NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Machine Learning Approach for Personalized Recommendations on Online Platforms – Uniplaces Case Study

Marta Maria Cabral Menéres Posser Villar

Project Work presented as partial requirement for obtaining the Master's degree in Information Management

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

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

MACHINE LEARNING APPROACH FOR PERSONALIZED RECOMMENDATIONS ON ONLINE PLATFORMS –UNIPLACES CASE

by

Marta Maria Cabral Meneres Posser Villar

Project Work presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Knowledge Management and Business Intelligence

Advisor / Co-supervisor: Mauro Castelli

November 2020

ABSTRACT

The goal of this project is to develop a model to personalize the user recommendations of an online marketplace named Uniplaces. This online business offers properties for medium and long-term stays, where landlords can directly rent their place to customers (mainly students). Whenever a student makes a reservation, the booking must be approved by the property owner. The current acceptance rate is 25%. The model is a response to this low acceptance rate, and it will have to show to each student the properties that are more likely to be accepted by the landlord. As a secondary objective, the model seeks to identify the reasons behind the landlord's decision to accept or reject bookings.

The model will be constructed using information from the users, landlord and the property itself kindly provided by Uniplaces.

This information will pre-process with data cleaning, transformation and features reduction (where two techniques were applied: dimensionality reduction, features selection). After the data processing, several models were applied to the normalized data. The predictive models that will be applied are already being used on other online markets and platforms like Airbnb, Netflix or LinkedIn, namely Support Vector Machine, Neural Networks, Decision Tree, Logistic Regression and Gradient Boosting.

The probability of acceptance proved to be very easy to predict, all the models predict 100% of the test dataset when using the Principal Component Analysis as the Dimensionality Reduction technique. This can be explained mainly by the fact that the new calculated features have a strong correlation with the target variable. All the algorithms predict 100% of the target variable when using Principal Component Analysis as a technique of dimensionality reduction.

KEYWORDS

Machine Learning, Data Mining; Predictive Models; Supervised Learning

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS AND ACRONYMS

ASM Attribute Selection Measure

NN Neural Networks

ML Machine Learning

DT Decision Tree

SVM Support Vector Machine

1. INTRODUCTION

In recent decades, with the expansion of the reach of the world wide web, a new business model has been growing more prevalent and has disrupted numerous industries worldwide and is highly likely to continue to do so in the future: the digital marketplace, where a company will simply provide a platform where consumers can search for third parties wishing to provide a particular service or good. The structure of this model creates value for all three parties involved: for the consumer, who now has a wide range of options to choose from, additional information and transparency which can now be used to make the purchasing decision and seek generally lower prices; for the third-party provider, who is now able to reach to additional clients he would not have had the chance to without the platform; and the company providing the market place keeps a portion of all the transactions done through the platform.

The ability to generate transactions by matching each consumer with the right vendor, a key tool towards in achieving this aim are the personalized recommendations. This technique is used across multiple platforms, having a growing impact on the business they are being applied to. When Airbnb launched its algorithm, the overall profitability of the platform increased by 0,75%. LinkedIn´s recommendation algorithms have increased not only the relevance of the candidates displayed whenever an employer makes a query, but also the probability that those candidates would be interested in a position with the employer. Netflix reports that 75% of what people watch on their platform is from some sort of recommendation. These are some examples that are further explored in this report, including what machine learning algorithms each of these companies have been using.

The second part of this project will focus on building a personalized recommendation algorithm for a company, that might help their business performance. The company in question is Uniplaces.

Uniplaces was founded in Portugal in 2012 by two former students of King´s College in London. With their first-hand knowledge of the difficulties student face when searching for accommodation away from home, they decided to create a platform through which students could directly contact owners of rooms, flats or homes who intended to rent these to students. The website was initially launched with accommodation in Lisbon, shortly after becoming a success, earning over 25 M €. It is now present in 8 European countries and it has booked more than 7 million nights since its foundation. Originally, the site was aimed only at students, but now any traveler can use it.

The main objective of this project is to increase the acceptance rate of the platform. Currently, only 25% of all requests for bookings are accepted by the landlord, the rest either expire or are rejected. The platform does not apply any kind of recommendation at the moment, so this project will develop an algorithm to recommend to users the properties that landlords are more likely to rent to them.

Another goal of this study is to identify what variables most influence the landlord when accepting or rejecting a booking.

To achieve these goals, seven models will be applied to Uniplaces' dataset that contains information pertaining to the proprietor, the student and the property itself. In the end the models with be compared with the recommendations from other online marketplaces (Leino & Räihä, 2007).

2. LITERATURE REVIEW

Learning how to rank for information retrieval has attracted a lot of interest in recent years. It has become the main challenge for online platforms: what information to provide for each user. It is very important for an online platform to accurately tune their ranking functions as it directly affects the search experience of millions of users (Chapelle & Chang, 2011).

There are multiple examples of how online platforms deal with their information ranking needs, the approach used as well as the effect their efforts has had on their business performance.

Airbnb, the online temporary housing marketplace, runs an algorithm to maximize the probability of renting a certain location, according to host and guest preferences. The platform deploys this algorithm to match queries with the locations that guests are most likely to book, according to their patterns and queries (Dolnicar, 2018). The transition of this algorithm to deep learning in the evolution of search ranking was a big step for Airbnb - it significantly increased bookings with a simple algorithm (Haldar, et al., 2020).

When Haldar and coauthors (Haldar, et al., 2020) built the first Airbnb's algorithm draft, they noticed that lower priced listings were closer to guests' preferences, with a preference for the more economical listing. However, it would not make sense for Airbnb to simply recommend the cheapest properties to users, so they decided to remove price as an input feature to the DNN. When the algorithm was released, the result was a drop of 2.3% in the average price of search results. Although, the increase in bookings more than offset the effect of the price drop on revenue, resulting in an overall increase of 0.75% (Haldar, et al., 2020).

An additional example can be seen in the efforts of the professional social network LinkedIn, a platform that has been designed to connect job providers and job seekers. LinkedIn serves as a marketplace for efficiently matching potential candidates and job openings. A key mechanism to help achieve these goals is the LinkedIn Recruiter product, which enables recruiters to search for relevant candidates and obtain candidate recommendations for their job postings. The goal is to determine a ranked list of the most relevant candidates in real-time among hundreds of millions of candidate profiles. The algorithm requires not just that a candidate shown must be relevant to the recruiter's query, but also that the candidate contacted by the recruiter must show an interest in the job opportunity. It uses metrics from the recruiter and from the candidate (e.g. the likelihood of a candidate receiving a message from the recruiter and also the likelihood of a positive response from the candidate) (Guo, et al., 2019).

The first model used by LinkedIn Recrutier for search ranking was a linear model, but since this could not capture non-linear features, it was replaced by the Gradient Boosted Decision Trees (GBDT). The latter model was an improvement on the original one because it proved to work better with feature co-linearity, handling features with different ranges and dealing with missing values. However, it proved that several challenges, such as GBDT did not work well with sparse id features (such as skill id, etc.) and it demonstrated to be an algorithm with insufficient flexibility in terms of design and model specification. In order to overcome these challenges, a Neural Network was explored, but it proved less efficient than GBDT (Guo, et al., 2019).

We can also discuss the case of the online retailer Amazon: "Any recommender system needs some input to generate personalized recommendations that are not simply based on the item popularity"

Juha leino and Kari-Jouko Raiha studied user behavior in a recommendation environment, through the online marketplace Amazon. The results underline that the way recommendations are shown affect which items get picked. While the keyword search was still the most common approach to finding products from a large number of possibilities, recommender systems played an important part in helping users find the books they wanted. The algorithm of recommendations helped the participants to find three out of seven purchases/items. The study concluded that Amazon uses two types of input: first, the user's long-term engagement with the site and second, the user's current activities (Leino & Räihä, 2007).

To satisfy the need for efficient and effective information ranking solutions, there are companies that are now seeking advantages through the academic community and by organizing competitions. In 2006, Netflix announced a machine learning competition for movie rating prediction – the Netflix Prize. The final team reported a combination of 107 algorithms, such as Singular Value Decomposition and Restricted Boltzmann Machines (RBM). Over the years, the platform discovered the huge impact that recommendation systems have on user activity, in such a way that 75% of what people watch is based on some form of recommendation. Therefore, personalization became a focus on Netflix: the home page consists of rows of films; the order of the rows is personalized, as well as the order in which the movies are presented in each category. Then, of course, the recommended movies are based on what has previously been watched (Zhou, Wilkinson, Schreiber, & Pan, 2008).

An obvious baseline for a ranking recommendation is popularity. However, popularity is the opposite of personalization. The goal of Netflix recommendations is to find a ranking that considers these two features (Zhou, Wilkinson, Schreiber, & Pan, 2008).

According to Netflix, the typical choices for supervised classification methods are Logistic Regression, Support Vector Machines, Neural Networks or Decision Tree-based methods such as Gradient Boosted Decision Trees (GBDT), RankSVM or RankBoost. Beyond these algorithms, other algorithms should be considered when working with personalization rankings, as Linear Regression, Elastic nets, Singular Value Decomposition, Restricted Boltzmann Machines, Markov Chains, Latent Dirichlet Allocation, Association Rules, Random Forests, Clustering techniques from simples k-means to novel graphical approaches such as Affinity Propagation and Matrix factorization (Zhou, Wilkinson, Schreiber, & Pan, 2008).

A second example of the use of competitions to optimize ranking algorithms can be found with Yahoo!.com. In March 2010, Oliver Chapelle and Yi Chang organized the Yahoo! Learning to Rank Challenge, a challenge to encourage the research community to develop new learning to rank algorithms. There were two tracks in the challenge: a standard learning to rank track and a transfer learning track where the goal was to learn a ranking function for a small country by leveraging the larger training set of another country. Some major findings in this event were:

- Decision trees were the most popular class of function among the top competitors;
- Ensemble methods, including boosting, bagging and random forests, were dominant techniques;
- The differences in accuracy between the winners were very small.

