difference is the subject to which the similarity refers to. In the user-based approach, the subject is the user and, in the item-based approach the subject is the item.

In conclusion, collaborative filtering models are very useful when implementing recommender systems and do not require much data. When trying to implement a recommender system taking in consideration historical data about the user and the item, this model is not the most valuable. Since this work project intends to use user and car data to suggest the right insurance package this type of model is not the most interesting instead, the content-based model can be used to achieve the objective.

3.2. Content-based model

Content-based model is another type of recommendation system that is very useful to come up with some predictions on classification problems. This type of model is very useful due to its ability to use user and item metadata. In our project that is the main purpose, therefore this type of model was the one selected for the modelling phase.

The content-based model works either with explicit feedback (for example ratings on movies) but it also works with implicit feedback (for example the number of clicks a user has made). Just like the collaborative filtering type of model, the content-based also has some practical challenges: the cold start is not only a problem in the collaborative filtering type of model, but it is also a problem in the content-based, if there is a new user with no information, it will be very difficult for the algorithm to recommend a product to that user.

Diving deeper into the content-based type of models, it is known that by inputting metadata about the user and the item the type of recommendation produced will be more accurate than in a scenario without metadata. In this type of algorithm, it is easy to explain the result since the latter is only dependent on the metadata that was fed into the model. The explainability characteristic is one of the three important objectives that were mentioned above that need to be taken into consideration when building the model for Fidelidade.

In this content-based algorithm, there are several models that are going to be very useful in order to achieve the desired results. Those are going to be explained in more detail in the following sections.

3.2.1. Decision trees

Decision trees are a type of model that aims to represent the decision-making process using a tree-like structure. Since this work project objective is to build a classification model

based on the user metadata one can think of a tree as being the path to reach the final insurance package.

This type of model is very popular due to its simplicity regarding the visualization of its results and due to the facility of implementation. Inside this "family" of tree-like structures there are several other models that will be used for our project, that is the case of the Random Forest and LightGBM, both are derivatives of the basic decision tree.

3.2.2. Random Forest

The random forest model can be seen as being a set of several decision trees. It is a very interesting model due to its diversity; it can be used both for the regression and classification types of problems. The parameters are almost the same as the ones used in the decision tree model, it is possible to define the number of branches, the number of trees, the weight of the classes… This model adds additional randomness to the model and, rather than looking for the most crucial features when splitting a node, it looks for the best feature from a random subset. In general, it delivers extremely good results. (Urwin 2022)

Another great quality of the random forest model is that it is very easy to measure the relative feature importance of each feature for the predictions. This characteristic is very useful when trying to improve the models' results, this being said, after knowing the importance of each feature, the ones that are not even considered for the model can be dropped and consequently increase the performance of this latter.

Comparing the performance of decision trees and the random forest

There are some differences between decision trees and random forests, in fact, a random forest is not just a set of decision trees. While a decision tree uses the input data to build a path to the label desired, a random forest uses random features and random observation to build its trees. This difference provokes, sometimes, a challenge of overfitting the decision trees that do

not happen on the random forest due to this random way of computing results. Another derivative of the decision tree algorithm is the LightGBM.

3.2.3. LightGBM – Light gradient boosting machine

LightGBM is a gradient-boosting framework that uses a decision tree based on learning algorithms to support efficient parallel training with faster training speed, lower memory consumption, higher accuracy, and faster processing of massive data (Wang 2019). LightGBM is a Gradient boosting decision tree (GBDT) that contains two different techniques: Gradient-based one side sampling and Exclusive Feature Bundling to improve the challenges that the original GBDT had when processing a large amount of data and a large number of features respectively. After several experimental results, it is shown that these two techniques enable LightGBM to outperform XGBoost in terms of computational speed and memory consumption. (Ke 2017).

Just like the random forest model, LightGBM has the possibility of computing the feature importance and consequently improving the performance of the model by dropping features that were not relevant to the model. This characteristic of the model is one of the most important attributes since it helps explain the result of the model and in the future will explain to the customer why product x was recommended to him.

Another interesting model that can be useful for our project is the Softmax regression, to understand this algorithm it is necessary first to explain the model that is in its base: the logistic regression.

3.2.4. Logistic regression and softmax regression

The logistic regression type of model is usually seen as being a binary classification model (predict only class 1 or 0), in this work project the multi-class functionality, which is available by defining the parameter multi-class equal to multinomial, will be needed. This is the step where the softmax activation function enters. The output of the softmax regression model is a vector of length c (c representing the number of classes existent) with the probabilities of each class. In other words, the cth value in the vector represents the probability of the prediction being the cth class. (S. Verma 2021) In this case these values represent the probability of a certain user acquiring one of the already pre-defined insurance packages. These types of values are very useful when thinking about, for example, the agents that sell by themselves auto insurance to clients, here they have the possibility by inputting certain information about the client to understand how likely is it that the client x will buy insurance y.

3.2.5. KNN k-nearest-neighbour

The KNN (k-nearest-neighbour) model is another example of a content-based algorithm. It is based on the idea that you can anticipate the features of a data point based on the features of its neighbours. This kind of prediction might work in some circumstances but not in others. Considering neighbours as a simple illustration of how this works. Frequently, two neighbours have similar interests. They most likely belong to the same socioeconomic group. They might work in the same field, send their kids to the same school, etc. However, this method is not as effective for other tasks. For instance, it would be absurd to try to determine their preferred colour by simply looking at one of the neighbours. To find the distance that can be used to locate the data points that are most nearby the point that is going to be predicted, also called neighbours, the Euclidean distance formula shown below is used.

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Where

- $(x_1, y_1)$are the coordinates of one point.
- $(x_2, y_2)$are the coordinates of the other point.
- Euclidean distance is the distance between both points.

In this work project, KNN will be useful since depending on the historical data about the users. Those will be compared to some neighbours and this way the insurance bought by the neighbour will be recommended to the client concerned.

To accomplish our objective, three content-based models will be used. In addition to these models and the collaborative filtering models, it's critical to highlight the final category of recommendation system models: the hybrid models.

3.3. Hybrid models

The term "hybrid recommendation system" refers to a particular kind of recommendation system that combines collaborative filtering with content analysis. In some circumstances, combining collaborative and content-based filtering can improve performance and help overcome the drawbacks of utilizing them independently. There are several ways to implement hybrid recommender system approaches, including employing content and collaborative-based methods to generate predictions separately, then combining the predictions, or simply enhancing a content-based approach with the capabilities of collaborative-based methods (and vice versa).

Using the methods, data can be separated into two categories that can then be used to create a recommendation system:

- Explicit Feedback: Information that includes direct user feedback. Explicit feedback may take the form of a user rating that expresses how the user feels about the product, such as whether he enjoyed it or not.

- Implicit Feedback: This information is not about the rating or score that the user provides; instead, it may be some details about clicks, movies viewed, music played, etc. (Y. Verma 2021)

Explicit feedback can be thought as being the component similar to the collaborative filtering, and the implicit feedback the component similar to the content-based.

In conclusion, as it has already been mentioned, the purpose of this project is to suggest insurance packages (Auto 1, 2, 3, 4 and autoestima) based on customer metadata. It's also important to remember that Fidelidade dataset lack ratings, each customer only bought one insurance for his car, and people who own many cars bought the same insurance for each one. This means that the aforementioned content-based models will be the main focus of our work as the hybrid model cannot be explicitly fed back and the item user matrix for collaborative filtering cannot be appropriately tuned to the model.

3.4. Evaluation metrics for recommender systems

Following the presentation of the models, it is critical to comprehend how algorithms will be assessed and what metrics will be most useful in achieving the desired outcome. Since the problem in consideration is a classification one the metrics that will be presented are those that can be useful for the project. First and foremost, it is important to comprehend that a successful outcome would be to anticipate with as much accuracy as possible the insurance package that the consumer would purchase. With this in mind, there are several metrics that are typically used to assess this kind of situation. That is the case of precision, recall and F1 score. (Korstanje n.d.)