The results of the challenge clearly showed that non-linear models such as trees and ensemble learning methods are powerful techniques. It was also surprising to notice that the relevance difference among the top winners is negligible, suggesting that the existing solutions to the ranking problem are quite mature and that the research on learning to rank should now go beyond the traditional setting that this challenge considered (Chapelle & Chang, 2011).

All of these examples of algorithm used to improve information ranking and user recommendations highlight the importance that data and solutions for processing it have assumed in the modern business landscape: "Data has an important and unique role to play in modern civilization: in addition to its historic role as the raw material of the scientific method, it has gained increasing recognition as a key ingredient of modern industrial and business engineering." (Ville, 2006).

2.1. MACHINE LEARNING

"The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience" (Mitchell, 1997).

In recent years, the application of machine learning methods has become very common in everyday life. When we look at a complex website like Facebook, Amazon or Netflix, it is very likely that every part of the site contains multiple machine learning models (Mueller & Guido, 2016). In the second half of the 20th century, this application evolved as a subfield of Artificial Intelligence (AI) transforming data into algorithms for making predictions. Instead of requiring humans to analyze large amounts of data, machine learning proposes a more efficient alternative for extracting conclusions from large amount of data, through a machine that gradually improves its performance (Raschka & Mirjalili, 2019).

There are three types of machine learning models: supervised learning, unsupervised learning and reinforcement learning. In supervised learning, we know the target variable, and in reinforcement learning, a measure of reward is defined for a particular action performed by the agent. However, in unsupervised learning we are dealing with unlabeled data and the target value is unknown (Raschka & Mirjalili, 2019).

The image 1 describes the typical workflow of a predictive model:

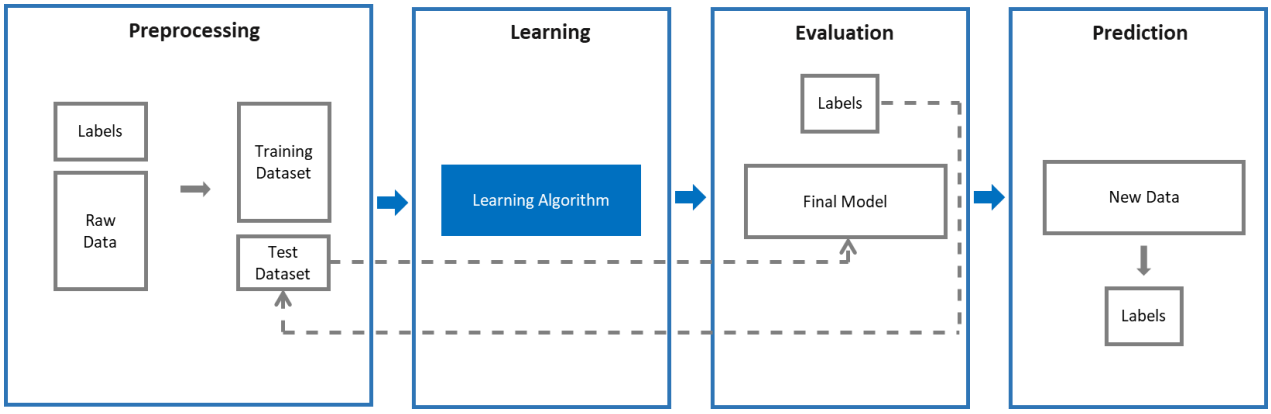

Figure 1: Predictive learning workflow (Raschka & Mirjalili, 2019)

The first step is called Pre-processing and it's one of the most crucial steps. It includes all kind of transformations in the data set, some algorithms for examples require that the selected features are on the same scale to achieve the optimal performance of the model, others require that the features are not highly correlated. Several techniques are used in this phase to clean the data set, such as dimensionality reduction that can improve the prediction performance of a model with a large number of irrelevant features. To analyze the performance of the algorithm, we randomly divide it into a test data set and training data set (Raschka & Mirjalili, 2019).

There is no perfect model, so it is essential to compare at least a handful of different algorithms in order to train and select the best performing model. To do so, there are several performance metrics that can be used to measure a model's relevance. One example is the confusion matrix, a square matrix that counts the number of cases that were well or poorly predicted.

The target is 1 when a data point belongs to a certain class C, 0 otherwise.

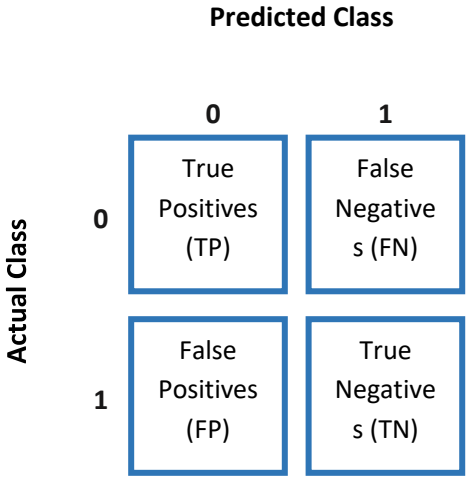

Figure 2: Confusion Matrix

Where,

TP is the number of elements belonging to C and that are classified as C
TN is the number of elements that don't belong to C and that are not classified as C
TP is the number of elements that don't belong to C but are classified as C
PN is the number of elements belonging to C and that are not classified as C

Using the information from the Confusion Matrix, we can calculate several useful metrics (Raschka & Mirjalili, 2019).

Measure	Formula
Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$
Error	$ERR = \frac{FP + FN}{TP + TN + FP + FN}$
Precision	$PRE = \frac{TP}{TP + FP}$
True Positive Rate (TPR)	$TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$
False Positive Rate (FPR)	$TPR = \frac{FP}{N} = \frac{FP}{FP + TN}$
Recall	$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$
F-score	$F1 = 2 * \frac{PRE * REC}{PRE + REC}$

Table 1: Evaluation Measure

Another technique often used to select the best classification model based on their performance is the Receiver operating characteristic (ROC) curve. The curve is created by plotting the TPR (True positive rate) against the FPR (False Positive Rate) at various values of the threshold. The perfect classifier would fall into the top-left corner of the graph with TPR = 1 and FPR = 0. The diagonal represents a random guessing meaning that a curve below the diagonal would represent a model worse than random guessing. Based on the ROC curve, the area under the curve (AUC) is often used to classify the performance of the model (Raschka & Mirjalili, 2019).

Figure 3: ROC Curve

One of the key steps in building a machine learning model is to estimate its performance on data that the model hasn't seen before. The model can suffer from underfitting (if the model is too simple), or from overfitting (if the model is too complex for the dataset).

In order to estimate the model's generalization performance, there are some cross-validation techniques such as holdout cross-validation and k-fold cross-validation. The holdout method split the dataset into training and test datasets – the test is used for model training and this training is used to estimate its general performance. In the k-fold cross validation, the data is divided in K sets and data splitting is performed K times. Each time, one of the K sets is used as a test set and the other sets are used as training sets.

2.2. FEATURE PRE-PROCESSING AND SELECTION

Variable and feature selection have become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available. In the past few years, most papers explore domains with hundreds to tens of thousands of variables or features (Guyon & Elisseeff, 2003).

Isabelle Guyon and André Elisseeff illustrate several benefits of variable and feature selection: facilitating data visualization and data understanding, reducing measurement and storage requirements, reducing training and utilization times and defying the curse of dimensionality to improve prediction performance (Guyon & Elisseeff, 2003).

Several feature selection techniques have been proposed in the machine learning literature. They can be divided into three types of methods: Filter methods, Wrapper methods and Embedded methods (Srinivasan & Pavya, 2017):

Filter approach

The most commonly used filter approaches are ranking techniques. A certain score is assigned to each variable using a classification criterion, and the variables with the lowest score are removed. The advantage of these methods is that they are computationally cheaper and avoid overfitting. On the other hand, filter methods ignore dependencies between variables, so the subset obtained might be redundant and not optimal.

Some of the basic filter techniques are as follows:

- Correlation Criteria: only detect linear dependencies between variable and target. The Pearson correlation is the simplest one:

$$\mathrm{R(i)} = \frac{cov(x_i, Y)}{\sqrt{var(x_i) * var(Y)}}$$

Equation 1: Pearson Correlation

  Where, x_i is the variable, Y is the output class, $var()$ is the variance and $cov()$ is the covariance

- Correlation-based Feature Selection (CFS): selects variables measuring the usefulness of each one for predicting the output variable and the level of inter-correlation between them. Thus highly correlated and irrelevant features are avoided.

- Fast Correlation-based Feature Selection: starts with a full set of features. Then, it calculates dependences of variables and finds the best subset of variables using the backward selection. The input variables are analyzed and ordered depending on a relevance score and it discards irrelevant variables; then, selects predominant features from the relevant set obtained before. It uses entropy and conditional entropy values to calculate the dependencies of features - the value 0 indicates that two features are totally independent and value of 1 indicates that using one feature, the other feature's value can be totally predicted.

Wrapper approach

These methods are better used to find optimal features rather than just relevant features. They use backward elimination to remove the insignificant variables from the dataset. These methods need predefined learning algorithms to identify the relevant feature. The Wrapper approach uses cross validation to avoid overfitting. Some algorithms that use this approach are Sequential Selection Algorithms and Heuristic Search Algorithms.

Embedded approach

The embedded method is a combination of the filter and wrapper methods – a feature selection method is incorporated into the learning algorithm and optimized for it. Embedded methods reduce the computation time of the wrapper methods.

An alternative approach to feature selection for dimensionality reduction is summarize the information content of a dataset by transforming it into a new feature subspace of lower dimensionality than the original one – feature extraction (Raschka & Mirjalili, 2019).

A very common technique of feature extraction is called Principal Component Analysis (PCA). This method identifies patterns in the data based on the correlation between the variables. It finds the directions of maximum variance in high-dimensional data and projects the data into a new space with equal or less dimensions than the original (Raschka & Mirjalili, 2019).