- Precision is the ratio between the True positives and all the positives. For Fidelidade's problem that would be the measure of customers that bought that auto insurance package out of all the patients that bought it. Mathematically it is represented by the equation represented bellow:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- Recall is the measure of the model correctly identifying True positives. Thus, for all the customers who actually bought the right insurance package, recall tells how many are correctly identified as buying the right insurance package. Mathematically:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- Accuracy is the ratio of the total number of correct predictions and the total number of predictions. The mathematical formula is the following:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}$$

- F1-Score is the harmonic mean of the precision and recall, it is calculated by the formula shown below:

$$F1 - score = 2 * \frac{Precision + Recall}{Precision + Recall}$$

All the metrics listed above can be calculated using the confusion matrix, which gets its name from the fact that it makes it straightforward to determine whether the system confuses

two classes. Both the existing labels and the expected labels are represented by the latter. The occurrences in a predicted class are represented in each column of the matrix, whereas the examples in an actual class are represented in each row of the matrix. The performance of our content-based models will be assessed using this matrix and all the aforementioned metrics. (Wikipedia n.d.)

4. Results

In the following section three individual work projects are going to be exposed. As mentioned in the introduction, each one of the individual parts will be talking about one of the three pre-defined objectives: predictability, explainability and flexibility.

4.1. Collaborative filtering as a tool to explain predictability

Collaborative filtering is an unsupervised machine-learning technique. This means that the outcomes are unknown. Compared to a classification where a target variable is given, and the classes which the users should be assigned to is already known before the classification is made, with an unsupervised machine learning technic such as clustering, the outcome is not known beforehand, there is no target variable. The goal is that clusters (user-groups in our case) should be created based on the similarity between records.

In this paper, models are built on two main approaches. Content-based recommendation systems and collaborative filtering models. Our assumption is that collaborative filtering will not yield as good results as content-based recommendation systems, especially if explainability is taken into account, which is one of the objectives of this paper. On the other hand, accounting for predictability is also an objective that must be fulfilled, thus trying out collaborative-filtering techniques is required for the completeness of this paper.

Before diving into the creation and the results of collaborative filtering, some differences between this work project and other cases where collaborative filtering is typically used must be discussed.

Collaborative filtering is typically used when there are multiple items that users are rating. A certain user will rate some of the items, but more likely he or she will not rate all the items. Then the similarity between either users or items is calculated and that similarity is used to predict a rating value for a certain item by a certain customer. This rating will be later used to recommend items to users. This is the case of movies or music recommender systems used by Netflix or Spotify. Comparing these latter with the purpose of this work project there are obvious differences. Most importantly customers (users) do not try out multiple insurance packages (items). This means that they either buy a product, and rate it with 1, or do not use a

product and rate it with 0. A customer will never rate two products with 1 because it would mean that the customer uses multiple insurance packages at the same time, which is impossible.

Taking into consideration, the lack of normal rating numbers and the lack of missing ratings of certain customers for certain items, this is hardly a dream scenario to use collaborative filtering. However, creating clusters of customers and building a user-item matrix where clusters are seen as users, insurance packages are seen as items and the values that fill the matrix are the ratings calculated based on how many customers in each cluster chose a certain item, is possible. With a matrix like this, it is possible to use collaborative filtering to recommend insurance packages to a customer.

4.1.1 Dimension reduction and data visualization

Visualizing the data would be very useful since it might help one identify certain clusters without using any clustering algorithm. The problem with the visualization of the data is that it is not two-dimensional. There are 76 variables, which would mean 76 different axes to visualize all the data points.

Thus, to visualize the data, dimension-reduction techniques needed to be used. The goal of dimension reduction is to reduce the dimensionality of the data without losing too much information. Principal component analysis (PCA), one of the best-known techniques, where the algorithm creates new variables that are linear functions of those in the original dataset, that successively maximize variance and that are uncorrelated with each other (Jolliffe 2016) was used. The algorithm was able to create two new variables, two principal components, in such matter that the explained variance ratio by these principal components was 88% and 7.2% respectively.

Another dimension reduction technique is called t-distributed stochastic neighbour embedding (t-sne). T-SNE is a technique that visualizes high-dimensional data by giving each data point a location in a two or three-dimensional map (Maaten 2008). T-SNE requires great

computing power and time, thus in the case of very high dimensional data, it is considered as good practice to use different dimension reduction techniques before using T-SNE. However, since there are 76 variables it is not necessary. The computation power of the machine provided by Fidelidade was enough to run the T-SNE on the whole dataset and it took 1102 seconds to run.

When comparing the results of the PCA and the T-SNE, seen in Figure 7, one can observe that T-SNE performs better than PCA. In both cases, the first component has a strong effect on the data, and on the outcome of the insurance plan, and the second component has almost none.

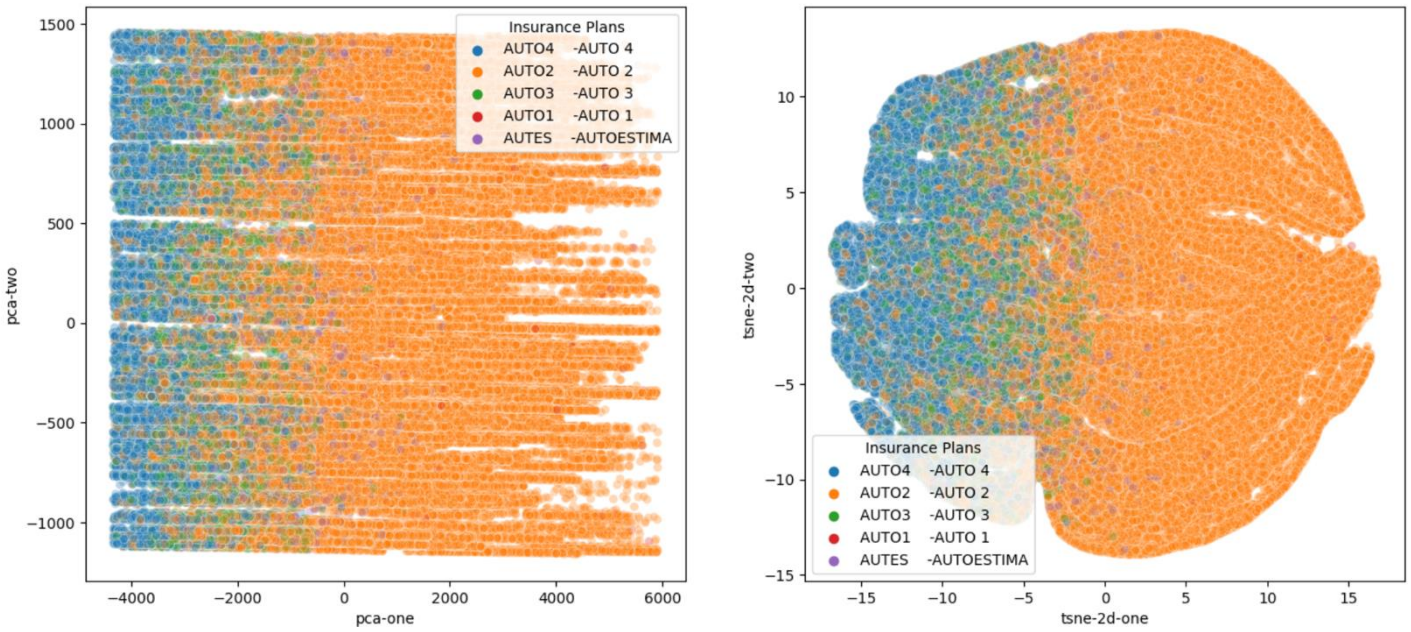


Figure 7 - Comparing PCA and T-SNE results.

One can also see that in both cases low values on the x-axis usually mean AUTO4 or AUTO3 insurance plans, and high values tend to respond to AUTO2 or rarely AUTO1. However, on the y-axis an observation like this cannot be made. On the PCA graph (left) within each value of y all the insurance packages can be found, there is no differentiation between data points based on the y-axis. On the t-sne graph (right) the situation is very similar, however, the separation is a little better than with the PCA, since under the -12 value on the y-axis one cannot find AUTO4 insurance plans anymore unlike with the PCA.

To conclude, the information that the data visualization revealed; Clear clusters are hard to identify, and it is not possible to observe clear groups of instances but rather a big mass of data points; The clusters are imbalanced in cardinality, as the data is imbalanced as well, there are way more Auto 2 customers than in any other insurance plan category; The distance between members of each cluster can be very long.

4.1.2 K-means Neighbours clustering

K-means clustering is an algorithm that creates a given number of clusters with centroids for each cluster and calculates the distance from these centroids for each data point to assign them to one of the clusters. Our assumption is that K-mean wouldn't perform well in this case since as it was concluded in the data visualization part, the distance between members of each cluster can be extensive, thus calculating means could be misleading.

Another downside of K-means is that this algorithm creates balanced (even-sized) clusters while a clustering algorithm that could handle imbalanced data would be needed, such as the OPTICS algorithm from the sklearn.cluster library. However, the computing power required by this algorithm is immense and not supported by the virtual machine provided by Fidelidade, thus it could not be included in this work project. Just as using K-median instead of K-means. K-median is very similar to K-means just instead of centroids and means it calculates with medians, thus it's less sensitive to outliers, which could help us create clusters where data points are far from each other.

When running the K-means Neighbours algorithm, the first thing that was observed was that the inertia, the distance between each data point, and its centroid is colossal, as was expected beforehand.

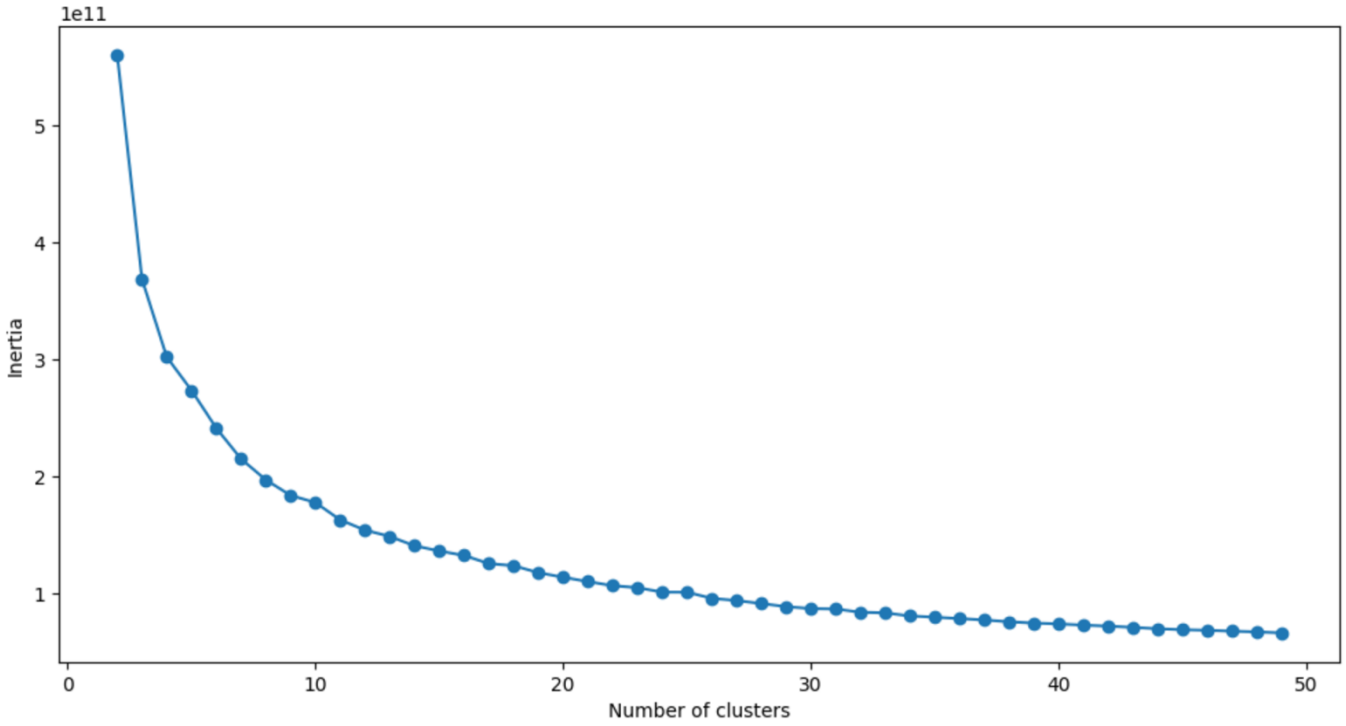


Figure 8 - Elbow Method

In Figure 8 one can see the relation between the inertia and the number of clusters. The more clusters created the smaller the inertia becomes. As a best practice, the number of clusters chosen should be the one that minimized the inertia (increasing the performance) and the one that does not increase the running time greatly. In other words, the number chosen should be the one where the line starts to flatten and converge to its minimum value.

By following the elbow method, the number of clusters should be between 9 and 13. However, it is necessary to keep in mind that this first clustering is only to create the user-item matrix, where these clusters should function as users. Thus, it is reasonable to choose a high number of clusters, as it would be preferable to have more than 50 customers rating the products.

In the end, K-means clustering with 1000 clusters was used, which resulted in inertia of more than 15 billion and where the most populated cluster had 538 instances in it, and the least populated one had 49. With these clusters, a user-item matrix was built, as seen below in Table 4 where the rows of the matrix represent the clusters, the columns of the matrix represent the items and the values of the matrix represent the sum of the products that had been bought by the members of each cluster.

CUSTOMER GROUP	AUTO1	AUTO2	AUTO3	AUTO4	AUTOESTIMA
0	1	327	0	0	3
1	0	71	67	107	3
2	0	239	0	0	8
3	2	373	0	0	0
4	0	54	45	70	3
...	...	...	...	...	...
995	1	387	0	0	0
996	0	51	46	146	0

Table 4 - User-item matrix. The customer group column represents the clusters built.

Now in any typical collaborative-filtering problem, the user-item matrix would be used to calculate the missing rating points and based on that recommend new items to the users. In our case, the KNNWithMeans algorithm from the surprise library was used, with a set-up to calculate a user-based similarity calculation (section User-based nearest neighbour algorithm).

With a 26.5373 Random Mean Squared Error, on a rating scale from 0 to 548, this suggests a 99.06 percent accuracy, which in theory would be our best model based on predictive power. However, our business problem was not how to offer new products to already existing customers, but how to offer products to new potential clients, who haven't bought any insurance yet. Using the user-item matrix a choice of auto insurance plan could be assigned to each customer group. Calculating the mean ratings of each insurance plan across all customer groups, the insurance plan with a rating where the difference between the rating and the calculated mean is the biggest will be assigned to each cluster. This way the desired insurance of each customer

group would be known, and the only thing that affects the model's predictive is the accuracy of the clustering algorithm.

4.1.3. Conclusion

Collaborative filtering is a powerful tool to use for great predictability. On the other hand, a big downside of this technique is that it lacks explainability. It would be impossible to know why the model offers a certain insurance plan to a customer since the values that the clustering was based on are unknown.

Our business problem was to recommend insurance plans to potential new clients, but with classic conditional filtering method only the recommendation for customers (or customer groups), who had already bought some insurance plans before, was possible. Therefore, with the combination of clustering and the user-item matrix the desired insurance plan was assigned to each customer group, and a method was created where the distance between a new entry and each cluster centroid is calculated, then the new entry is assigned to the cluster with the centroid being closest to it. This way the model can recommend insurance plans to new clients as well.

Since the data is unbalanced, finding, and evaluating different clustering algorithms that create unbalanced sized clusters, such as OPTICS, for example, could be the subject of further studies

4.3 Recommending coverage packages with the focus on flexibility and personalization

The main purpose of this paper is to find a solution for the business problem of recommending insurance packages for customers with a bigger efficiency than the current agent-based, manual method that dominates the insurance industry, with a focus on predictability, explainability, and flexibility. In the following chapter, with a focus on flexibility, it will be demonstrated how to create a system that not only recommends basic products (insurance packages) offered by the company, but also extra coverages included in each of those insurance packages, thus increasing the personalization aspect of the model. Various machine learning models are presented referring to the ones mentioned in the Models section of this paper, models which are built based on two significantly different approaches. Considering the length of this paper, only the best performing models are discussed in detail by both approaches.