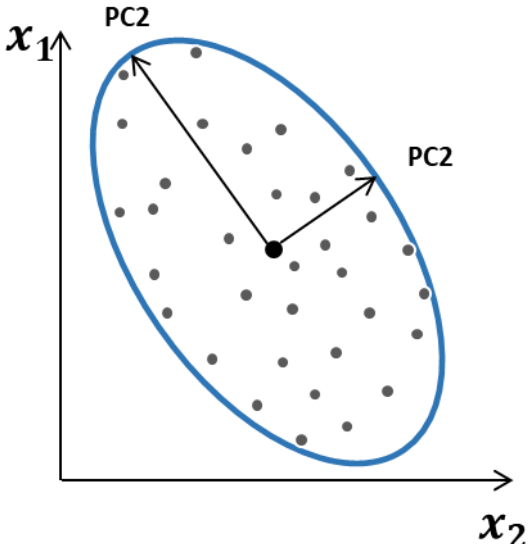

Figure 4: Principal Component Analysis (PCA)

Where, $x_1$and x_2 are the original features axes, and PC1 and PC2 are the principal components.

2.3. PREDICTED MODELS

2.3.1. Logistic Regression

Logistic regression is a classification model that is very easy to implement and performs very well on linear separable classes. It is one of the most commonly used algorithms for classification problems (Raschka & Mirjalili, 2019).

It is a special case of linear regression because it uses a log of odds as the dependent variable. The odds can be written as (Raschka & Mirjalili, 2019).

$$odds = \frac{p}{(1-p)}$$

Equation 2: Odds

where, p represents de probability of the positive.

The logit function, being the logarithm of the odds, is defined as:

$$logit(p) = log\frac{p}{(1\ -\ p)}$$

Equation 3: Logit function

The logit function has two possible values for the dependent variable, in this problem 0 if the landlord rejects the booking request, 1 if the landlord accepts the booking request.

Using the logistic regression model, we are interested in predicting the probability that a certain example belongs to a class, that is the inverse of the logit function, called the sigmoid function:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Equation 4: Sigmoid Function

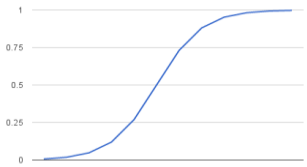

Figure 5: Sigmoid Function

Where, z is the input, the linear combination of weights and the inputs (the features associated with the training examples):

$$z = w_0x_0 + w_1x_1 + \cdots + w_nx_n$$

2.1.1. Support Vector Machine (SVM)

SVMs, first introduced by Vladimir Vapnik, are a type of linear learning machine much like the famous perceptron algorithm, and thus function to classify input patterns by first being trained on labeled data sets (supervised learning). However, SVMs represent a significant enhancement in functionality over perceptron's. The power of SVMs lies in their use of non-linear kernel functions that implicitly map input into high dimensional "feature spaces." In feature spaces linear classifications may be possible, which become non-linear in the transformation back to the original input space. Thus, although SVMs are linear learning machines with respect to feature spaces, they are in effect non-linear classifiers (Taylor & Cristianini, 2000)

Support Vector machines realize the following idea: map a n-dimensional input vector into a high dimensional feature space and construct an optimal separating hyperplane in this space. Different mappings construct different SVMs.

When the training data is separable, the optimal hyperplane is the one with the maximal distance between the hyperplane and the closest image of the vector from the training data. For non-separable training data, a generalization of this concept is used.

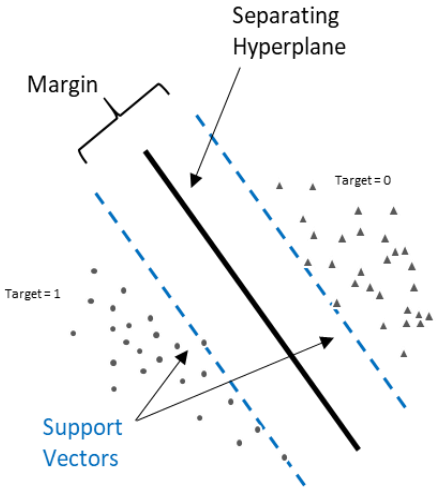

Figure 6: Support Vector Machine – linear problem

Linear problems can be generalized in order to solve non-linear problems. However, there are many cases in which it is not possible to divide the training data satisfactorily by a hyperplane. An example is shown in Figure 7, in which the use of a curved boundary would be more appropriate for class separation (Lorena & Carvalho, 2007).

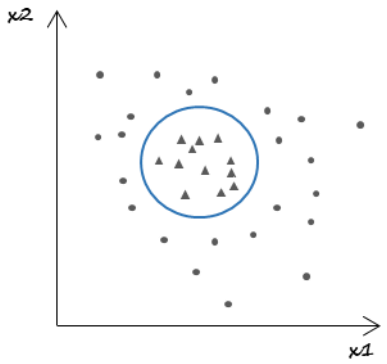

Figure 7: Support Vector Machine – non-linear problem

The basic idea of the SVM to solve non-linear problems is to transpose the original data (input space) into a more dimensional space, called the feature space, with the help of real functions ($\phi 1$,..., ϕm) defined in training data space. Using the kernel functions, it is possible to adjust the margin maximization in a space of greater dimensionality. The most commonly used kernel functions are shown in Table 2.

Kernel type	Function $k(x, y)$
Polynomial kernel	$k(x, y) = (x.y + 1)^d$
Gaussian kernel	$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$
Gaussian radial basis function (RBF)	$\mathrm{k}(\mathrm{x}, \mathrm{y}) = \exp(-\gamma\|x - y\|^2)$
Sigmoid kernel	$\mathrm{k}(\mathrm{x}, \mathrm{y}) = \tanh(\alpha x^T y + c)$

Table 2: Most common Karnel Functions (Lorena & Carvalho, 2007).

2.1.2. Decision Tree

Decision trees are a class of data mining techniques that have roots in traditional statistical disciplines such as linear regression. Decision trees also share roots in the same field of cognitive science that produced neural networks (Neville, 1999).

A Decision Trees is, as the name says, a decision-making tree for dividing data into groups. The first rule divides the entire data set into pieces, and then another rule can be applied to a piece, different rules to different pieces, forming a second generation of pieces. In general, a piece can be divided or left alone to form a final group (Neville, 1999).

The leaves of the tree are the final groups, the unsplit nodes. For some reason, trees are always drawn upside down, like an organizational chart. For a tree to be useful, the data in a leaf must be similar with respect to some target measure, so that the tree represents the segregation of a mixture of data into purified groups (Neville, 1999).

Figure 8 represents a Decision Tree. Internal nodes represent a decision rule, the branch represents a decision, and each leaf node represents the output.

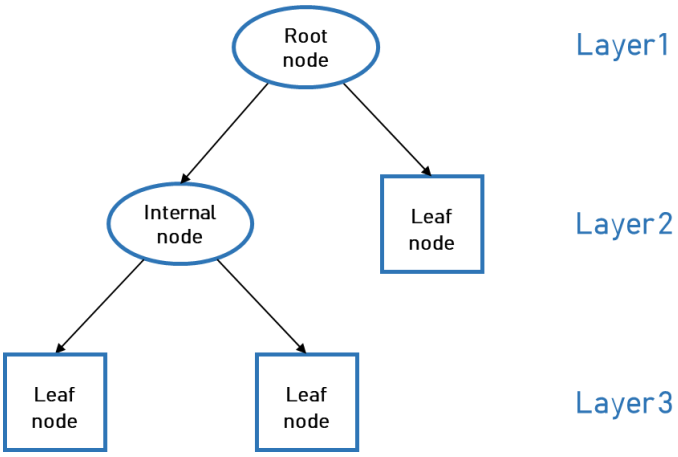

Figure 8: Decision Tree

Decision trees produce results that communicate very well in visual terms. It is easy to produce, easy to understand, and easy to use it. It can include qualitative and quantitative measurements. Decision trees readily adapt to various twists and turns in data—unbalanced effects, nested effects, offsetting effects, interactions and non-linearities—that frequently defeat other one-way and multi-way statistical and numeric approaches (Ville, 2006). Decision trees can handle high dimensional data. In general, this method offers good accuracy (Aruna & Nirmala, 2013)

The key requirements to use decision trees are (Aruna & Nirmala, 2013):

1) Attribute-value description: object or case must be expressible in terms of fixed collection of properties or attributes (e.g. hot, mild, cold)

2) Predefined classes (target attribute values): The categories must have discrete output values (supervised data).

3) Discrete classes: A case does or does not belong to a particular class, and there must be more cases than classes.

4) Sufficient data: enough training cases should be provided to learn the model, usually hundreds or even thousands of training cases.

Attribute Selection Measures

Attribute selection measures are used to select the best criteria to split the data in the best possible way. ASM provides a rank for each feature by explaining the dataset. The best score attribute is selected as splitting attribute (Aruna & Nirmala, 2013) . There are several selection measures proposed by the authors, three of them are Information Gain, Gain Ratio and Gini Index.

- Information Gain

Claude Shannon invented the concept of Entropy, studying the value or "information content" (Shannon 1948). The author refers to entropy as the impurity in a group of examples. The objective of this measure is to identify the attribute with the highest discriminating power. Thus, an entropy is maximum (entropy=1) when an impurity is maximum, meaning that the probability of an individual belonging to a class j is equal to 0,5. On the other hand, entropy is minimal when all those individuals belong to the same class.

The entropy of a split is found by computing the entropy in each of the leaves and by summing the entropy of all the leaves of a split. The variability of an outcome in a leaf is computed using the formula $-log2(p_i)$. The sum entropy of all the leaves of a split is

$$-\sum_{i=1}^{v} p_i \, log_2(Pi)$$

Equation 5: Entropy

Where,

p_i is the proportion of a particular class,

i in the collection of categories contained in the branch.

Information Gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. It measures how well a given attribute separates the training examples according to their target classification.

Information Gain is generally used in the algorithm C4.5 (an improvement of ID3) (Neville, 1999).

The information Gain is calculated with the difference between the original information requirement (i.e. based on the classes) and the new requirement (i.e. obtained after portioning on A) (Aruna & Nirmala, 2013).

$$\mathrm{Info}_A(D) = \sum_{j=1}^{V} \frac{|Dj|}{|D|} * \mathrm{Info}(D_j)$$

$$\mathrm{Gain(A)} = \mathrm{Info(D)} - \mathrm{Info_A(D)}$$

Equation 6: Information Gain

Where,

Info(D) is the average amount of information needed to identify the class label of a tuple in D,

$\frac{|Dj|}{|D|}$ in the weight of the j partition,

$\mathrm{Info_A(D)}$ is the expected information required to classify a tuple from D based on the partitioning by A.

The attribute A with the highest information gain, Gain(A), is chosen as the splitting attribute at node N().

- Gain Ratio

Information Gain is biased for the attribute with many outcomes – it prefers the attribute with a large number of distinct values.

The Gain Ratio is an extension of the information gain since it handles the issue of bias by normalizing the information gain.

$$SplitInfo_A(D) = -\sum_{j=i}^{v} \frac{|Dj|}{|D|} log_2\left(\frac{|Dj|}{|D|}\right)$$

$$GainRatio(A) = \frac{\text{Gain(A)}}{SplitInfo_A(D)}$$

Equation 7: Gain Ratio

Where,

$\frac{|Dj|}{|D|}$ is the weight of the j partition,

V is the number of discrete values in attribute A.

The attribute with the highest gain ratio is chosen as the splitting attribute (Nair, Mohandas and Sakthivel, 2010).

- Gini Index

Gini index, introduced for the first time by the economist Corrado Gini, is mostly used in the algorithm CART (Classification and Regression Tree).

This measure provides a simple method of quantifying the deviation from a uniform distribution, and it also has the advantage that its value is a number in the [0, 1] interval, like an order parameter. The Gini index is 0 when all members of the investigated society are equal in the relevant quantity, and it is 1 if one member is monopolizing the whole of the available resources (Néda & Biró, 2020).

The computation of the Gini measure for a set of objects with J classes is:

$$Gini = 1 - \sum_{i=1}^{J} {p_i}^2$$

Equation 8: Gini measure

where,

pi is the proportion of objects in which the target class is equal to i.

2.1.3. Neural Network (NN)

Neural Networks (NN) are computer systems with interconnected nodes that work as the neurons of the human brain. This first concept was introduced by Warren McCulloch and Walter Pitts, the so-called McCulloch-Pitts (MCP) neuron in 1943. A few years later, based on the MCP neuron model, Frank Rosenblatt proposed an algorithm that would automatically learn the optimal weight (w_n) of each neuron and then multiplied with the input values (x_n) in order to predict if a new data value belongs to one class or another (in case of a classification problem) (Raschka & Mirjalili, 2019).

Assuming the context of a binary classification problem, we can define an activation function $\phi(z)$ that takes the linear combination of the inputs (x_n) and the corresponding weights (w_n).

$$z = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n$$

Equation 9: Net Input Function

If the net input is greater than a defined threshold, θ , it is predicted that the point belongs to class 1, otherwise to class -1. This corresponds to the activation type Threshold.

$$\phi(z) = \begin{cases} 1, & if\ z \geq \theta \\ -1, & otherwise \end{cases}$$

Equation 10: Activation Function: Threshold

In machine learning, a negative threshold or weight is usually called the bias unit.

The activation function $\phi(z)$ is needed for hidden layers of the NN to introduce non-linearity. Without it, NN will be the same as perceptions. Some activation functions examples are linear, threshold and sigmoid function.

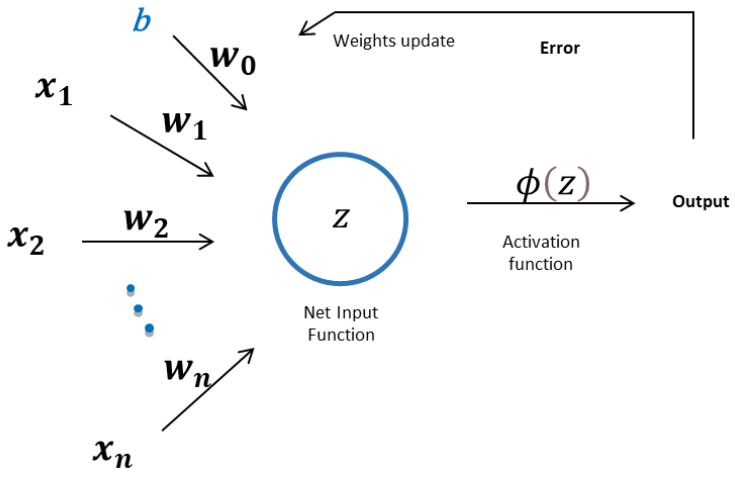

Figure 9: Operations done by a neuron

A Neural Networks is a representation of the human brain, neurons interconnected to other neurons usually represented in columns. The first column is represented by the input neurons, the ones in the middle are the hidden layers, and the last one is the output layer.

The neurons are connected by links. Each link has its own weight and every neuron consists of more than one weight as well as the adjusted weight. From input neuron, links will move towards the other nodes. This is known as feed-forward (FF) neural network. Each link has a numeric and associated weight that connect it through the output. (Farizawani, Puteh, & Rivaie, 2019).

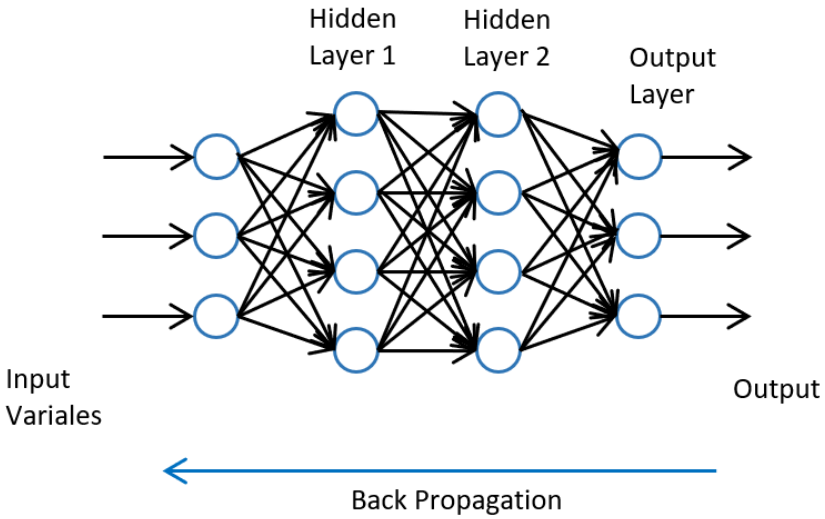

Figure 10: Neural Networks

The number of the layers usually depends on the complexity of a problem to be solved. The work of Haldar and coauthors (Haldar, et al., 2020), presented that they started to build Airbnb's search algorithm using DNN with several layers. Soon they understood that adding layers met nothing but neutral test results. The conclusion of the exercise was that increasing layers was an effective technique for convolutional neural networks, but not necessarily for all Depp Neural Networks. For fully connected networks, the authors defend that two hidden layers are enough (Haldar, et al., 2020).

The weights of each observation that is submitted to the network are adjusted according to the difference between the expected output and the output obtained. This is called the back-propagation, where the weights are adjusted in the opposite direction, that is, from the output layer to the input layer (Rojas, 1996).

The algorithm is stopped when the value of the error function has become sufficiently small. This is a very rough and basic formula for back-propagation algorithm. There is some variation proposed by other scientists, but Rojas definition seems to be quite accurate and easy to follow. The last step, weight updates is happening throughout the algorithm (Rojas, 1996).

Rojas claimed that back-propagation could be broken down into four main steps. After choosing the weights of the network randomly, the back-propagation algorithm is used to compute the necessary corrections. The algorithm can be decomposed in the following four steps: (1) Feed-forward computation, (2) Back-propagation to the output layer, (3) Back-propagation to the hidden layer and (4) Weight updates (Cilimkovic, 2015).

According to Farizawani and coauthors (Farizawani, Puteh, & Rivaie, 2019), there are four main steps in neural networks used in the classification task: (1) Initialization; (2) Activation, (3) Weight training and (3) Iteration. However, these steps may be change according to the problem being solved (Farizawani, Puteh, & Rivaie, 2019).

2.1.4. Gradient Boosting Decision Tree – LightGBM

Ensemble methods combine different classification models to correct their individual weaknesses, which often leads to a more accurate model. Depending on the technique, an ensemble method can be built from different classification algorithms, for example, decision trees, support vector machines, logistic regression, classifiers, and so on. Alternatively, it is possible to use one single method with different subsets of the training dataset, like the random forest algorithm that combines different decision tree classifiers. The following image represents the general approach of an ensemble method (Raschka & Mirjalili, 2019).

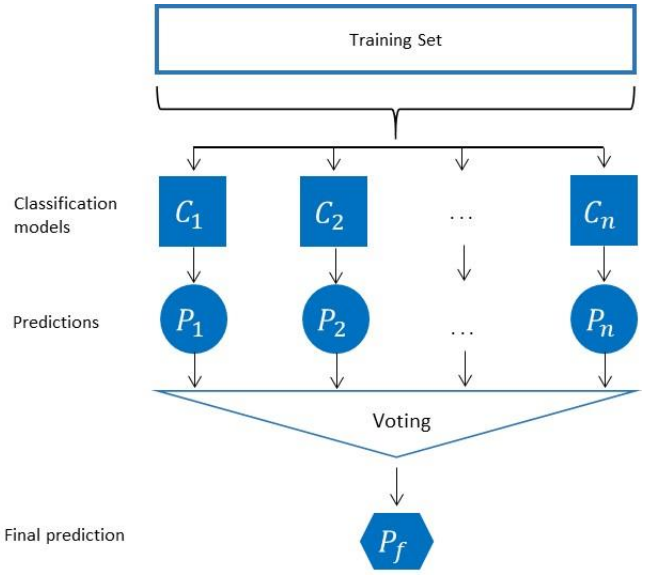

Figure 11: Ensemble Method approach (Raschka & Mirjalili, 2019)

A Gradient boosting decision tree (GBDT) is a commonly-used machine learning algorithm due to its efficiency, accuracy and interpretability. It is made up of an ensemble of weak decision trees by adding them together (trained iteratively – one tree at a time). This algorithm is used to solve many predictive problems like classification, click prediction and learning to rank (Ke, et al., 2017).