4.3.1 Exploring coverages

Fidelidade offers extra coverage for each of their insurance package. While in some cases a coverage is mandatory, thus already included in the package, in other insurance packages the same coverage appears as extra. A more thorough description of each package and the coverages offered in them can be found in the appendix in Table 20 and Table 22.As our purpose is not to build a recommendation system for insurance plans anymore, but to build one for coverages, our target variable changes as well. A new target variable means that our data must be explored again in the reflection of the new target variable. While the course of the EDA process was already discussed in detail in the Exploratory data analysis section of this paper, in this section only a short review of the funding is provided, that our EDA revealed about the connection between the dependent and independent variables. There are 14 ‘FLAG_VEIC’ variables in the data. Apart from ‘FLG_VEIC_COB_DP_M0’, each of these variables represents weather a customer has that certain coverage for their car in question, while

'FLG_VEIC_COB_DP_M0' is a flag that indicates whether a customer had extra coverages included in the plan associated with his certain car. All customers expect the ones with AUTO1 insurance package to have at least one extra coverage included in their plan, while AUTO1 is a base package that offers no extra coverage for the customers. 'FLG_VEIC_COB_PROT_JUR_M0' is the flag responding with the coverage that covers any expenses that a client has associated with jurisdiction. 99.77% of the clients have this coverage included in their package. Analyzing the distribution differences between all customers and the ones that do not have this coverage, while the distribution is quite similar, clients from Aveiro tend to not have this coverage with a higher percentage when compared to other locations (**Error! Reference source not found.**)

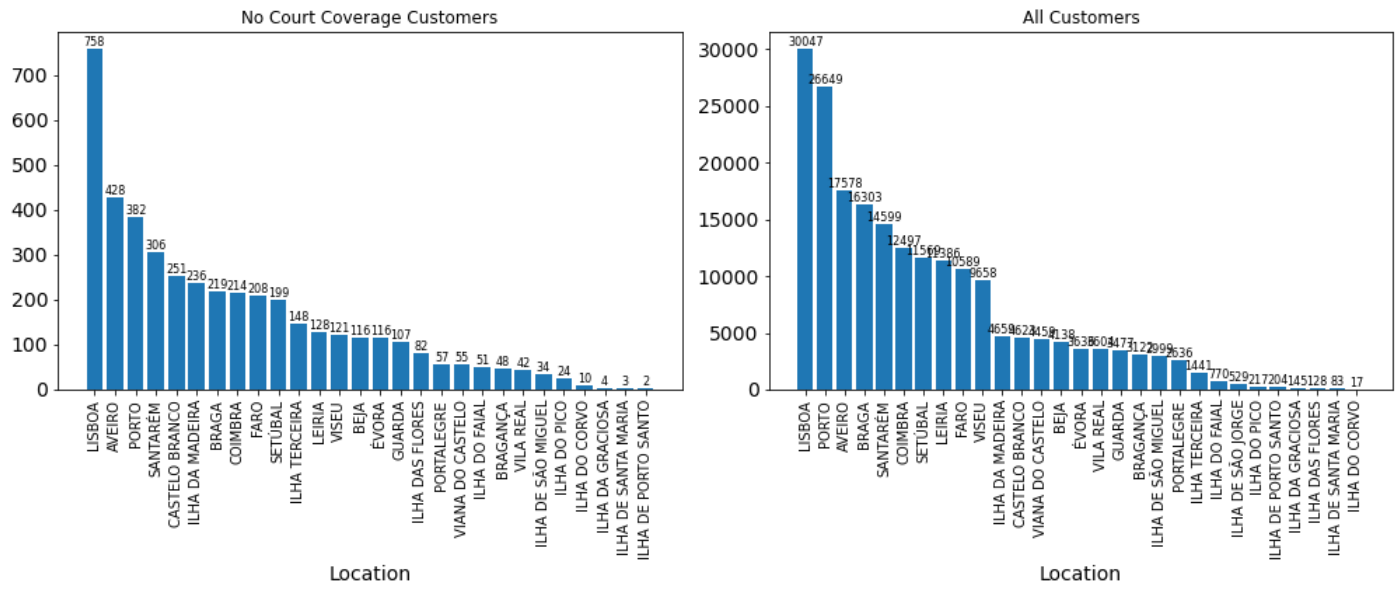


Figure 9 -Comparing the distribution of No Court Coverage Customers and All Customers based on location.

This observation indicates that the 'DSC_DISTRITO' variable might have a slight influence on the outcome of our target variable. Another observation similar to this is the correlation between 'EUROTAX_MODEL' and 'FLG_VEIC_COB_PROT_VIT_M0', where the former represents the model of the car in question, while the latter represents whether the client included driver protection with the bigger capital option in his or her insurance plan associated with the car. The observation can be made that clients with bigger capital driver

protection coverage tend to drive different, more expensive models such as Mercedes A Classe and Nissan Qashqai, than the average customers. (Figure 13)

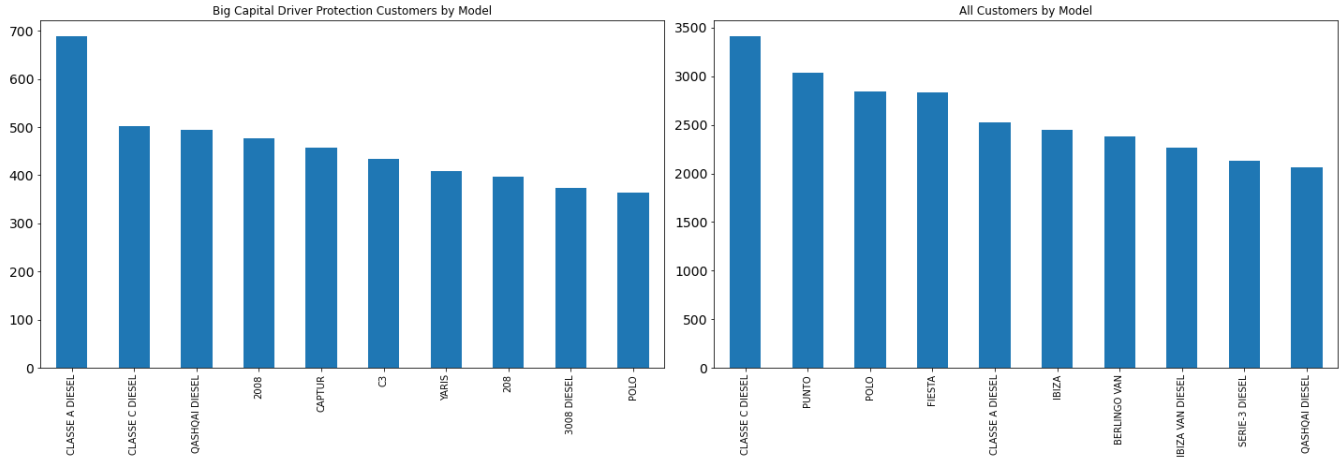


Figure 10 - Comparing distribution of customers with bigger capital driver protection with all customers.

The degree of difference in the distribution of data in these groups suggests a higher correlation between 'EUROTAX_MODEL' and 'FLG_VEIC_COB_PROT_VIT_M0' when it was seen before with 'DSC_DISTRITO' and 'FLG_VEIC_COB_PROT_JUR_M0'.

4.3.2 Creating Target Variable

To recommend extra coverages to customers two strategic approaches are considered, very different in their nature, with small variations within each of these two approaches. Both approaches centered around the problem of creating the target variable for our model, as this problem is the one that has the biggest effect on the outcome of our model. Each approach must fulfill the fallowing criteria; Predicting the target variable must help us recommend extra coverages to old and new customers with certain characteristics, on top of their base coverage package. The model also must be able to recommend these coverages with high certainty (predictive power) and must be able to explain why it offers certain insurance plans to certain customers (explainability). However, the focus of this passage is flexibility and personalization, thus the objective is to look for the perfect ratio in the tradeoff between personalization and predictive power. In other words, this work project intends to look for a model that recommends

coverage packages as personalized as possible with a good enough predictive power.

Our first approach (approach of extended insurance packages) is that all the possible combinations of optional coverages within each insurance plan are created, and the model recommends one of these combinations to clients. 57 different combinations were created, 1 for AUTO1, 24 for AUTO2, 13 for AUTO3, 13 for AUTOESTIMA and 6 for AUTO4. The full list of combinations can be found in the appendix in Table 22. Keep in mind that these coverages were created strictly based on the flag variables. This means that capital variables were not taken into consideration. For example, 'FLG_VEIC_COB_VEIC_SUB_M0' is the flag that indicates whether the customer had a replacement car coverage included in his/her insurance plan or not. While AUTO4 provides different types of replacement cars with different capital associated with them, the type of replacement car they chose is not represented in our target variable, as this information is only stored in the capital variables. The flag indicator would be 1 for each of the replacement car types. Also, note that not each of the created combinations had customers choosing that exact combination historically. However, there were still over thirty different categories in our target variable after including only the combinations with historical client usage, which is too many for our models to effectively categorize into. To reduce the number of categories our tactic was to only recommend the 5 "best seller" / most used coverage packages to those clients that fit the characteristics and recommend the base insurance package to the others. This would mean that our model has to classify customers into 10 different categories.

The idea behind the second approach (approach of coverage package types) was to use the base model first, the one that recommends basic insurance plans to customers without taking into inconsideration the extra optional coverages, then use another model that will recommend extra coverages for customers inside the first plan already recommended to them.

First, 3 different classes were created: base, extra and full. The base class represents the customers that have no extra optional coverages in their insurance plan, the extra represents customers, who have at least one extra optional coverage but does not have all, while the full class represents customers who had all the possible coverages included in their plan. Please note that, since in this scenario of creating the target variable, the type of insurance plan (AUTO1, AUTO2, etc.) recommended to a certain customer was already available for the model, the model only categorizes the data into "base", "extra" or "full" categories. Unlike in the previous approach, only the combination of the two model's results would give the whole picture of what to recommend to a certain customer.

Both approaches have their strength and weaknesses that will be discussed in detail in the upcoming part, where the results of certain models following these approaches are going to be represented and compared. Since the focus of this individual work project is on flexibility, some alternative options within the approach of coverage package types will be provided. These alternatives are based on the same principles as the two main approaches; Thus, it would not make sense to handle them separately, but it is definitely worth mentioning the results of these alternatives and seeing if they yield any benefits.

4.3.3 Comparing Models and Results

While the mechanisms of machine learning models that were used during the creation of the recommendation system were already discussed in the earlier **Models**section of this paper, it will not be discussed here again in detail. However, that previous knowledge will be used to understand the strength and weaknesses of these models in this certain scenario. In the "Comparing Models and Results" section various models are presented for the reader, comparing their results grouped by both approaches that were mentioned earlier.

An overall description of the predictive capabilities of the models for the approach of extended insurance packages can be found in the table below(Table 10) based on the selected KPIs, precision, and accuracy.

	Precision	Accuracy
Random Forest	**0.41**	**0.42**
LightGBM	0.39	0.43
Soft-max Regression	0.26	0.33
Naiv-Bayes	0.32	0.36

Table 5 - Overall evaulation of model performances with the first approach

The meaning of these KPIs will not be discussed in this section, since a description of both of them was already given in the Evaluation metrics for recommender systems section of this paper. Although, the obtained result along our selected metrics is discussed. The accuracy gives a good overall description of the performance of each model, although it is not sufficient on its own, thus precision and the confusion matrix were chosen as well as important KPI. Precision was chosen because our focus is on personalization, so our goal is to recommend each customer the precisely the package of their desire. Although the corporate world is still driven by profits, thus recommending precisely the insurance packages that are most used by customers, is more important than recommending insurance packages used by only a small number of very specific customers, which is also in connection with the confusion matrix. A confusion matrix for the best model in each approach is provided in this section, all other confusion matrixes can be found in the appendix along with the whole classification report for all models that had been created in this section.

Based on our metrics the best model was The Random Forest Classifier, with 40% of precision. It is built up from several Decision Trees. In each tree, there is a root node, containing all instances. The algorithm poses a question about the data, thus splitting the root node into

two decision nodes. These questions are always yes-no-type questions; Is the customer older than 25? Is the brand of the car Peugeot? Etc. All cases where the answer to the question was yes form one group, while all the cases where the answer was no form the other. The algorithm keeps splitting the data based on these yes-no questions until it is left with a homogenous group of instances, where almost all members of that group are labeled as one of the classes from the target variable (for example, “base” or “extra” in our case). Keeping in mind the mechanism of the decision trees, it should also be considered that all machine learning models in the scikit-learn python library, including the Random Forest Classifier, can only calculate with numeric data. Thus, all categorical/label variables must be encoded into numerical ones.

To deal with this problem encoders are used, which assign a certain value to each unique appearance of a categorical/label-type variable. One of these encoders is called one-hot-encoder. This encoder creates new dummy variables, each representing a unique value in the feature in question, so only one dummy variable at a time can have value of 1. Using one-hot-encoder for all non-numeric features would result in a huge dataframe because there are some non-numerical variables in the data with high number of unique values. This raises memory issues when one tries to perform any calculations on the whole dataset, such as fitting on a pipeline or machine learning model. This is a limitation of one-hot-encoder, it was not designed for labels. To deal with this problem, the label-encoder from the skit-learn/preprocessing python library is used. This encoder does not create extra variables, thus not increasing the memory usage, but instead, it assigns a number to each unique value in a variable, from zero to the number of unique values in the feature minus one. When the gathered information about the mechanism of the Decision Trees and label encoding are combined, it is possible to obtain a rather significant downside of using Random Forest Classifier in this specific case. When the classifier separates the data, it does it in a manner where it raises a question of whether the instances are greater or lower than a certain number. It puts all the instances that are greater in one group and all of the instances that are lower in the other. This wouldn’t cause a problem

with features such as age or dates, where the numbers in the feature are ordinal by nature.

However, in cases when values in a feature represent classes the algorithm of the decision tree faces some obstacles. When the algorithm raises the question of whether a value of a certain record based on a categorical variable encoded with a label encoder is smaller or larger than the differentiating number, it places all the records smaller than the number in one group and the rest into another one. In most cases, this is not what was desired. Take into consideration the following example. The 'COD_SEGMENTO_CLIENTE_MKT' variable is a unique description of client types in Fidelidade. The variable has 11 unique values, thus the corresponding label-encoded variable has the same number of unique values, the range of numbers from 0 to 10. The Table 11 contains all the classes in 'COD_SEGMENTO_CLIENTE_MKT' with their corresponding encoded forms.

Code	Description	Encoded Equivalent
A2A	Lost in Translation	1
A1P	Human Seeker	0
S1P	Brand Oldie	10
J2H	Rational	5
PN	Pequenos Negócios	9
A3H	Wealthy Investor	3
J1R	Remote Low Budget	4
PE	Pequenas Empresas	8
A2R	Online Price Grabber	2
ME	Médias Empresas	6
NN	Novos Negócios	7

Table 6 - COD_SEGMENTO_CLIENTE_MKT encodings

The classifier raises the question whether the value of 'COD_SEGMENTO_CLIENTE_MKT' is greater or smaller or equal to 3 for example. Splitting the data into two groups, one with clients who belong to Human Seeker, Lost in Translation, Online Price Grabber or Wealthy Investor client types (0,1,2 and 3), and another group with clients who belong to Remote Low Budget, Rational, Médias Empresas, Novos

Negócios, Pequenas Empresas and Brand Oldie (4,5,6,7,8,9 and 10). Whereas a separation with client types 0,4,6,9 and 1,2,3,5,7,8,10 might divide the data into more homogenous groups. However, such separation with the Random Forest Classifier is impossible, thus the model cannot make much use of non-numerical data, which hurts the predictive capabilities of the model. As a solution a condition was made, where all the non-numerical variables with more than 15 unique values in them were encoded using label encoder, and all other numerical values were encoded with one-hot encoder. This way the strength of each of these encoders is capitalized, resulting in a dataframe small enough to handle while also maximizing the use of categorical variables. The features considered most useful for classification by the model created with this hybrid pipeline can be seen in the table below (Table 12).

Features	Feature importance
Data_Matricula	0.060027
EUROTAX_TYLACODE	0.059792
Veiculo_Valor_Actual_Eurotax	0.049401
ID_CLIENT_ANONY	0.039757
QTD_ANOS_ANTIG_CLIENTE	0.039283
DT_INICIO_OBJETO	0.038430
DATE_AND_ID	0.036462
QTD_IDADE	0.035916
EUROTAX_PNOVO	0.034039
TAX_CRIME_CONDUCAO_ALCOOL_DICO	0.030418

Table 7 - Feature importance of Random Forest Classifier with the first approach.

Despite our efforts, no categorical features were considered as most important. The accuracy of the Random Forest Classifier is 42 percent. While at first look it seems rather disappointing, one must keep in mind that the model has to classify into 10 target classes, which means randomly assigning the data into certain classes would result in somewhere close to 10% accuracy. Thus, using this model is still four times better than using none. Looking at the confusion matrix (Figure 14), it tells us that, not so surprisingly, the model can classify much better into classes with high number of instances in them.