There are two different strategies to build the trees: level-wise and leaf-wise. The level-wise strategy grows the tree level by level. In this strategy, each node splits the data prioritizing the nodes closer to the tree root. The leaf-wise strategy grows the tree by splitting the data at the nodes with the highest loss change. Level-wise growth is usually better for smaller datasets, whereas leaf-wise tends to overfit (Ke, et al., 2017).

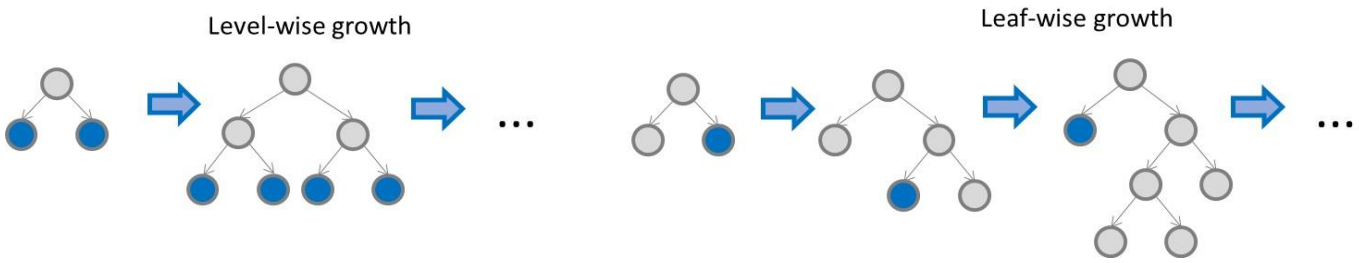

Figure 12: Level-wise and Leaf-wise Growth Strategy

One of the most popular implementations of GBDT is XGBoost. Among the 29 winning solutions in Kaggle's competition[1], 17 used this algorithm. Although, the efficiency and scalability were still unsatisfactory when using high features dimensions and large datasets, so in 2017 Microsoft developed a new algorithm to face these problems: Light GBM.

Both Light GBM and XGBoost use the leaf-wise growth strategy to grow the tree, the most flexible way although more prone to overfitting too, which makes it a better choice for large datasets.

The key challenge in training a GBDT is to find the best split for each leaf, considering that this step requires going through every feature of every data point potentially amounting billions of interactions. One of the main advantages of the new algorithm LightGBM is that it presents two more methods of splitting, beyond the ones presented by XGBoost (Ke, et al., 2017). Gradient-based One-Side Sampling and Exclusive Feature Bunding. The baseline of the first method is that not all data points contribute equally to training, the ones with small gradients tend to be better trained. , In the second method, the baseline is that features that are mutually exclusive can be bundled into a single feature without losing any information.

The authors of this algorithm defend the following advantages compared to XGBoost: (1) Higher efficiency as well as faster training speed, (2) usage of lower memory, (3) better accuracy, (4) Supports Parallel and GPU learning and (5) large-scale data can be handled (Ke, et al., 2017).

In LightGBM, there are three ways to evaluate the importance of a feature:

- "gain" measure implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model
- "frequency" measure is the percentage representing the relative number of times a particular feature occurs in the trees of the model. Meaning that it tells us how often the feature is used in the model.

[1] Kaggle is a Google subsidiary, an online community for data scientists that promotes machine learning competitions to solve data science challenges (www.kaggle.com)

The “gain” measure is the most relevant attribute in interpreting the relative importance of each feature.

3. DATA AND BOOKINGS PROCESS FLOW

The booking process in Uniplaces is quite complex (Figure 8). A student sends a booking request to a landlord, and they can accept it, reject it or do nothing, and in this case, the request status changes to *Expired* after a few days. If the landlord rejects the request, the booking is set as *Rejected*, and if it is accepted it, the booking is set as A*ccepted* pending payment. If the student does not make the payment, the booking is set as *Cancelled*, otherwise it is concluded as *Confirmed*.

As explained earlier, the goal of this study is to predict the landlords' action after a booking request is made so only the information obtained until the student requests a booking (Point A) is considered. The bookings with status *Cancelled* and *Confirmed* are considered *Accepted*, since it is relevant for this study if the student makes the payment or not.

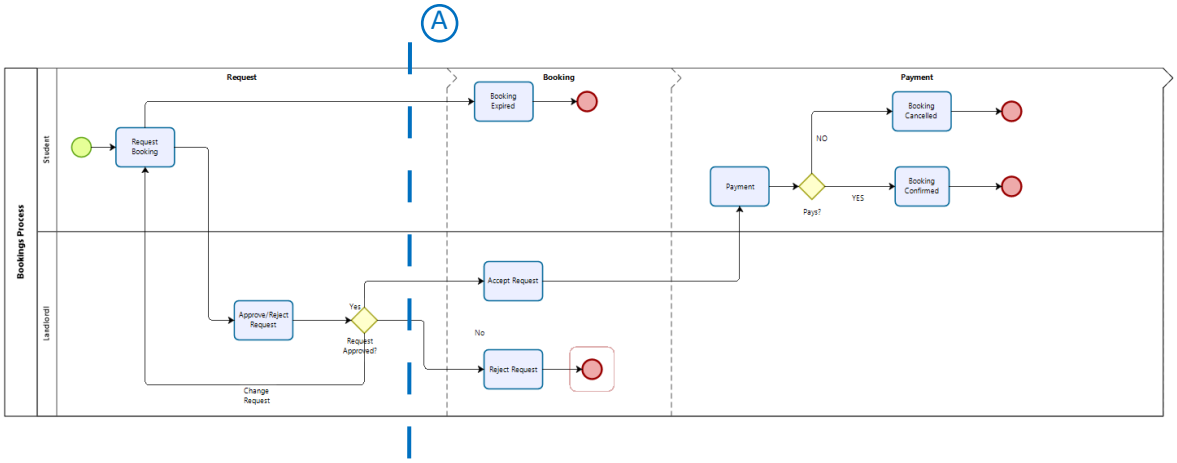

Figure 13: Bookings Process Flow

To construct the algorithm, data from 4 years of company activity was collected: a total of 764.948 bookings and a total of 49 variables. This information can be divided into four groups:

- Student information: sex, age and nationality
- Landlord information: age range, nationality, if he/she lives in the property or not, if he/she has pets and allows pets
- Accommodation information: availability date, if the offer is exclusive to the platform or not, smoking rules, type of accommodation (bed, room or entire house), number of units (beds or rooms), when it was published on the platform, if the property has been verified by Uniplaces
- Booking Information: Request date, acceptance/rejection date, payment date, duration of the request, if it is an instant booking or not

3.1. VARIABLES

In table 3 are described the variables that are considered in the future models.

Variable Name	Variable Description
booking_id	booking id
guest_id	student id once he does the registration on the platform
requested_at	booking date
accepted_at	date in which the booking is accepted (not paid yet)
expired_at	date in which the booking is expired
rejected_at	date in which the booking is rejected
paid_at	date in which the booking is paid
cancelled_at	date in which the booking is cancelled (not paid)
confirmed_at	date in which the booking is confirmed (after payment)
current_state	current booking state (accepted/ expired/ rejected/ confirmed(?))
available_from	property availability (next period available) - present
offer_exclusive	is the property uniplaces' exclusive?
guest_move_in	when the student wants to enter
guest_move_out	when the student wants to leave
new_move_in_at	new entry date proposed by the landlord
new_move_out_at	new departure date proposed by the landlord
rejected_reason	why the landlord is rejecting the booking
profile_gender	student's gender
profile_birth_date	student's date of birth
profile_nationality	student's nationality
accommodation_provider_id	landlord's id
address_city_code	city's code
has_landlord_resident	does the landlord live on the property?
landlord_resident_gender	landlord's gender
landlord_resident_age_range	landlord's age
landlord_resident_occupation	If the landlord’s leaves on the property
landlord_resident_pets	does the landlord have pets?
landlord_resident_family	landlord's family leaning in the property?
restrictions_gender_female	restrictions regarding to female gender?
restrictions_gender_male	restrictions regarding to male gender?
rules_smoking_allowed	allowed to smoke
address_city_code	address city code
verification_internal_verified	Uniplaces has personally visited this place?
offer_type	bedroom / bed / entire-property
id as offer_id	offer id
published_at	when the property was created in the platform

property_id	property id
number_of_units	Number of beds/rooms in the property
availability_last_update_at	when was the last time that the landlorad updated the availability of the property on the platform
ao.available_from	property available in the platform for booking since
created_at	when the property was created in the platform
is_instant_booking	is the booking instant?
created_at.1	when the landlord was created in the platform
ap.out_of_platform	If the property has ever been removed from the platform

Table 3: Variables Description

3.2. DATA VISUALIZATION

The objective of this step is to better understand the data available, detect some trends and analyze the context of the problem. As it was explained earlier, the data is divided into four groups, some plot were employed for each group of data.

The first data group is composed by variables related to bookings. To check the distribution of the bookings through the years, a histogram was plotted.