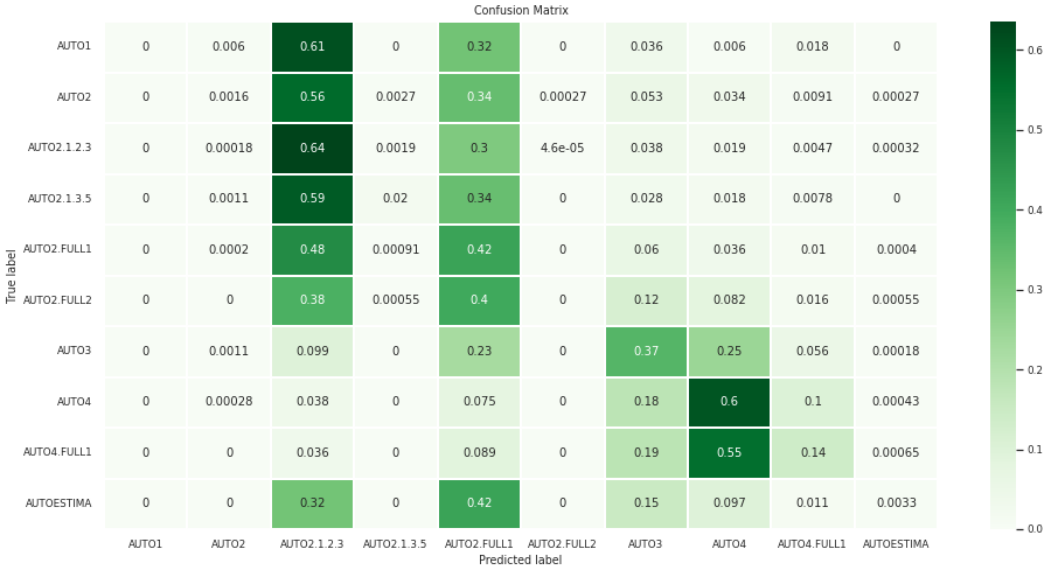


Figure 11 - Confusion Matrix of the Random Forest Classifier with target variables made by the method described in the first approach.

The confusion matrix has the true labels in the y axes and the predicted labels on the x axis. Each value in the matrix represents the percentage of time when a true label was predicted as a certain label (some other label or itself). As it was established earlier, each value in the matrix represents the percentage of the time when a true label was predicted as a certain label, thus the sum of each row in the matrix equals 1. For example, one can see that AUTO1 was never predicted correctly, and in 61% percent of the time it was predicted as AUTO2.1.2.3. What is more interesting is the AUTO2 class. While the model was almost never able to correctly classify into this category, it also never identified the members of this class as a member of another base insurance plan (such as AUTO3 or AUTO4), which tells us that

although the model does not have great results, it works better as the numbers suggest it at first look.

Diving into the approach of coverage package types, the overall performance can be seen in Table 13, just as before with first approach.

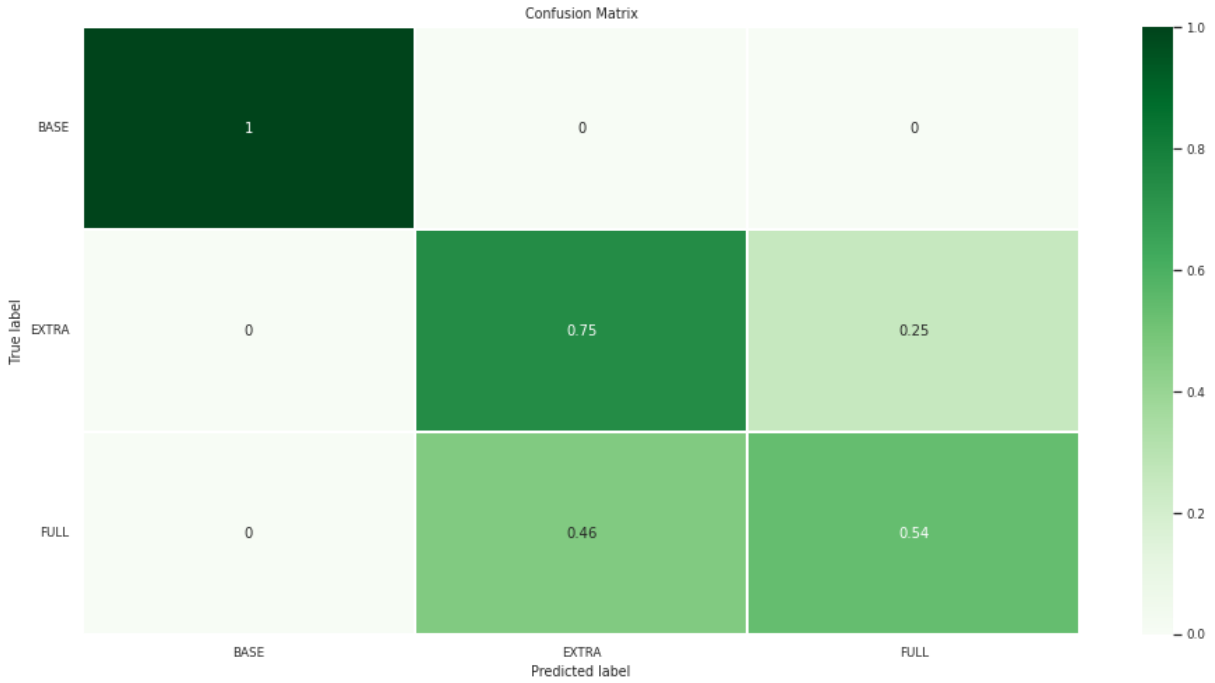
	Precision	Accuracy
Random Forest	0.63	0.63
LightGBM	***0.65***	***0.65***
Soft-max Regression	0.55	0.49
Naiv-Bayes	0.60	0.60

Table 8 - Overall evaulation of model performances with the approach of coverage package types

The model with the highest precision and accuracy is the LightGBM classifier in the case of the second approach. LightGBM is also a tree-based classifier (just as Random Forest) mixed with gradient boosting. The are many advantages of using LightGBM classifier, such as that unlike other tree-based algorithms LightGBM splits the tree leaf-wise which can lead to lower loss and higher accuracy, or that it uses GPU learning as well, which makes it appliable for working on huge dataframes with heavy memory (CPU) usage. (H. 2007) The disadvantage of this model is the same as for Random Forest's, namely that it is hard to make much use of categorical features with it, thus for the building of this model the same hybrid encoding technique was used as discussed before at the approach of extended insurance packages with Random Forest Classifier.

The precision and accuracy of our model are both 65%. At first glance, one might conclude that this approach yields better results than the one before, although one must keep in mind while evaluating the performance of this model, that now there are only three different categories to which our model has to assign records to. This means that running a model that classifies into these categories randomly it should achieve an accuracy of around 33.33%, if the

number of records is high enough. Thus, this model only works roughly two times better than randomly splitting into these categories which is way worse than the approach of extended insurance packages, where our model performed four times better compared to randomly classifying. However, jumping to conclusions before carefully evaluating all aspects of a result is never optimal. Thus, it would be useful to see how our model made its mistakes, with the



confusion matrix (Figure 15) just as before.

The first thing that jumps when looking at this confusion matrix, is that our model was able to classify into the base category with 100% accuracy. A 100% accuracy always means that there is some information in the data that absolutely correlates with the given variable/label. In other words, the value of some cells in the dataframes can determine whether the customer uses a base insurance package. In the Exploring Coverages part earlier in this section, it was stated that every customer except AUTO1 customers have at least one extra coverage included in their insurance plan, which explains the anomaly of why our model can classify into the base

Figure 12 - Confusion Matrix of the LightGBM Classifier with target variables made by the method described in the second approach.

class perfectly. Our second observation is that while the model could assign records to the extra class with quite a high accuracy, 75% percent, the same number for the full class is only 54%. This indicates that some of the records that should be labeled as extra are very similar to ones that should be full. If one recalls the way this approach was made, the reason behind the similarities between some cases of the extra class and the full class becomes clear. A record was labeled extra if it had at least one extra coverage but not all possible extra coverages. This means that the customers who choose AUTO 2 insurance plan, with just one extra coverage, for example, glass breaking coverage, were labeled as extra, just as another AUTO2 customer who had all the possible extra coverages provided by AUTO2 expect one. Meanwhile, in practice, these two customers probably have very different characteristics, thus our model cannot identify the differentiating characteristics between these two groups.

Because of this, an alternative version of the approach of coverage package types was inevitable to create. In this alternative approach (approach base-plus-extra-full), a fourth class, the plus class is introduced. The target variable is created in a way, that the base and full classes remain the same, but the extra class is spitted into two classes, plus and extra, where the plus class is the one with low number of extra coverages in an insurance package, and the extra class is the one with high number of extra coverages, but not all of them. The low and high number of extra coverages was determined manually on insurance package level. While AUTO2 offers more extra coverages then AUTO4 for example, it is not the same number that was considered to be low or high for these two cases. The most optimal way to find these limits for creating the “plus” and “extra” labels is the matter of future analysis, as it is not included in this paper. As the best performing model was the LightGBM and the Random Forest both with 55% precision and 56% accuracy, as it is an alternative approach, the results are not going to be discussed in such details as it was done in earlier approaches. However, an interesting observation worth mentioning in this case is the feature importance of the Random Forest Classifier. A cooperation

of the most important features of the Random Forest Classifier in the approach of coverage package types (approach 2) and the approach base-plus-extra-full (alternative approach) can be seen in Table 14**Error! Reference source not found.**.

Features	Feature importance	Features	Feature importance
Approach 2	**Approach 2**	**Alternative Approach**	**Alternative Approach**
Data_Matricula	0.043211	DSC_OPCAO_COB_AUTO3 -AUTO 3	0.114102
ID_CLIENT_ANONY	0.042553	DSC_OPCAO_COB_AUTO4 -AUTO 4	0.051309
DT_INICIO_OBJETO	0.041227	DSC_OPCAO_COB_AUTO2 -AUTO 2	0.050155
QTD_ANOS_ANTIG_CLIENTE	0.041149	DSC_I_ZONA_SISM_APS	0.046472
EUROTAX_TYLACODE	0.039570	Data_Matricula	0.036313
DATE_AND_ID	0.036889	EUROTAX_TYLACODE	0.034695
QTD_IDADE	0.036623	DSC_DISTRITO	0.034404
EUROTAX_PNOVO	0.034873	DSC_DISTRITO	0.034263
Veiculo_Valor_Actual_Eurotax	0.033771	Data_Matricula	0.032139
TAX CRIME CONDUCAO ALCOOL D			

Table 9 - Comparation of feature importance of the Random Forest Classifier.

Int the base-plus-extra-full approach the most important features were the ones that represented an insurance package category, such as AUTO3, AUTO4 or AUTO2, meanwhile the approach of coverage package types they were not even considered while splitting the tree. This is a huge advantage of the base-plus-extra-full approach, because the main idea behind creating this type of approaches was to make use of the results that are already available, namely the insurance package types, that would be already predicted while running this model.

4.3.5 Conclusion

Multiple ways of recommending more personalized coverage packages were introduced in this section in the name of flexibility. The best working model was the one that recommended the Random Forest Classifier with the first approach of creating target variables. However, the alternative approach shows interesting results, since it was possible to “force” the Random Forest model to use the insurance packages as an important feature when classifying into base, plus, extra, or full categories. Finding the optimal way to create these labels as target variable

could be a subject of further studies which might yield better results than the first approach.

While the focus in this section was on flexibility and to present multiple possible solutions of recommending coverage packages, each of this model could be further improved, with focusing on predictability and spending more time on tuning the hyper-parameters.

Please also note that these coverage package recommendation systems would not be able to satisfy the needs of the company as they are right now, although they are sufficient enough to be used as a decision-supporting system for agents. Especially if they recommend three-four different coverage packages to a customer that he can choose from, based on the confusion matrixes.

4. Limitations and Next Steps

Throughout the project, several limitations difficulted the path to achieve the best possible results. That was the case of not having enough RAM in the virtual machine to run a Grid search with more parameters to improve the models' metrics, and it was also the case of having an unbalanced dataset.

In the future, some work can be done to receive data from more customers with insurance packages that are less representative, and this way create a more balanced dataset.

Regarding the three initial objectives, they were all successfully anteing. Predictability, flexibility, and explainability were tested and thoroughly analyzed. There are several possibilities for Fidelidade to implement the findings of this work project:

i) Fidelidade can incorporate in their simulator more questions, by doing this, they can receive more data about the customer creating more personalized recommendations. In this scenario, Fidelidade should pay attention to the time it takes to complete the form since it is one important aspect to have in mind since most customers don't want to fill a form if it is too long.

ii) Fidelidade can pass to their agents the characteristics that are more important to well recommend an insurance package, this way agents can build more personalized offers to new customers.

iii) Fidelidade can build new offers, more detailed, and sell them. In this scenario, data should be gathered and in the future fed into new models.

5. References

Fidelidade. 2021. "Relatório e contas." https://www.fidelidade.pt/PT/a-fidelidade/QuemSomos/QuemSomos/Documents/Relatórios-2021/RC-RS_FIDELIDADE_20212.pdf.

2021. *Pordata.* https://www.pordata.pt/portugal/veiculos+rodoviarios+motorizados+em+circulacao+total+e+por+tipo+de+veiculos-3100-262483.

CGIP. 2021. "O melhor seguro automóvel." *Consumer Guidance Institute Portugal.*

Expresso. 2022. "5,6 milhões: em dez anos, nunca houve tantos carros em Portugal." *Expresso.* https://expresso.pt/sociedade/2022-01-15-56-milhoes-em-dez-anos-nunca-houve-tantos-carros-em-portugal.-como--porque--e-o-que-fazer-para-reverter-a-tendencia-.

Económico, Jornal. 2022. "Associação de seguradoras admite subida nos preços dos seguros." *Jornal económico.*

IBM. 2020. *Exploratory data analysis.* August 25. https://www.ibm.com/cloud/learn/exploratory-data-analysis.

UCLA. 2021. *WHAT IS THE DIFFERENCE BETWEEN CATEGORICAL, ORDINAL AND INTERVAL VARIABLES?* https://stats.oarc.ucla.edu/other/mult-pkg/whatstat/what-is-the-difference-between-categorical-ordinal-and-interval-variables/.

Urwin, Matthew. 2022. "Random Forest Classifier: A Complete Guide to How It Works in Machine Learning."

Wang, Hui Dong. 2019. "Research on the Features of Car Insurance Data Based on Machine Learning." *3rd International Conference on Mechatronics and Intelligent Robotics.*

Ke, Guolin. 2017. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *31st Conference on Neural Information Processing Systems .*

Verma, Suraj. 2021. "Softmax Regression in Python: Multi-class Classification." *towardsdatascience.*

Verma, Yugesh. 2021. "A Guide to Building Hybrid Recommendation Systems for Beginners." *DEVELOPERS CORNER.*

Korstanje, Joos. n.d. "The k-Nearest Neighbors (kNN) Algorithm in Python." *Real Python.*

Wikipedia. n.d. *Confusion Matrix.* https://en.wikipedia.org/wiki/Confusion_matrix, n.d.

Jolliffe, Ian T. 2016. "Principal component analysis: a review and recent developments." April 13. https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202.

Maaten, Laurens van der. 2008. "Visualizing Data using t-SNE." *Journal of Machine Learning Research 9 (2008) 2579-2605.*

H., Shi. 2007. *Best-first Decision Tree Learning.* Hamilton, New Zelannd: The University of Waikato.

6. Appendix

Coverages / Warranties	Auto 1	Auto 2	Auto 3	AUTOESTIMA	AUTO 4	2 RODAS
Civil Liability	❑ Mandatory Minimum	❑ Mandatory Minimum ❑ 50.000.000E	❑ Mandatory Minimum ❑ 50.000.000E	❑ Mandatory Minimum ❑ 50.000.000E	❑ Mandatory Minimum ❑ 50.000.000E	❑ Mandatory Minimum ❑ 50.000.000E
Travel Assistance	• Basic	• Plus	• Plus	• Plus	• Plus	• Plus
Legal Protection	-	❑ Level 3	• Level 3	• Level 3	• Level 3	• Level 1
Crash, Collision or Rollover	-	-	-	•	•	-
Fire, Lightning or Explosion	-	-	•	•	•	-
Theft or Robbery	-	-	•	•	•	-
Natural Phenomena	-	-	•	•	•	-
Acts of Vandalism	-	-	•	•	•	-
Isolated Glass Breaking	-	❑ 1.000€	• 1.000€ ❑ 2.000€	• 1.000€ ❑ 2.000€	• 1.000€ ❑ 2.000€	❑ 1.000€
Driver Protection (Death or Permanent Disability / Treatment Expenses)	• 10.000€ / 1.000€	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€	-
Driver's Life Protection	-	❑ 500.000€	❑ 500.000€	❑ 500.000€	❑ 500.000€	❑ 500.000€
Replacement Vehicle	-	-	❑ Level 1	❑ Level 1	• Level 1 ❑ Level 2	-
Vehicle Occupants (Death or Permanent Disability / Treatment Expenses)	-	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€	❑ 10.000€ / 1.000€ ❑ 25.000€ / 2.500€	-
Civil Liability Cargo	-	❑	❑	❑	❑	-
Acquisition value	-	-	-	-	❑	-
Secure Debt	-	-	-	-	❑	-

Table 10 - Fidelidade's current offer in detail and potential extra coverages.