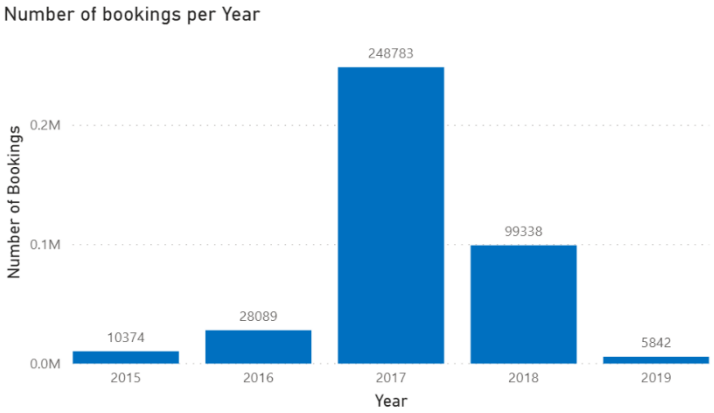

Figure 14: Bookings Distribution by Years

Another measure analyzed in this group was the percentage of instant bookings present in the dataset. The bookings that are approved automatically were not considered in the study. It was observed that only 1% of the bookings do not require the landlord´s action for approval.

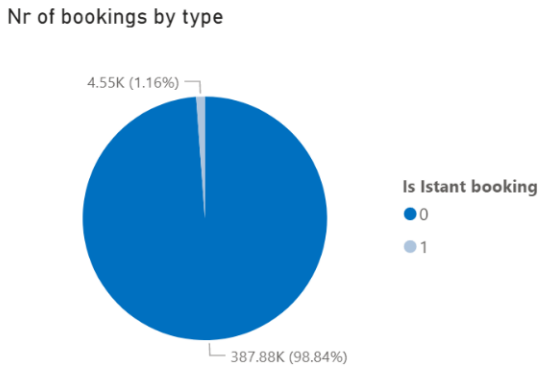

Figure 15: Bookings by type

The second group of variables are related to landlord information. The first graph plotted represented the percentage of landlords that live in the place available for rent in the platform.

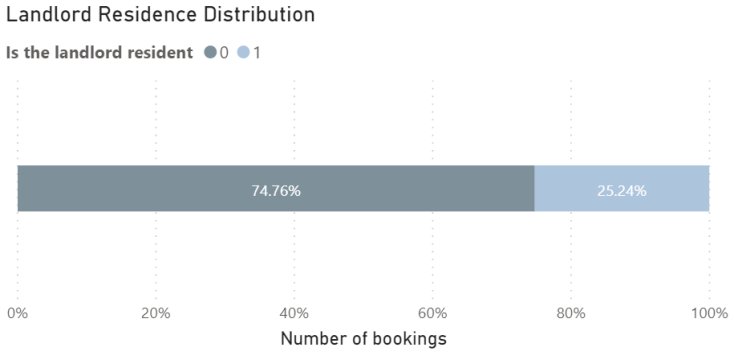

Figure 16: Landlord Residence Distribution

Another visualization plotted was the relation between the age of the landlord and their actions. It is possible to conclude that there is no evident age based behavior when analyzing both measures together. The percentage of acceptance rate seems to be pretty the same independently of the landlord's age.

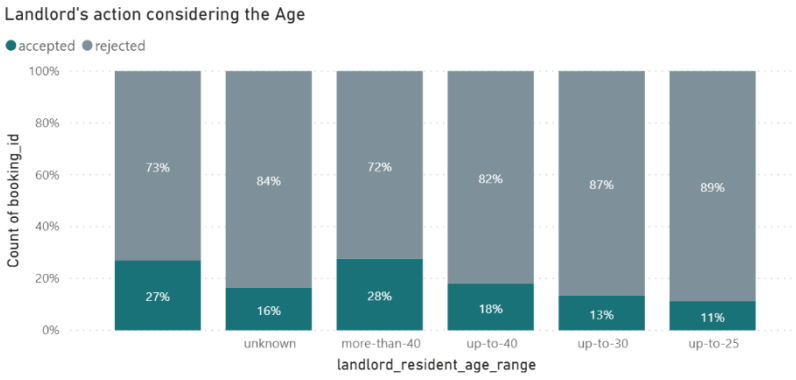

Figure 17: Percentage of acceptance rate considering the age

One of the metrics analyzed in the student information group was the variance of the acceptance rate per gender. It is possible to conclude that this the acceptance rate does not depend on the students gender.

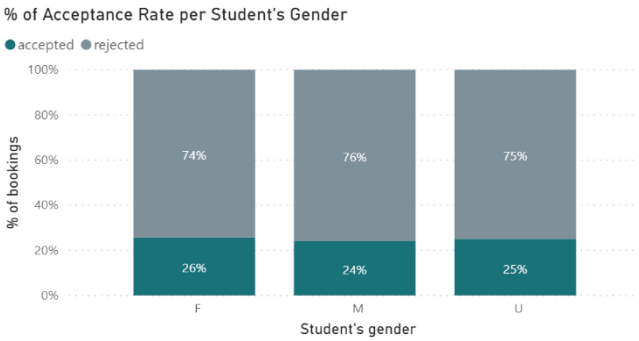

Figure 18: Acceptance rate per student' gender

From the last group, composed by the variables related to the property information, one of the metrics visualized was the acceptance rate considering the smoking rules of the property. It is possible to conclude that it is not allowed to smoke in the majority of properties, and that there is no evident relation between the variables.

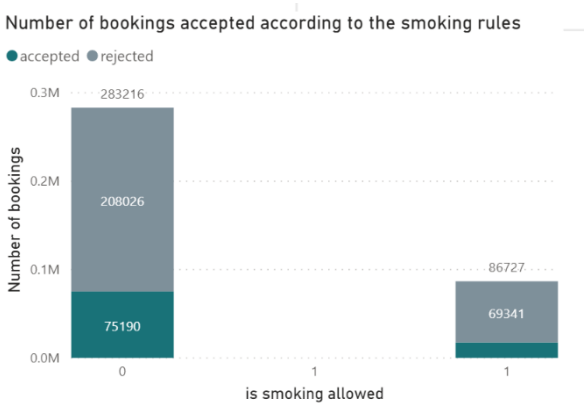

Figure 19: Number of bookings according to smoking rules

4. METHODOLOGY

4.1. APPLICATION OF THE METHODOLOGY

4.1.1. Data Pre-Processing

This part of the project will explore data cleaning and data transformation.

In the first step, data cleaning, it was decided which variables should be considered according to the timeline of the booking process flow (chapter 3. DATA AND BOOKINGS PROCESS FLOW). All the variables that are obtained following the landlord decision to accept or reject a booking request were not taken into consideration.

Regarding the missing values, only eight variables showed some empty values, and only five had a relevant percentage (more than 0,02%):

Variable	%	Nr of missing values
Student's age	0.00%	8
Student's nationality	0.02%	61
Landlord's gender	74.33%	274952
Landlord's age range	74.33%	274952
Landlord's occupation	74.33%	274952
Landlord has pets (T/F)	74.33%	274952
Landlord's family lives in the property (T/F)	74.33%	274952
Number of units	0.01%	24

Table 4: Variables with missing values

The variables with more than 10% of missing values were deleted since it is unlikely they will influence the output variable with only a few registers. These variables had so many missing values because they were implemented on the platform a few years later than the others (the dataset has information from five years, and these variables were implemented in the third year). The other missing values add up to a total of 93 rows (0,03% of the data) so they were also deleted.

In data transformation, to ensure the data is coherent, the following conditions were analysed:

- The days between the student creating the booking until the landlord rejects/accepts the booking must be positive;
- The occupation days must be positive;
- The days since the property was published until the booking day must be positive;
- The number of days since the landlord did the registration on the platform must be positive;
- The student's age must be positive.

All these conditions were consistent except the student's age. This variable has values ranging from -254 to 117. Obviously, there are records that are not true. However, the decision whether the landlord accepts a student or not takes these values into account (meaning that it is less likely a landlord accepts a booking from a 117 -year-old student than a 20- year- old student). As this variable can have a considerable impact on the output variable, the values that are less than 0 were deleted.

Another step of data transformation is dealing with outliers. To detect if there were outliers, boxplots of each of the numerical values were extracted: These outliers were kept in the analysis, because all of them influence the landlord's decision, including those representing false information. For example, most of the outliers from the variable *student's age*, may be false information (it is highly improbable that someone more than 90 years old would try to rent a place on this platform). However, it makes sense that a landlord avoids renting a place to someone that gives false information regarding their age.

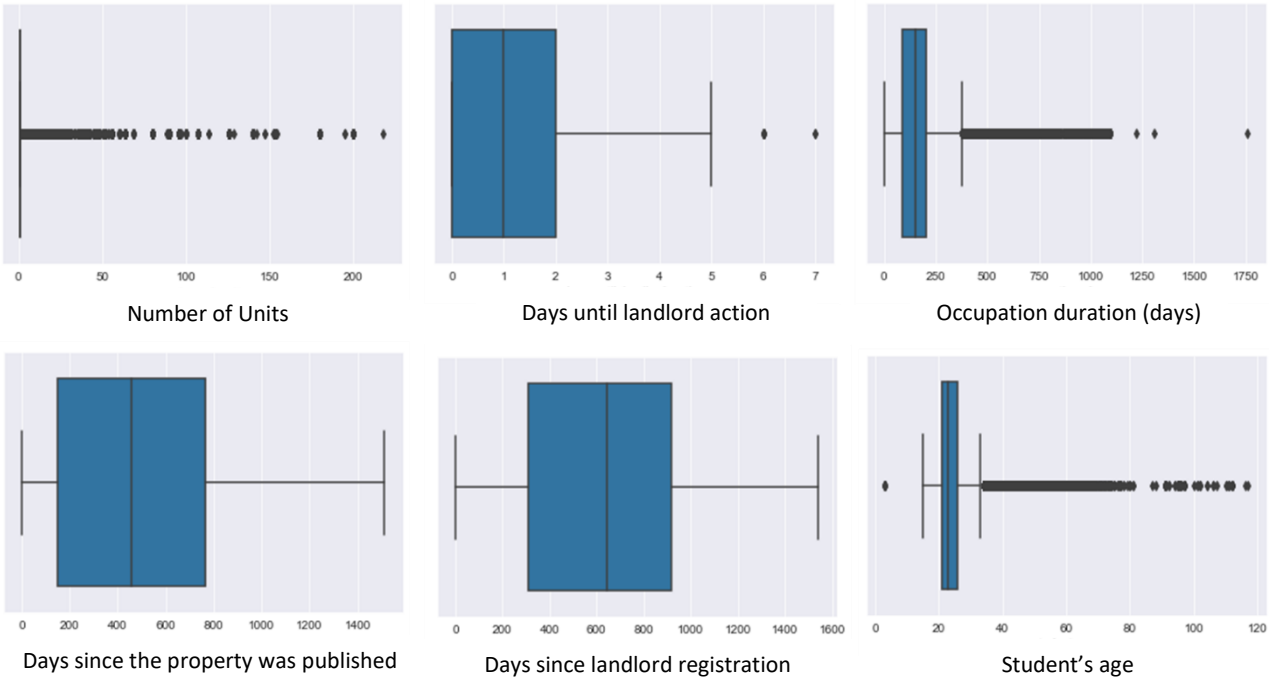

Figure 20: Box Plots

In last step of the data pre-processing phase, several variables were created:

- number of bookings approved by landlord;
- number of bookings accepted by property:
- percentage of bookings approved per property;
- percentage of bookings approved per landlord.

All these variables presented a significant weight when running the models.

4.1.2. Feature Selection

As explored in chapter 2.3. FEATURE PRE-PROCESSING AND SELECTION, there are several techniques to select the best variables that should be used with each model to find the optimal algorithm. Since it is very difficult to find which one is the best method, some of them will be used and divided into different sets:

4.1.2.1. Filter approach - Correlation based Feature Selection

With this technique, the correlation between the variables is analyzed as well as the correlation between each variable and the target. If two values are highly correlated, the one that has the highest correlation with the target is the chosen one, considering Pearson correlation.

Using this approach, it becomes difficult to know exactly how many variables should be considered. Therefore, two sets of independent variables are created:

Set 1: five variables: exclusive offer, days until landlord action, number of bookings accepted per property, number of bookings accepted per landlord, percentage accepted per landlord

Set 2: seven variables: exclusive offer, days until landlord action, number of bookings accepted per property, number of bookings accepted per landlord, percentage accepted per landlord, landlord verified by Uniplaces (T/F), out of platform (T/F)

4.1.2.2. Dimensionality Reduction (PCA)

Another technique previously explained in the Literature Review is the Principal Component Analysis. To decide the number of principal components that should be used, a scree plot was analyzed .

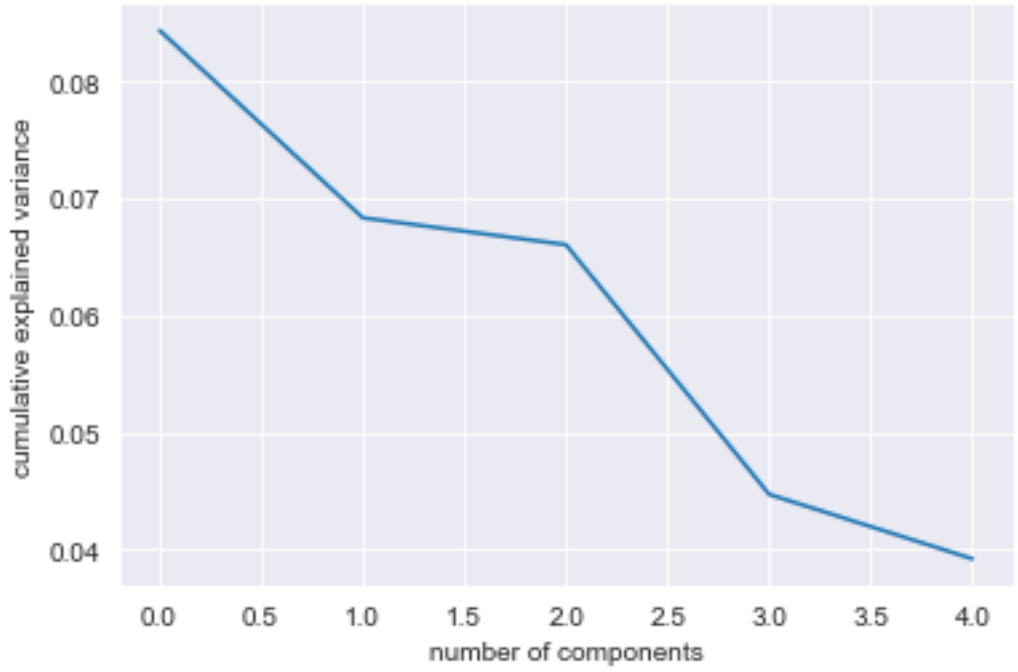

Figure 21: Scree plot

We can see from the above scree plot that the point at which the proportion of variance explained by each subsequent principal component drops off is from the third principal component. That is the number of PCA that will be used in the model (Set 3: 3 Principal Components).

4.1.2.3. No features selection

Finally, the last set of variables is composed of all the variables. There are some models that allow and even work better with a high number of variables that can be correlated between them (for example the Light GBM). It is expected that most of the models shows worse results with dataset.

4.1.3. Data Partition

As discussed in the machine learning theoretical framework, the data splitting method is required to evaluate the models' performance.

Accounting for the target distribution (stratified sets), the data is divided in:

- Training Set (70%): the models will be derived based on this set
- Test Set (30%): the models will be tested in this set and the results will be compared with the real value

4.1.4. Supervised Learning Models

As explored in the literature review chapter, a lot of models have been used and recommended by other online platforms. In this project we will use several of them: Neural Networks, Logistic regression, Support Vector Machine, Decision trees and Gradient Boosting (LightGBM).

4.1.4.1. Logistic regression

The first model applied to the normalized dataset was logistic regression. To avoid overfitting, a 10-cross validation was applied.

The parameters considered in this model were the following:

- C: is a positive floating-point number that defines the relative strength of regularization. Smaller values indicate stronger regularization (used the default value 1.0)
- class_weight: is a dictionary, 'balanced', or None (default) that defines the weights related to each class. When None, all classes have the weight one (using the default value None)
- max_iter: is an integer that defines the maximum number of iterations by the solver during model-fitting (using the default value 100)
- l1_ratio: is either a floating-point number between zero and one or None. It defines the relative importance of the L1 part in the elastic-net regularization (using the default value None).

4.1.4.2. SVM

The most common library to implement machine learning algorithms in Python is scikit-learn, which was used to build this algorithm.

The parameters considered in this model were the following:

- C: It is the regularization parameter, C, of the error term (using the default 1.0)
- kernel: It specifies the kernel type to be used in the algorithm. It can be 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed', or a callable (using "linear").
- degree: It is the degree of the polynomial kernel function ('poly') and is ignored by all other kernels (using the default value 3)
- gamma: It is the kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If gamma is 'auto', then 1/n_features is used instead (using the default value "gamma")

4.1.4.3. Decision Trees

The most challenging part of this algorithm is optimizing it by pruning the tree. This means, there are some parameters that must be tested and the best one must be selected, according to the current model:

- Criteria: this parameter allows to choose the best attribute selection measure. The criteria selected was the gini index;
- Splitter: this parameter allows to choose the best split strategy ("best" or "random"). The criteria selected was "best."
- max_depth: this parameter allows to choose the maximum depth of a tree. The higher this value, the more susceptible it is to overfitting, and the lower this value is, the more susceptible to underfitting. The maximum depth selected was 3.

Non-normalized data was applied to this model.

4.1.4.4. Neural Network

After some data preparation such as standardization and encoding, the four datasets were applied to Neural Network.

After several parameters were tuned, this model was optimized with 3 hidden layers, a learning rate of 0,01 and it used Adam, a momentum-based optimizer. The loss function used is binary_crossentropy (usually the best choice for binary classification problems that give output in the form of probability) and the model is trained for 50 epochs with a batch size of 1.

4.1.4.5. Gradient Boosting – LightGBM

According to the LightGBM literature (Ma et al., 2018; Ke et al., 2017), the algorithm achieves the optimal performance when using the following parameters (MINASTIREANU & Mesnita, 2018).

- max_depth: this parameter controls the max depth of the trees. Higher value of max_depth increases model complexity but at the same time can lead to overfitting. Typical values range from 3- 10 (set as 3)
- learning_rate: this parameter controls the rate of learning considered in the algorithm (set as 0,2)
- num_leaves: this parameter controls the number of leaves (set as 7)
- gamma: this parameter controls the penalty on model complexity. Higher gamma decreases model complexity and decreases the chance of over-fitting. Typical values range from 0-2. (set as 0,9)
- min_child_weight: this parameter controls the minimum sum of instance weight of all observations needed in a child (leaf). Higher value prevents over-fitting. Typical value ranges from 1-20 (0)

5. PRESENTATION OF THE RESULTS

On the following pages, the results of all the models are shown. As explained previously, each model was tested with three or four datasets.

5.1. MOST IMPORTANT VARIABLES

One of the main goals of this project was to identify the main reasons for a landlord to accept/reject a booking. There are several ways to identify which variables most influence the landlord's behavior. For example, it can be used the correlation matrix to calculate the correlation matrix can be used to calculate the correlation between the target variables and the input variables. The variables with the highest correlation influence have a higher influence on the landlord's decision.

Variable	Corr(Var, Predicted Variable)
percentage_acceptance_landlord	0.892297568
percentage_acceptance_property	0.890454844
property_nr_bookings_accepted	0.619796501
days_until_landlord_action	0.45114873
landlord_nr_bookings_accepted	0.448463272
offer_exclusive	0.213499844
verification_internal_verified	0.144247701
out_of_platform	0.124168603

Table 5: Variables with highest correlation with dependent variable

The three variables that most influence the landlord's behavior are the following calculated variables: landlord acceptance rate (% of bookings accepted by the landlord), property acceptance rate (% of

bookings accepted per property) and number of bookings accepted per property. The other variables that have a significant influence are the number of days between the booking request and he landlord's decision, number of bookings accepted by the landlord, if the property is exclusive to Uniplaces, if the landlord is verified by Uniplaces and if the property has ever been removed from the platform.

5.2. RESULTS OBTAINED WHEN USING FILTER SELECTION

In the table 6 are shown the results of the models when using correlation-based feature selection, with 5 and 7 variables.