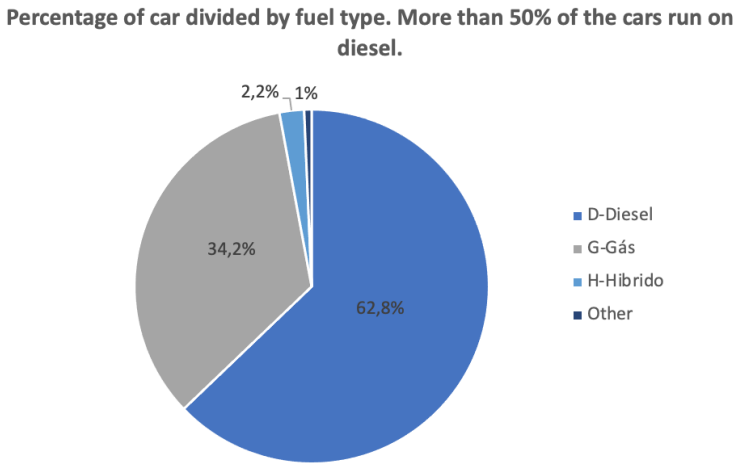


Figure 14 - Car percentage by fuel type.

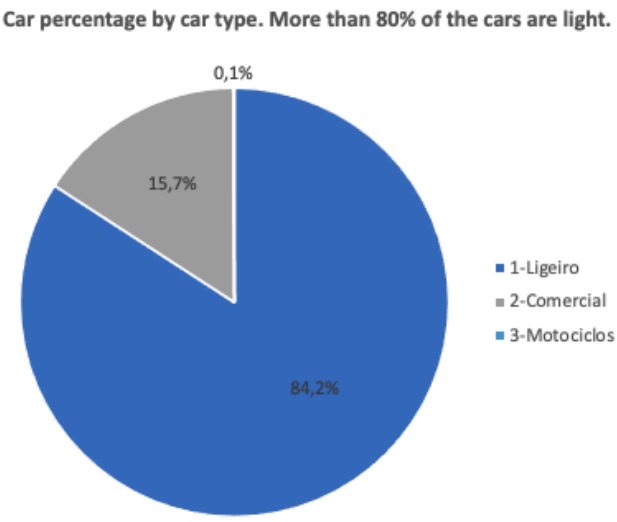


Figure 13 - Car percentage by car type.

Name	Description
AUTO1	basic auto1 package
AUTO2	basic auto 2 package
AUTO2.1	basic auto 2 with paying expenses for court (FLG_VEIC_COB_PROT_JUR_M0)
AUTO2.2	basic auto 2 with window coverage (FLG_VEIC_COB_QUEB_VID_M0)
AUTO2.3.	basic auto 2 with driver protection (FLG_VEIC_COB_PROT_CON_M0)
AUTO2.4	basic auto 2 with bigger capital driver protection (FLG_VEIC_COB_PROT_VIT_M0
AUTO2.5	basic auto 2 with protection of passengers (FLG_VEIC_COB_OCU_VIAT_M0)
AUTO2.1.2	basic auto 2 with paying expenses for court + window coverage
AUTO2.1.3	basic auto 2 with paying expenses for court + driver protection
AUTO2.1.4	basic auto 2 with paying expenses for court + bigger capital driver protection
AUTO2.1.5	basic auto 2 with paying expenses for court + passenger protection
AUTO2.2.3	basic auto 2 with window coverage + driver protection
AUTO2.2.4	basic auto 2 with window coverage + bigger cap driver protection
AUTO2.2.5	basic auto 2 with window coverage + passenger protection
AUTO2.3.5	basic auto 2 with driver protection + passenger protection
AUTO2.4.5	basic auto 2 with bigger capital driver protection + passenger protection
AUTO2.1.2.3	basic auto 2 with paying expenses for court + window coverage + driver protection
AUTO2.1.2.4	basic auto 2 with paying expenses for court + window coverage + bigger cap driver protection
AUTO2.1.2.5	basic auto 2 with paying expenses for court + window coverage + passenger protection
AUTO2.1.3.5	basic auto 2 with paying expenses for court + driver protection + passenger protection
AUTO2.1.4.5	basic auto 2 with paying expenses for court + bigger capital driver protection + passenger protection
AUTO2.2.3.5	basic auto 2 with winow coverage + driver protection + passenger protection
AUTO2.2.4.5	basic auto 2 with window coverage + bigger capital driver protection + passenger protection
AUTO2.FULL1	basic auto 2 with paying expenses for court + window coverage + driver protection + passenger protection
AUTO2.FULL2	basic auto 2 with paying expenses for court + window coverage + bigger capital driver protection + passenger protection
AUTO3	basic auto 3 package

Table 12 – Description of coverage packages part 1.

Name	Description
AUTO3	basic auto 3 package
AUTO3.1	basic auto 3 with driver protection (FLG_VEIC_COB_PROT_CON_M0)
AUTO3.2	basic auto 3 with bigger capital driver protection (FLG_VEIC_COB_PROT_VIT_M0)
AUTO3.3	basic auto 3 with replacement car (level1) (FLG_VEIC_COB_VEIC_SUB_M0)
AUTO3.4	basic auto 3 with passenger protection (FLG_VEIC_COB_OCU_VIAT_M0)
AUTO3.1.3	basic auto 3 with driver protection + replacement car
AUTO3.1.4	basic auto 3 with driver protection + passenger protection
AUTO3.2.3	basic auto 3 with bigger cap driver protection + replacement car
AUTO3.2.4	basic auto 3 with bigger cap driver protection + passenger protection
AUTO3.3.4	basic auto 3 with replacement car + passenger protection
AUTO3.FULL1	basic auto 3 with driver protection + replacement car + passenger protection
AUTO3.FULL2	basic auto 3 with bigger capital driver protection + replacement car + passenger protection
AUTOESTIMA	basic autoestima package
AUTOESTIMA.1	basic autoestima with driver protection (FLG_VEIC_COB_PROT_CON_M0)
AUTOESTIMA.2	basic autoestima with bigger capital driver protection (FLG_VEIC_COB_PROT_VIT_M0)
AUTOESTIMA.3	basic autoestima with replacement car (level1) (FLG_VEIC_COB_VEIC_SUB_M0)
AUTOESTIMA.4	basic autoestima wit passenger protection (FLG_VEIC_COB_OCU_VIAT_M0)
AUTOESTIMA.1.3	basic autoestima with driver protection + replacement car
AUTOESTIMA.1.4	basic autoestima with driver protection + passenger protection
AUTOESTIMA.2.3	basic autoestima with bigger capital driver protection + replacement car
AUTOESTIMA.2.4	basic autoestima with bigger capital driver protection + passenger protection
AUTOESTIMA.3.4	basic autoestima with replacement car + passenger protection
AUTOESTIMA.FULL1	basic autoestima with driver protection + replacement car + passenger location
AUTOESTIMA.FULL2	basic autoestima with bigger capital driver protection + replacement car + passenger location
AUTO4	basic auto 4 package
AUTO4.1	– basic auto 4 with driver protection
AUTO4.2	basic auto 4 with bigger capital driver protection
AUTO4.3	basic auto 4 with passenger protection
AUTO4.FULL1	basic auto 4 with driver protection + passenger protection
AUTO4.FULL2	basic auto 4 with bigger capital driver protection + passenger protection

Table 11 – Description of coverages packages part 2.