	Logistic Regression		SVM		DT		NN		Light GBM	
	5 Variables	7 Variables	5 Variables	7 Variables	5 Variables	7 Variables	5 Variables	7 Variables	5 Variables	7 Variables
Accuracy	0.996	0.996	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000
Precision	0.995	0.995	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000
Recall / True Positive Rate (TPR)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
False Positive Rate (FPR)	0.015	0.014	0.003	0.003	0.000	0.000	0.000	0.000	0.000	0.000
F1-score	0.998	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AUC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 6: Results obtained when using filter selection

5.3. RESULTS WITHOUT USING ANY FEATURE SELECTION

The table 7 shows the results obtained with the dataset with all normalized variables. The only models that must be performed considering all variables are the Neural Networks and the Gradient Boosting (Light GBM). Both models predict 100% of the test dataset.

	NN	Light GBM
	All variables	All variables
Accuracy	1.000	1.000
Precision	1.000	1.000
Recall / True Positive Rate (TPR) sensitivity	1.000	1.000
False Positive Rate (FPR)	0.000	0.000
F1-score	1.000	1.000
AUC	1.000	1.000

Table 7: Results obtained without using any feature selection

5.4. RESULTS OBTAINED WHEN USING PRINCIPAL COMPONENTS ANALYSIS

In the following table, the results are obtained applying the principal component analysis. As It was expected after the previous results, all models predict to 100% of the test dataset.

	Logistic Regression	SVM	DT	NN	Light GBM
	PCA				
Accuracy	1.000	1.000	1.000	1.000	1.000
Precision	1.000	1.000	1.000	1.000	1.000
Recall / True Positive Rate (TPR) sensitivity	1.000	1.000	1.000	1.000	1.000
False Positive Rate (FPR)	0.000	0.000	0.000	0.000	0.000
F1-score	1.000	1.000	1.000	1.000	1.000
AUC	1.000	1.000	1.000	1.000	1.000

Table 8: Results obtained when using Principal Components Analysis

5.5. SYNTHESIS OF THE RESULTS

As explained in detail previously, the objective of this thesis was to approach a business issue faced currently by the marketplace Uniplaces, making use of machine learning models and the available four years of transactional data.

The main conclusion of this work is that the stated problem can be easily predicted with the mentioned available data.

All of the mentioned models presented 100% of AUC without applying any feature selection, when applying PCA or filter selection the models maintain the AUC of 100% as expected. So, there are no difference in the results when applying a feature reduction technique.

These results can be explained if we analyze the correlation between the input variables and the predicted variable. Three variables have a correlation with the predicted variable higher than 60%, and two of them achieve almost 90% of correlation.

These results were not foreseen before the execution of this thesis but in hindsight and after some analysis it can be easily explained.

In summary, Uniplaces could easily improve their acceptance rate by applying any of the models mentioned in this thesis to their recommendation system.

6. LIMITATIONS

Although all models present very good results, there are some steps in the process that can be improved.

First, the model is only presented as a solution to improve the acceptance rate, considering that at this point, the majority of the users are students and it does not make sense to include their behavior in the model because the choice of the property will be always based on the lowest price. Beyond that, the average booking per user is very close to 1, meaning that most of the students only use the platform once, making it difficult to study their behavior and preferences through the platform usage. If this environment changes, user preferences could be taken into consideration when creating a recommendation system.

Regarding data collection, some new variables can be explored and analyzed by the company such as how often each landlord logs in to the platform, when they last logged in, when they last booked the property, etc.

Some of the variables were recently implemented on the platform, so they do not have registers from previous periods, that represent a high number of missing data in the current dataset. These variables can be used when they have sufficient historical data.

Also, deeper outliers' analysis and other dimensionality reductions or features selections could be explored.

Furthermore, many other types of predicted models can be developed, such as the ones referenced in the literature review: RankSVM, Random Forests, etc.

7. CONCLUSIONS AND FUTURE WORK

As previously stated, the objective of this model was to assist the online market Uniplaces with a business problem related to the information ranking displayed to their users. Currently only 25% of the bookings made by their users are accepted by the property owners. Of the 75% of total users whose booking has not been rejected or ignored, only 10% try to book another property.

This low acceptance rate has a very significant effect on the profitability of Uniplaces business, and any improvement in the acceptance rate would have a direct and immediate effect on the business bottom line, as higher acceptance rates would imply a higher number of approved bookings and consequently a higher amount of fees billed by Uniplaces.

The solution proposed is to build a model to rank the properties for each user seeking accommodation, based on the probability of being accepted. If the ranking of properties displayed is aligned with the probability of acceptance of each property, the probability that a user chooses a property that will be accepted increases, thus increasing the overall acceptance rate.

Looking at bookings over 4 years, it was possible to predict whether a landlord accepts or rejects a booking, building and training a classification model. This model is able to automatically pattern the behavior of the landlord, and then, for each student that logs in to the platform, it can then list the properties in order of probability of being accepted.

The information collected to build the algorithms, collects information regarding the student and the property. All this information was cleaned and transformed to fit the model's requirements. After that, several machine learning models were used. These models were chosen based on what other online platforms are currently using and recommend, namely Support Vector Machine Model, Neural Networks, Decision Tree, Gradient Boosting (LightGBM) and Logistic Regression.

The variables that represented a strong influence on the target variable were the percentage of bookings accepted per landlord and per property, the number of bookings accepted per property, the time since the user requests the bookings until the landlord accepts it or rejects it, the total number of bookings accepted by the landlord and whether or not the property is exclusive to the platform.

The final decision for the models takes the performance in terms of AUC into consideration and its variance in a further validation with a new portfolio.

The problem turned out to be very easy to predict, mostly because of the high correlation between the target variables and the input variables. When applying PCA, all models predict 100% of the test dataset. When applying the filter selection method, there was no difference in the results between the dataset with 5 variables and the dataset with 7 variables. The only models that do not achieve 100% of accuracy are the logistic regression and Support Vector Machine; all the other models predict all the test dataset applying either feature selection technique or PCA.

Although the results are very good, some steps in the algorithms can be improved. In data pre-processing, more sophisticated techniques for parameter- tuning could be developed, especially for artificial neural networks, as well as other predictive models.

Also, to be considered in future analyses, these algorithms presented as a solution for the low acceptance rate are only applicable in the current context of the company: the majority of the users that look for a place to stay are students and the average number of bookings per user is one. If the average number of bookings per user increases, the user's preferences should be tracked on the platform and considered in the recommendation system.

8. REFERENCES

Skorpil, V., & Stastny, J. (2006). Neural Networks and Back Ppropagation Algorithm.

Aruna, R., & Nirmala, K. (April 2013). Construction of Decision Tree : Attribute Selection. *International Journal of Advancements in Research & Technology*.

Aryafar, K., Guillory, D., & Hong, L. (2017). An Ensemble-Based Approach to Click-Through Rate Prediction for Promoted Listings at Etsy.

Chapelle, O., & Chang, Y. (2011). Yahoo! Learning to Rank Challenge Overview.

CHAPELLE, O., VAPNIK, V., BOUSQUET, O., & MUKHERJEE, S. (2002). Choosing Multiple Parameters for Support.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.

Cilimkovic, M. (2015). Neural Networks and Back Propagation Algorithm.

Dolnicar, S. (2018). *Peer-to-Peer Accommodation Networks: Pushing the boundaries.* Goodfellow Publishers Limited.

Farizawani, A., Puteh, M., & Rivaie, A. (2019). A review of artificial neural network learning rule. *Journal of Physics: Conference Series*.

Guo, Q., Geyik, S., Ozcaglar, C., Thakkar, K., Anjum, N., & Kenthapadi, K. (2019). The AI Behind LinkedIn Recruiter search and recommendation systems.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*.

Haldar, M., Ramanathan, P., Sax, T., Abdool, M., Zhang, L., Mansawala, A., . . . Liao, J. (2020). Improving Deep Learning For Airbnb Search.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting.

Kenthapadi, K., Le, B., & Venkataraman, G. (2017). Personalized Job Recommendation System at LinkedIn: Practical Challenges and Lessons Learned.

Leino, J., & Räihä, K.-J. (2007). Case Amazon: Ratings and Reviews as Part of Recommendations.

Lorena, A., & Carvalho, A. (2007). Uma Introdução às Support Vector Machines.

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook.*

MINASTIREANU, E.-A., & Mesnita, G. (2018). Light GBM Machine Learning Algorithm to Online Click Fraud. *Journal of Information Assurance & Cybersecurity*.

Mitchell, T. (1997). *Machine learning.*

Mueller, A., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists.*

Nair, B. B., & Mohandas, V. (2010). A Genetic Algorithm Optimized Decision TreeSVM based Stock Market Trend Prediction.

Néda, Z., & Biró, T. (July 2020). Gintropy: Gini Index Based Generalization of Entropy.

Neville, P. (1999). Decision Trees for Predictive Modeling.

P. N. (August de 1999). Decision Trees for Predictive Modeling.

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2.*

Rojas, R. (1996). *Neural Networks - A Systematic Introduction.* Springer-Verlag Berlin Heidelberg.

Shi, Y., Li, J., & Li, Z. (2018). Gradient Boosting with Piece-Wise Linear Regression Trees.

Srinivasan, D., & Pavya, K. (2017). Feature Selection Techniques in Data Mining: A Study.

Taylor, J., & Cristianini, S. (2000). An introduction to support vector machines.

Varma, A., Jukic, N., Pestek, A., J. Shultz, C., & Nestorov, S. (2016). Airbnb: Exciting innovation or passing fad?

Ville, B. (2006). *Decision Trees for Business Intelligence and Data Mining.*

Yin, D., Hu, Y., Tang, J., Daly Jr, T., Zhou, M., Ouyang, H., . . . Chang, Y. (2016). Ranking Relevance in Yahoo Search.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). *Large-Scale Parallel Collaborative Filtering for the Netflix Prize.*

Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm.