UPPSALA UNIVERSITET

UPTEC STS 22005

Examensarbete 30 hp

Januari 2022

Predicting purchase intentions of customers by using web data

To identify potential customer groups during sales processes in the real estate sector

Olle Kåhre Zäll

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA
UNIVERSITET

Predicting purchase intentions of customers by using web data: To identify potential customer groups during sales processes in the real estate sector

Olle Kåhre Zäll

Abstract

This master thesis aims to investigate the possibilities of predicting purchase intentions of customers during their sales processes in the real estate sector. Also, the web activity of customers on a real estate company's web site is used as the basis for the forecasting. A machine learning framework has been developed, where its compliance with the GDPR is also assessed. Five supervised machine learning algorithms – logistic regression, *k*-nearest neighbors, decision tree, random forest, multilayer perceptron – have been utilized for predicting the classes of the customers: buyers and non-buyers. Three data sets were generated, which represented the total number of active customers at different points in time: at the same day as a sales process starts (day 0) and 10 and 20 days after it. The algorithms were applied and evaluated on these data sets to identify when it is suitable to predict the purchase intentions of customers. To increase the generalization capability of the algorithms, hyperparameter optimization along with data resampling by combining undersampling and synthetic minority over-sampling techniques, *k*-fold cross validation and mutual information, as feature selection, were applied.

The results show that the number of visited web pages, sessions, searched projects (concerning accommodations) and searched locations were relevant for all three data sets. The average price (in total and per square meter) of the most frequently visited web page regarding projects were also included in all the data sets. In addition, the total number of registration of interests sent, and the total amount of time spent on the company's web site were considered in the second (day 10) and third data set (day 20). Further, a multilayer perceptron – applied 10 days after the start of a sales process – was considered as the optimal model for classifying the purchase intentions of customers. Moreover, the developed machine learning framework is argued to be compliant with the GDPR. Further evaluation regarding the compliance needs to be conducted if the methodology of this machine learning framework would be implemented in practice.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Helena Sjöberg Ämnesgranskare: Niklas Wahlström

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig sammanfattning

Bostadsmarknaden är osäker och under en säljprocess är det ofta svårt för ett företag i den här branschen att veta huruvida en kund antingen köper eller inte köper en bostad. Att identifiera köpavsikterna tidigt i kunders säljprocesser är således ett incitament för att kontrollera och reducera ovissheten.

Den här studien genomfördes på ett fastighetsutvecklingsföretag med syftet att klassificera köpavsikten av företagets kunder som har varit involverade i säljprocesser, för att identifiera deras intentioner så tidigt som möjligt. Som grund för det här har kundernas aktivitet på företagets hemsida använts. Övervakad maskininlärning har applicerats för att klassificera kunderna till sina respektive kundgrupper. I det här fallet har alltså redan befintliga kunder varit kategoriserade, vilket olika maskininlärningsalgoritmer har tränats utifrån och testats på. Totalt har fem sådana algoritmer applicerats och utvärderats vid olika tidpunkter: ett dataset från dagen då säljprocessen börjar, ett från 10 dagar senare och ett från 20 dagar efter starten. Vidare har olika metoder använts för att öka algoritmernas generaliserbarhet; egenskapen att kunna skatta så korrekt som möjligt när ny data utreds.

Tre frågeställningar har utformats i den här studien. Den första utreder vilka variabler som har använts i respektive dataset. Sedan utforskas den optimala maskininlärningsalgoritmen och därmed när en kunds köpavsikter bör skattas under en säljprocess. Slutligen utreds huruvida studiens utvecklade maskininlärningsramverk följer GDPR, eftersom data kopplade till individer har använts.

Resultaten visar, till att börja med, att en kunds aktivitet, exempelvis att besöka webbsidor som handlar om bostäder och tiden spenderat på företagets hemsida, är relevanta variabler. Fortsättningsvis var en mer komplex algoritm den optimala bland alla som utvärderades, vilken skattade alla aktiva kunders köpavsikter 10 dagar från att deras respektive säljprocess hade börjat. Således föreslås det att skatta en kunds köpavsikter 10 dagar efter att en säljprocess påbörjas. Slutligen anses det att det utvecklade maskininlärningsramverket följer GDPR. Det påstås dessutom att fortsatt utredning bör göras om företaget vill implementera liknande metodik i produktion för att klassificera kunder till sina respektive kundgrupper.

Acknowledgments

This Master Thesis project has been conducted together with a real estate company. I would like to thank my supervisor Helena Sjöberg for the constructive discussions and valuable insights throughout the report. Also, I want to thank the company for providing me with the data sets. Lastly, I would like to thank my subject reader Niklas Wahlström for the help and guidance.

Olle Kåhre Zäll
January 2022

Abbreviations

AUC	Area under the curve
CRM	Customer relationship management
DPIA	Data protection impact assessment
DT	Decision tree
EU	European union
FPR	False positive rate
GDPR	General data protection regulation
HTTP	Hypertext transfer protocol
IR	Imbalance ratio
LR	Logistic regression
MI	Mutual information
ML	Machine learning
MLP	Multilayer perceptron
NA	Not available
k**-NN**	k-nearest neighbors
NPV	Negative predictive value
PPV	Positive predictive value
PR	Precision-recall
RC	Random classifier
RF	Random forest
ROC	Receiver operator characteristic
SD	Standard deviation
SMOTE	Synthetic minority over-sampling technique
SP	Sales process
TOM	Time on the market
TNR	True negative rate
TPR	True positive rate
URL	Uniform resource locator

Table of Contents

1. Introduction

The sales process (SP) is a critical step for companies in the real estate industry (Cheng et al., 2008). In fact, the time it takes to sell a property – also known as the time on the market (TOM) – is a measure for liquidity. This is due to the very nature of this industry: the transactions of products are infrequent with a limited number of buyers, where the factors contributing to the TOM of an accommodation are many. Thus, it is a tedious challenge to control and understand the SP as a real estate company.

Several studies have been conducted to understand the TOM. In their study, Ferreira and Jalali (2015) identified the plausible factors which affect the housing sales and TOM by considering ideas and thoughts from a panel of experts. Dombrow and Turnbull (2007) discovered how the influence of individual companies affects this process by studying their characteristics and strategies. Cheng et al. (2010) focused on the optimal period of selling a property, since the timing affects the willingness of customers to buy accommodations. Banaitis et al. (2016) developed a framework which identified the most valuable components in a service provided by the real estate company to the customer.

This study is in collaboration with a real estate development company which sees a high value in using the data of its customers to investigate if their web activity can be used for understanding the outcome of a SP. More specifically, the company wants to investigate whether the customers' web activity on its web site can be utilized for predicting potential buyers and non-buyers – customers who lose a SP – as early as possible during the SPes by using machine learning (ML). However, at the time of writing this thesis, the author has not found any study that investigates the possibilities of controlling the TOM in this manner. Instead, web usage mining techniques, that often are applied in e-commerce businesses for understanding purchase intentions of customers (Liu, 2011, pp. 449–483), have been utilized.

In this thesis, a state-of-the-art supervised ML framework to predict the purchase intentions of customers as early as possible in SPes by applying and evaluating classification models at different points in time has been developed.

1.1 Research aim

The purpose of this thesis is to explore the possibilities of predicting the customer groups – buyers and non-buyers – of customers based on their web activity on a company's web site. Five classification algorithms – logistic regression, k-nearest neighbors, decision tree, random forest and multilayer perceptron – are evaluated with different hyperparameters during different points in time to identify when it is suitable to classify the purchase intention of a customer during the SP. Different data sets are generated, where certain input variables are considered in each data set by using a filtering method as feature selection. Furthermore, since data related to individuals are considered, the ML framework is evaluated whether it is compliant with regards to the GDPR.

The following research questions will thus be explored:

- Which are the most relevant input variables in each data set?
- When is it recommended to classify the purchase intentions of customers during SPes?
- Are there any challenges for this ML framework with respect to the GDPR?

1.2 Delimitations and limitations

The characteristics of buyers and non-buyers are many. However, the company was interested in identifying potential, or future, buyers and non-buyers of customers who have not purchased an accommodation. Thus, delimitations regarding each customer group have been considered. To begin with, buyers who have purchased more than one accommodation was excluded from this thesis. Next, since a customer can be involved in several SPes, the latest one – which the customer lost – was considered for the non-buyer.

The company stores data of all the SPes. This means that each customer's SP is also recorded, where the customer's status during a SP is handled manually by the company. This enables to have an overview of the stages of a SP for each customer, and how long this process was. However, this methodology has not always been utilized in the company and some customers' SPes could not be utilized. Thus, those customers were discarded from this thesis.

1.3 Disposition

This thesis consists of eight chapters. The next chapter, Chapter two, covers the web mining field with a certain focus on web usage mining techniques. Additionally, how the GDPR affects the usage of ML is explored in this part. Chapter three discovers the ML algorithms. Also, how to improve the performance with additional techniques in a ML framework and how to evaluate a classifier is provided. Next, the database systems, in which data have been utilized from, are introduced in chapter four. Chapter five explores the methods used for constructing the ML framework. In chapter six, the results are presented and a discussion of those is included in the following chapter. Chapter eight consists of the conclusions of this report. Furthermore, supplementary materials are included in Appendix A.

2. Web usage mining

In this chapter, the connections between data mining and web usage mining is first explored. A section regarding how the data is prepared for the web usage mining methods is thereafter presented, and followed by a section of how patterns are discovered and analyzed in this field. Finally, a section regarding ML with respect to the GDPR is included.

2.1 Data mining and web mining

According to Hand et al. (2001, pp. 1–4), data mining can be defined as "... the analysis of [...] data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner". This process contains several steps: choose the target data, perform preprocessing, use transformations (if necessary), apply data mining techniques to discover patterns and connections and, finally, assess and analyze the findings. The foundation of the data mining field is a combination of statistics, databases and algorithms.

A sub-category within the data mining field is web mining (Cooley et al., 2000). Web mining is defined as the process of utilizing data mining techniques to discover patterns in data on the World Wide Web (Liu, 2011, pp. 7, 451). There are mainly three different approaches in the web mining field: web content mining, web structure mining and web usage mining. Web content mining is the process of extracting information - such as the text and graphics shown for the user - from a web page, whereas web structure mining is mainly used for studying the data related to the structure of a web site. These two approaches are not included in this thesis and will not be further explored. Web usage mining (Figure 1) is the process of finding behavioral patterns of users that are interacting with a web site. The patterns extracted from users can be utilized for different reasons: to predict future behavior, to get a better understanding of the user segments, to provide personalized content by recommender systems, or to improve the structure of the web site.

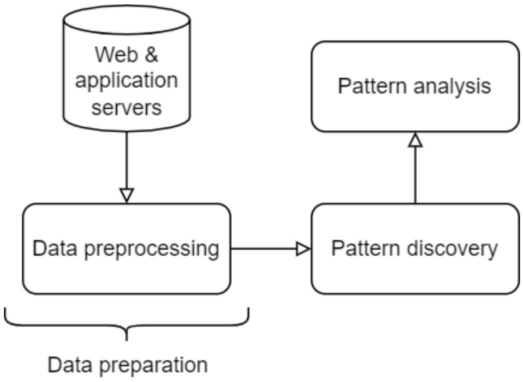

Figure 1. The web usage mining process.

The web usage mining process can be divided into three phases: data preparation, pattern discovery and pattern analysis (Liu, 2011, p. 449), which will be explored further in Section 2.2 and 2.3.

2.2 Data preparation

Data in the web usage mining field, denoted as web data henceforth, is information about the web user visiting a web site that is collected by the web or application servers (ibid., p. 452). Each activity connected to a web user on a web site corresponds to a hypertext transfer protocol (HTTP) request to the server(s), which is stored in the server access logs where information, e.g., cookies can be included. Cookies are small blocks of data that are sent to the client machine the first time the web user visits a web page – to be able to identify the web user in the future (Ramesh and Thushara, 2016). This can be achieved by implementing a Javascript on a web page, which extracts the values from the cookies. Further, every time a web user visits a web site, the activity will be recorded in so-called sessions (Liu, 2011, pp. 453–454). A session can be described as the trail of pages belonging to a unique user, which contains the activity for a period of time, and cookies can be utilized to identify sessions connected to web users.

The most basic level of data abstraction, of web data, is the pageview (ibid., pp. 449–458). This represents the activities achieved by a web user, e.g., clicking on a web page or adding an item to the shopping cart on an e-commerce website. A session is constructed by the collection of pageviews. Furthermore, the pageviews are preprocessed to generate user transactions: high-level representations of the web users' activity on a web site, i.e., all the visited pageviews in one or more sessions can be formulated and prepared for pattern discovery and analysis. In addition, user data – the personal information of a web user that is stored in the web owner's database(s) – is also common to use in the web usage mining field. This can be different kinds of information, such as history of purchases, user ratings and user interests.

2.3 Pattern discovery and analysis

Several techniques are used for different purposes in the web usage mining field (Cooley et al., 2000). To begin with, descriptive statistical analysis is usually applied on web users' sessions. This can be used to, e.g., explore the most accessed web pages on a web site, the average time spent from all sessions, the average navigational path length traveled through a web site etc. Moreover, another common method is association rules. This technique is useful for exploring the relations between web pages on a web site connected to a specific web user's session. Exploring the users' activity on a web site using this method can give rise to valuable marketing and business insights for the web owner. Furthermore, clustering is a common method in the web usage mining field to group records with similar characteristics on the web site. It is often applied to discover usage clusters – to identify how different web user groups behave. For example, clustering can be used for grouping web users with similar navigational pattern behavior to retrieve insights about them, which can be useful for market segmentation or to provide personalized web content.

Another commonly used method is dependency modeling: a pattern discovery technique which can be useful to find a unique navigational, or behavioral, pattern of a web user (Cooley et al., 2000). This method is valuable to distinguish users apart from each other; some may only be casual visitors, while others might be potential buyers. Probabilistic ML methods are common in this area, such as hidden markov models and bayesian belief networks. Another common ML model utilized within the dependency modeling approach is the recurrent neural network (Kastro et al., 2019). Predicting user behavior can give insights about the web site, where it can help the business unit to change strategies to increase sales of specific products (Cooley et al., 2000). Lastly, classification, the process of mapping data to predefined categories, specified by the web owner, is another method used in this field. This technique is useful for identifying a specific class of a web user, and different classification models can be applied for this purpose. This ML approach is useful for personalization and understanding the web user profile.

The last step in the web usage mining process is pattern analysis (ibid.). The results from the previous step are now filtered and the interesting rules, patterns or statistics are only considered at this point. At this step, interesting findings are subjective and defined by the web owner.

2.4 General data protection regulation

The General data protection regulation (GDPR) is a European Union (EU) law which aims for protecting the personal data of individuals of the member states which is processed by controllers (European Union, 2016). In Article 4(1) of the GDPR, personal data is defined as "... information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person". Further, the controller can be a public authority, company etc. which processes data of individuals that are, e.g., collected on web sites by using cookies (Recital 30 in the GDPR). The pattern of an individual's activity can be connected to data within a controller's servers to generate a profile of a data subject. If a controller breaches Article 22, it will receive a fine of up to 20 million euros or "... up to 4% of the total worldwide annual turnover of the preceding financial year..." (Article 83(5) in the GDPR).

Using personal data of data subjects is known as profiling, which is the procedure of evaluating personal aspects – such as economic situations, personal preferences, interests, behavior etc. – for different prediction purposes (Article 4(4) in the GDPR). It can be applied for classifying data subjects to predetermined categories, where ML algorithms are commonly used in the profiling process (Gutwirth et al., 2017, p. 94). The results can be used as the basis for decision-making.

According to Gutwirth et al. (ibid., p. 94), profiling is a process which is constituted by three parts: (I) collection of data[1], (II) development of ML algorithms and (III) the process of making decisions based on the results. Both fair and transparent processing of data is a must, where discrimination of an individual's sensitive data such as "... racial or ethnic origin, political opinion, religion or beliefs, trade union membership, genetic or health status or sexual orientation..." must be prevented (Recital 72 in the GDPR). It is necessary to evaluate the impacts of how the protection of personal data is affected by processing, which is also known as a data protection impact assessment (DPIA) (Article 35(1) in the GDPR). As stated in Article 35(3)(a) in the GDPR, DPIA should be applied when personal characteristics are evaluated in profiling, where individuals can be significantly affected by the decisions taken from the results. Certain security measures must be conducted during this assessment, as stated in Article 7(d) GDPR. According to the authors Gutwirth et al. (ibid., p. 100), one of those would be to evaluate ML algorithms in a simulation environment to "... identify problems with biases in the data and mitigate potential negative outcomes before being used on a larger scale". Since values are absorbed by an algorithm, bias may be introduced in a ML application (ibid., pp. 103–106). It can be either direct, e.g., actively filtering individuals based on sensitive data, or indirect, e.g., generating unreliable results by unintentionally using data sets where a minority group is underrepresented. The DPIA is therefore necessary to utilize for preventing unfair discrimination (ibid., pp. 103–106). This is an urgent task for assessing the fairness of a ML application (Cate et al., 2017). Gutwirth et al. (2017, pp. 103–106) also state that it is important to evaluate if a DPIA is required in each ML project which concerns personal data. Further, a controller should be able to pseudonymize data subjects, which is a method of detaching personal data connected to a specific individual (Article 4(5) in the GDPR). The individuals can however still be identified to their personal data but this information should be kept separately. According to Recital (28) in the GDPR, pseudonymization is a useful technique partly to protect the privacy of data subjects, partly to help, e.g., controllers to follow the GDPR regulation.

In terms of transparency, a controller needs to present information whether automated

[1] A controller can record data of data subjects directly and indirectly (Gutwirth et al., 2017, pp. 94–96), and the data subject has the right to object to this according to Article 6(f) in the GDPR.

decision-making or profiling exists to the data subject when personal data is collected (Article (13)(2)(f) in the GDPR). Furthermore, “meaningful information about the logic” should be provided as well (Article (13)(2)(f) in the GDPR). According to Cate et al. (2017), this should not be interpreted as to give a full detailed description of a ML application or the code behind it, but rather “a high-level, non-technical, description of the decision-making process...”. Furthermore, Gutwirth et al. (2017, p. 108) state that it is possible to explain this to a data subject without revealing intellectual property rights, i.e., how the collection and labeling of training data sets have been conducted, which models that are utilized and their chosen hyperparameters, and the performance of the considered algorithms. In their article (Powles and Selbst, 2017) the authors stress the importance of providing meaningful descriptions to an individual which may not have a technical knowledge about ML. The authors further interpret “meaningful information” to be flexible, meaning that the controller can express the logic of a ML application without describing the actual algorithms.

Moreover, the authors Gutwirth et al. (2017, p. 101) stress that human involvement is necessary in a decision-making process which utilizes results from a ML application; it is stated in Article 22(1) in the GDPR that a decision cannot be completely based on automated processing and profiling. This means that a human needs to be involved in, e.g., the profiling process; decisions cannot be made solely based on the results of a ML application (ibid., p. 101).

3. Supervised machine learning – Classification

Classification is a supervised ML technique, which is "... the task of learning a target function f that maps [...] $\mathbf{x}$ to one of the predefined class labels y" (Kumar et al., 2014, p. 146). It is a widely used concept that can be applied in different contexts; from detecting spam email messages, to identifying cells as either malignant or benign from MRI scans (ibid., pp. 145–149). Each record in the data set consists of features, also named as input variables, ($\mathbf{x} = [x_1, x_2, ..., x_n]$) and a target variable (y), which consists of C number of classes ($C = 1, 2, ..., m$).[2] The general methodology in classification is to, first, learn a ML algorithm (also denoted as classifier) with specific hyperparameters on a training data set, which is a subset of all the available records. The algorithm is then applied on a test data set: another, but smaller, subset of the data set which is used as the final evaluating of a classifier's performance.

The key objective in the ML process is to reach a high generalizability, a classifier will otherwise not be able to predict the classes properly when new records are considered (ibid., p. 148). This can be achieved with different approaches. Firstly, hyperparameter tuning can be utilized to obtain a model with the optimal hyperparameters from the training data set, which is the process of evaluating the model with a combination of different hyperparameter values (Lindholm et al., 2021, pp. 110, 238). Secondly, applying more than one classifier on the data set. This is useful to identify the best performing ML algorithm when different complexity levels are considered. Thirdly, k-fold cross validation, a method to identify the average performance of a ML algorithm of different validation data sets in the training data (ibid., p. 61). Fourthly, data resampling techniques, which is a method of increasing the balance between the classes in the training data set (Fernández et al., 2013). Finally, feature selection using mutual information (MI): the method of identifying the most useful input variables in a data set (Kumar et al., 2014, p. 3).

These topics will be explored in the following sections. Five classifiers are discovered and how to tune the hyperparameters are included in those sections as well. Further techniques as k-fold cross validation, data resampling and feature selection will be covered separately. Additionally, a section about how to evaluate the performance of a classifier will be provided.

3.1 Logistic regression

The first considered classification model is the logistic regression (LR); a model, $g(\mathbf{x})$, which labels records by using the conditional class probabilities $p(y|\mathbf{x})$ (Lindholm et al.,

[2] Binary classification, when $C = 2$, is only considered hereinafter.

2021, p. 44). Since only two classes are considered, y can either be 1 or -1, and $g(\mathbf{x})$ can be expressed as (Lindholm et al., 2021, p. 44)

$$g(\mathbf{x}) : p(y = 1|\mathbf{x}), \text{ and} \tag{1}$$

$$1 - g(\mathbf{x}) : p(y = -1|\mathbf{x}). \tag{2}$$

The LR for the positive and negative class can be defined as (ibid., p. 45)

$$g(\mathbf{x}) = \frac{e^{\boldsymbol{\beta}^T \mathbf{x}}}{1 + e^{\boldsymbol{\beta}^T \mathbf{x}}}, \text{ and} \tag{3}$$

$$1 - g(\mathbf{x}) = \frac{e^{-\boldsymbol{\beta}^T \mathbf{x}}}{1 + e^{-\boldsymbol{\beta}^T \mathbf{x}}}, \tag{4}$$

where $\beta^T \mathbf{x}$ represents the input variables and their respective parameter

$$\boldsymbol{\beta}^T \mathbf{x} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n. \tag{5}$$

The LR is constructed by replacing the input variable, s, in the logistic function $h(s) = \frac{e^s}{1+e^s}$ with $\beta^T \mathbf{x}$, which maps the output values to the interval [0,1] (ibid., p. 45). Learning the LR is the process of predicting the parameter values, and it can be expressed as

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{j=1}^{n} \ln(1 + e^{-y_j \beta^T \mathbf{x}_j}) \tag{6}$$

which is achieved by solving the cross-entropy loss (ibid., p. 46). Predicting a record to a specific class is based on whether the conditional probability is either above or below (or equal to) the decision threshold, r (often set to $r = 0.5$)

$$\hat{y} = \begin{cases} 1 & \text{if } g(\mathbf{x}) > r, \\ -1 & \text{if } g(\mathbf{x}) \leq r. \end{cases} \tag{7}$$

One approach to tune the LR is to introduce regularization: a method to avoid overfitting (ibid., p. 52). The idea is to prioritize the most meaningful features for the model by penalizing the features which are the least meaningful. The regularization parameter, ρ ($\rho \geq 0$), is a hyperparameter that can be modified, and it affects the influence of the parameters of the input variables. When L_1- or L_2 regularization is utilized, a penalty term will be added to equation (6), where ρ is included, and this penalty term can be calculated in different ways. In L_1 regularization, the sum of the absolute values of the parameter values are included (Friedman et al., 2009, p. 125)

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{j=1}^{n} \ln(1 + e^{-y_j \beta^T \mathbf{x}_j}) + \rho \frac{1}{2} \sum_{j=1}^{n} |\beta_j|, \tag{8}$$

whereas in L_2 regularization, the penalty term is calculated by taking the sum of the squared parameter values (Lindholm et al., 2021, p. 53)

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n} \sum_{j=1}^{n} \ln(1 + e^{-y_j \beta^T \mathbf{x}_j}) + \rho \frac{1}{2} \sum_{j=1}^{n} \beta_j^2. \tag{9}$$

3.2 *k*-nearest neighbors

The next considered ML algorithm, the k-nearest neighbors (k-NN), is a non-parametric method which classifies a record based on its location in relation to the k ($k = 1, 2, ..., p$) nearby records (Kumar et al., 2014, pp. 224–226). The classification of a record is based on majority voting: when most neighboring records are, e.g., in the positive class, the record will be labeled in the same category, and vice versa. This can be expressed as

$$\hat{y} = \arg\max_{c} \sum_{j=1}^{k} I(c = y_j). \tag{10}$$

where c represents a class label, k is the number of neighboring records, y_j is the class of one of the nearest neighbors to a record and $I(\cdot)$ is an indicator function which returns 1 or 0 when the argument is true or false (ibid., pp. 225–226). In addition, when there are equally many neighbors considered, the record is classified randomly.

There are different approaches to tune the hyperparameters in the k-NN. To begin with, instead of handling the influence of all the nearest neighbors equally, the distance between the record and its neighbors can be utilized (ibid., p. 226). This method, distance-weighted voting, classifies a record based on the closest nearest neighbors, rather than the total number of nearest neighbors. To do so, the weight term w, which represents the influence by taking the inverse of the distance between a record and a nearest neighbor, is included in equation (10)

$$\hat{y} = \arg\max_{c} \sum_{j=1}^{k} w \times I(c = y_j). \tag{11}$$

Also, the number of k neighbors is another hyperparameter to control. When k is a small value, the algorithm can be sensitive to noise, which, in that case, leads to overfitting (ibid., p. 226). Meanwhile, the algorithm may be underfitting in the opposite situation when k is a large value.

3.3 Decision tree

The third considered model is the decision tree (DT): a non-parametric, rule-based method, where the key idea of this algorithm is to classify records based on the combination of features for specific threshold values (Kumar et al., 2014, p. 150). The process of a DT is to let it 'grow'; meaning to first introduce a root node and, from this, construct child, or internal, nodes. The combination of features for specific values are done by splitting the 'branches' in a DT. This is performed by, first, calculating the proportion of records in a node for a candidate

$$\alpha = \frac{1}{n}\sum_{j=1}^{n} I(c = y_j), \tag{12}$$

where n are the total number of records in a split, c is a specific class and y_j is the class label of a sample (Lindholm et al., 2021, p. 31). Afterwards, the impurity for all classes, C, is evaluated (here using the Gini impurity measure)

$$I = \sum_{c=1}^{C} \alpha(1 - \alpha). \tag{13}$$

The impurity of candidate features, with a specific value, are compared, and the one with the lowest value is chosen to split a node (ibid., p. 31). The process of this algorithm starts by identifying one input variable as the root node (Kumar et al., 2014, pp. 158–165).[3] Thereafter, internal nodes will be recursively constructed where a chosen input variable is evaluated in the same manner as for the root node. Finally, the leaf node occurs, which is the final stage of a branch in a DT, and the considered records (n) at this stage are classified in a category based on majority voting

$$\hat{y} = \arg\max_{c} \sum_{j}^{n} I(c = y_j). \tag{14}$$

The termination of the tree-growing process can be done in different ways. This can happen when the records in an internal node belong to the same class or when these share the same feature values (ibid., pp. 165–177). The expansion of a branch also stops when the impurity level of a child internal node is higher than a parent internal node. Moreover, this can also be achieved by the design of the algorithm as well. A DT contains of different hyperparameter values, the maximum depth and minimum samples in the leaf are two of them. The former sets how deep a DT can be, meaning how many nodes there can be in

[3] When the class distribution is equally large in a binary context – i.e., the records are equally distributed in both the negative and positive classes – the node has the highest possible impurity. Meanwhile, when 100% of one class appears in a node, the impurity is 0 – the lowest possible value.

one branch. When the maximum depth of a DT is high, the algorithm is then susceptible to noise, and vice versa. Regarding the latter, it states how many records there can be in a leaf node. Also, the total number of records for splitting a node can be tuned.

3.4 Random forest

Random forest (RF) is the fourth considered model, which is an ensemble method that generates multiple DTs using the bagging resampling technique (Friedman et al., 2009, p. 588), where the main idea with this approach is to train a set of unique models which all contribute for understanding the relationship between the target and input variables (Lindholm et al., 2021, p. 135).

In the RF, the overall correlation is reduced since a set of DTs are randomly generated (Friedman et al., 2009, p. 588). The algorithm is constructed by first generating N specified data sets where the records are randomly selected with replacement. A DT is applied on each data set, where the internal nodes are chosen in the same procedure as before however from a random subset of the features $l \leq L$ (often set to $l = \sqrt{L}$). This process is applied for all the N constructed DTs, and classifying a record is done by the taking majority vote of all the generated trees

$$\hat{y} = \arg\max_{c} \sum_{n}^{N} I(c = \hat{y}_n). \tag{15}$$

Since the RT utilizes DTs, the previous hyperparameters can be tuned in this algorithm as well (ibid., p. 589). In addition, it is also possible to modify the number of trees considered, since the performance of the model increases when the number of DTs increase (however it usually stabilizes when it exceeds a certain number of trees).

3.5 Multilayer perceptron

The last considered supervised ML model is the multilayer perceptron (MLP), which is a type of artificial neural network that has a feed-forward architecture (Chen et al., 2010). It is one of the most used artificial neural networks for classification and regression which can map nonlinear relationships. The MLP consists of neurons, also known as nodes, in different layers, and the neurons between two layers are fully connected (Figure 2).

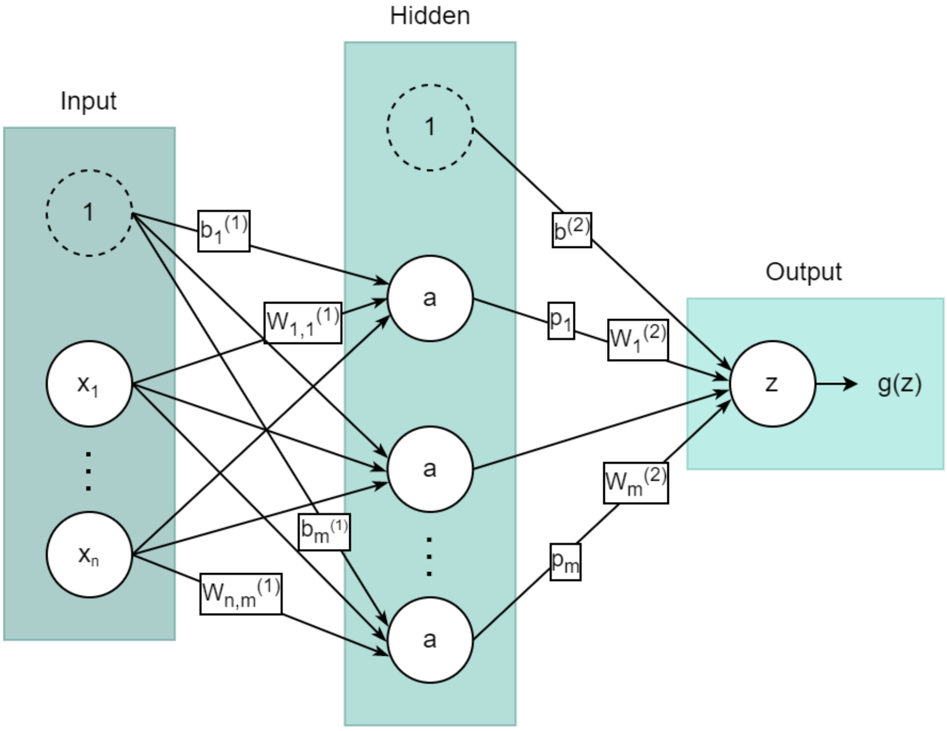

Figure 2. The MLP architecture with one hidden layer.

The architecture of a MLP with one hidden layer consists of three sections. The first one, the input layer, contains nodes representing the input variables considered for the model (Lindholm et al., 2021, pp. 113–115). The features, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$, are multiplied with the weights

$$\mathbf{W}^{(1)} = \begin{bmatrix} W_{1,1}^{(1)} & W_{1,2}^{(1)} & \cdots & W_{1,m}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ W_{n,1}^{(1)} & W_{n,2}^{(1)} & \cdots & W_{n,m}^{(1)} \end{bmatrix} \tag{16}$$

using the dot product and this information is sent to the corresponding nodes in the MLP. In addition, bias values, $\mathbf{b}^{(1)} = [b_1^{(1)}, b_2^{(1)}, ..., b_m^{(1)}]$, are added to all the in-going signals for each node in the next section (ibid., p. 115). The hidden layer consists of neurons containing activation functions, a, which map the input signals to new values using a transformation function. The outgoing signals from all the activation functions a can be expressed as

$$\mathbf{p}^{(1)} = a(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}). \tag{17}$$

Equation (17) is then multiplied with the weights $\mathbf{W}^{(2)} = [W_1^{(2)}, W_2^{(2)}, ..., W_n^{(2)}]$ and added with the bias signals $\mathbf{b}^{(2)} = [b^{(2)}]$, which results in the output signal $\mathbf{z} = [z_1, z_2, ..., z_m]$

(Lindholm et al., 2021, p. 115)

$$\mathbf{z} = \mathbf{W}^{(2)}\mathbf{p}^{(1)} + \mathbf{b}^{(2)}. \tag{18}$$

To classify a record in a binary context, the output is thereafter transformed to a class probability using the sigmoid function of a signal z (Bishop, 2006, p. 228)

$$g(z) = \frac{1}{1 + e^{-g(z)}} \tag{19}$$

where a record is categorized as either 1 or -1 based on this conditional probability as in equation (7).

Since the MLP is a parametric algorithm, the optimization problem

$$\hat{\beta} = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) \tag{20}$$

is solved by minimizing the cost function in the training data $J(\boldsymbol{\beta})$ (Lindholm et al., 2021, pp. 119–120). However, the MLP does not have a closed form solution. Thus, it must be optimized numerically, where the parameters are iteratively updated until a certain point has been reached. One common method to perform this is the backpropagation algorithm, which computes $J(\boldsymbol{\beta})$ and the gradient with respect to the parameters. The algorithm consists of two parts. In the first step, the forward propagation, the cost function $J(\boldsymbol{\beta})$ is calculated, whereas the backward propagation is initialized where $J(\boldsymbol{\beta})$ with respect to **z** and **p** (the gradients) is calculated in the next step

$$d\mathbf{z} = \left[\tfrac{\partial J(\boldsymbol{\beta})}{\partial z},\right] \text{ and } d\mathbf{p} = \begin{bmatrix} \frac{\partial J(\boldsymbol{\beta})}{\partial p_1} \\ \vdots \\ \frac{\partial J(\boldsymbol{\beta})}{\partial p_m} \end{bmatrix} \tag{21}$$

where computations start from the last to the first layer. When a specific criterion is met, the last predicted parameter $\hat{\boldsymbol{\beta}}$ will be used as the final parameters in the MLP (ibid., p. 119).

The hyperparameter tuning of a MLP with a single hidden layer can first be achieved by evaluating the algorithm by using different activation functions, and two common are the nonlinear functions: logistic ($h(s) = \frac{1}{1+e^{-s}}$) and ReLU ($h(s) = max(0, s)$) (ibid., pp. 114, 236). Additionally, the number of neurons in the hidden layer affects the performance of the model.

3.6 Techniques to increase the generalization capability

To increase the generalization capability in the ML process, additional methods can be utilized besides from hyperparameter tuning. These methods will be explored in the following subsections.

3.6.1 *k*-fold cross-validation

One approach within the ML field is to split the available data into a training and a test data set (Lindholm et al., 2021, pp. 58–62). By doing so, a ML algorithm can, first, be learned and evaluated on all the training records, where, e.g., the optimal hyperparameters can be identified. Then, the algorithm can be assessed on the test data set to see how it performs when new records are considered. However, when a ML algorithm is learned on all training records, the expected training error (E_{train}) tends to be less than the expected test error (E_{test}) received on the test data set. This gap can be reduced by utilizing the k-fold cross validation method applied on the training data set. The idea is to: (I) split the training data randomly into k equal sized batches, where $k-1$ batches are used as training data meanwhile one batch is used as the validation data, v, to evaluate the performance; (II) learn the model on the training batches and utilize the validation set to calculate the hold-out validation error (E_v); and (III) iterate this process k times to evaluate the model so all the batches have been used as validation data sets (Figure 3).

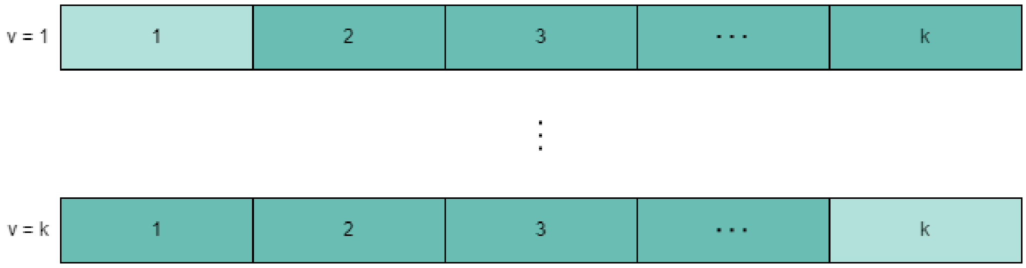

Figure 3. k-fold cross-validation.

When the process is finished, the k-fold cross-validation error

$$E_k = \frac{1}{k}\sum_{l=1}^{k} E_v \tag{22}$$

is obtained, which is the average error value on the validation data sets; this reduces the variance of the errors (ibid., pp. 60–61). In a context where the available data set is not considered to be sufficiently large, E_k provides a better estimate of E_{test} in comparison with E_{train}.

3.6.2 Data resampling

The proportion of one class is often significantly higher than the other(s) in a classification problem, which is known as the class imbalance problem (Fernández et al., 2013). Since classifiers tend to maximize the general performance, the algorithms often favor the majority class ("negative class"), whereas the performance of predicting the minority class, "positive class", is low. The distribution between the classes can be expressed by the imbalance ratio (IR), which is the ratio between the number records in the majority class and minority class. To handle the class imbalance problem, several methods can be utilized, e.g., algorithm-level techniques, cost-sensitive techniques, ensemble methods and data resampling techniques (Esposito et al., 2021), and the last mentioned will be considered hereinafter. Data resampling techniques is an approach to modify the training data set by reducing the imbalance between the classes (Fernández et al., 2013). Undersampling is one method, where the approach is to randomly remove a proportion of records from the majority class. Another method is the Synthetic Minority Over-sampling Technique (SMOTE); a more advanced method of oversampling (Bowyer et al., 2002). Instead of duplicating a sample of the minority class randomly, SMOTE generates synthetic data of the minority class. Additionally, a classifier's performance on the test data set can be increased when both undersampling and the SMOTE technique is combined.

3.6.3 Feature selection

In a ML context, identifying the most valuable input variables, i.e., feature selection, is necessary for increasing the performance of a classifier (Ding et al., 2005). The goal with feature selection is to maximize the performance of a ML algorithm by using a subset of features and, e.g., to avoid overfitting (Cheng et al., 2018). Feature selection can be divided into three main categories: filter, embedded and wrapper methods (ibid.).

In this thesis, a filter-based feature selection method has been applied, using MI. This method estimates the shared information that are found in two discrete random variables[4], and it can be used for measuring the MI between the input and target variable (Kastro et al., 2019). The entropy is used for measuring the amount of information stored, in bits, in one random variable X

$$H(X) = -\sum_{x} p(x) \log p(x) \tag{23}$$

where $p(x)$ is the probability distribution of X (Cover and Thomas, 1991, p. 5). To estimate the MI between two random variables, X and Y, the entropy is first applied to the

[4] Continuous random variables needs to be discretized (Cheng et al., 2018).

conditional probability of X given Y

$$H(X|Y) = -\sum_x \sum_y p(x,y) \log p(x|y) \tag{24}$$

and then calculated in the following manner

$$I(X;Y) = H(X) - H(X|Y) = \sum_x \sum_y p(x,y) \log \frac{p(x|y)}{p(x)}, \tag{25}$$

where $I(X;Y) \geq 0$ holds (Cover and Thomas, 1991, p. 21). Further, the Equation (25) is symmetric and can also be expressed as

$$I(X;Y) = H(Y) - H(Y|X). \tag{26}$$

If the MI score is 0, the random variables are independent (ibid., p. 7). Thus, the higher the value, the higher the dependence.

3.7 Metrics for evaluating the performance

In binary classification, the results from a ML model can be expressed in a confusion matrix (Table 1) (Lindholm et al., 2021, p. 75). There are four different outcomes, and a record can be predicted as a: true positive (TP), false negative (FN), false positive (FP) or true negative (TN). A record is classified correctly as either TP or TN and misclassified otherwise.

Table 1. Confusion matrix.

	$\hat{y} = 1$	$\hat{y} = -1$
$y = 1$	TP	FN
$y = -1$	FP	TN

From Table 1, several metrics can be used for, e.g., comparing the performance of different ML models (ibid., p. 76). One measurement is accuracy (ACC) which is defined by

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{27}$$

However, ACC may not always be appropriate for all data sets, since every class is treated as equally important (Kumar et al., 2014, pp. 295–297). This can be problematic when learning classifiers based on imbalanced data sets, where the number of classes are not equally distributed between the records. To handle this, other metrics can be utilized to measure the performance of a classifier. To begin with, positive predictive value (PPV), precision, and true positive rate (TPR), recall, can first be utilized to evaluate the performance of predicting the positive class.[5] The first mentioned metric is expressed as

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{28}$$

where the number of correctly predicted records, TPs, are divided by the true and false positive predicted records (Kumar et al., 2014, p. 297). The higher the value of PPV is, the lower the number of FPs exists in the predicted records, and vice versa. The second metric

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{29}$$

considers instead the proportion of the TPs that are classified both correctly and incorrectly (ibid., p. 296). The higher the value of TPR, the fewer number of FNs occurs in the predicted records, and vice versa.

Moreover, another measurement, F_β, utilizes both the TPR and PPV scores (ibid., pp. 297–298). If both metrics are equally important in a classification problem, $\beta = 1$ is utilized

$$F_1 = \frac{2}{\frac{1}{\mathrm{PPV}} + \frac{1}{\mathrm{TPR}}}. \tag{30}$$

The F_1 measure is a representation of a harmonic mean (ibid., pp. 297–298): when one of the two variables is lower than the other, the value of the F_1 measure will have a score closer to the variable with the lower value.[6]

In addition, to evaluate the performance of classifying the negative class, both negative predictive value (NPV) and true negative rate (TNR), specificity, can be utilized (Lindholm et al., 2021, p. 76). The former is calculated in the following manner

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}, \tag{31}$$

which compares the TNs with respect to the false labeled class. The latter, TNR, is calculated as

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}, \tag{32}$$

where only the negative class has been considered: either classified correctly or as a positive class (FP). Furthermore, the false positive rate (FPR)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}, \tag{33}$$

[5] The positive and negative class will correspond to the minority and majority class, respectively, hereinafter.

[6] When $\beta < 1$ PPV is favored, whereas the same occurs regarding the TPR when $\beta > 1$.

is calculated by taking the true falsely predicted records divided by the sum of the same records and the TNs (Lindholm et al., 2021, p. 76).

Additionally, to evaluate the performance of predicting the positive class with different decision thresholds in an algorithm, area under the curve (AUC) is a useful metric (Davis and Goadrich, 2006). Firstly, it can be used for analyzing the relationship between the TPR and FPR visualized in a plot; also known as the Receiver operator characteristic (ROC) curve. The optimal classifier has an AUC (also known as AUC-ROC) score equal to 1. Meanwhile, the performance of a random classifier (RC) – when a ML algorithm predicts the classes randomly – is 0.5; this occurs when the TPR and FPR of a classifier is the same for all the decision thresholds. Secondly, it can be used to assess the relationship between the PPV and TPR for all decision thresholds, which also can be visualized as a plot – Precision-recall (PR) curve (ibid.). The optimal AUC score for this metric (also known as AUC-PR) is also 1 (Lindholm et al., 2021, p. 76). In this context, a RC is a model which has a PPV score equal to the prevalence

$$\text{Prevalence} = \frac{\text{TP} + \text{FN}}{n} \tag{34}$$

for all TPR scores, where n is the number of records in the data set (ibid., p. 76).

4. Data

In this Chapter, the concepts regarding the company's customer relationship management and Eloqua databases – which store the utilized data in this thesis – are explored as well as the data itself is introduced. The web data collection of web users is also elaborated. Finally, a subsection about how the customers are pseudonymized in this thesis is presented.

4.1 Customer relationship management data

The company has adopted a Customer Relationship Management (CRM) concept, which is a strategic process with a long-term focus of in part providing value to the customers, and partly to gain insights of the customer relationships (Kumar and Reinartz, 2018, pp. 5, 35). To achieve this goal, information about the customers' needs to be recorded and can be stored in a CRM database; a system that stores profile and purchase behavior data of the customers. Profile data covers how the customer has reacted to campaigns[7], whereas purchase behavior can contain customer types, such as "existing customers", "prospects" and "defectors" (ibid., p. 212).

The company has a CRM system that stores information about the customers that have been a potential buyer for a unit or a project. A unit is a specific accommodation, whereas a project is a collection of units that are in the same area. Information about a customer with regards to its status and registration of interest to a specific project is registered in the CRM system. The purchase behavior is expressed in the company's SP, which is a filtering process to find an appropriate customer – or prospect – for a specific unit. The process contains several steps (Figure 4). It is initiated when a customer is defined as a prospect for a project. This can be done either before the sales start of units in a project, New - Before Sales Start (NBS), or after sales start, which is known as New - After Sales Start (NAS). The next step is to evaluate if a customer accepts the price of a unit. If the prospect accepts it, Sales Accepted (SA), the person will, later, be qualified as a candidate, Sales Qualified (SQ), for the project. If the customer does not accept it, Not Sales Accepted (NSA), it is still possible that the company accepts another price of the customer. After being a SQ candidate, a prospect has chosen a specific unit to purchase. However, there can be a queue for a certain unit and, in this case, the customer needs to be in an ordered reservation list, Name Reservation in Queue (NRQ), for the unit. When the customer is first in line, Name Reservation (NR), it is possible to purchase the unit of interest. During this phase, Signed Reservation (SR), the prospect cannot have active signed reservations for other units. When this is finished, the customer purchases, Sold (S), the unit and is defined as an existing customer in the CRM system. In addition, it is possible to lose the

[7] A collection of actions of a company to market or promote, e.g., products (Kumar and Reinartz, 2018, p. 208).

SP, Lost Sales Process (LSP), and there are different reasons for this. For example, the defector: chooses to quit the process; does not find the prices suitable (after NSA); or loses it because another customer has bought the same unit. Also, the company might have chosen another customer for a specific unit. Hereinafter, a buyer is referred to a customer that has purchased a unit, meanwhile a customer which has lost a SP is represented as a non-buyer.

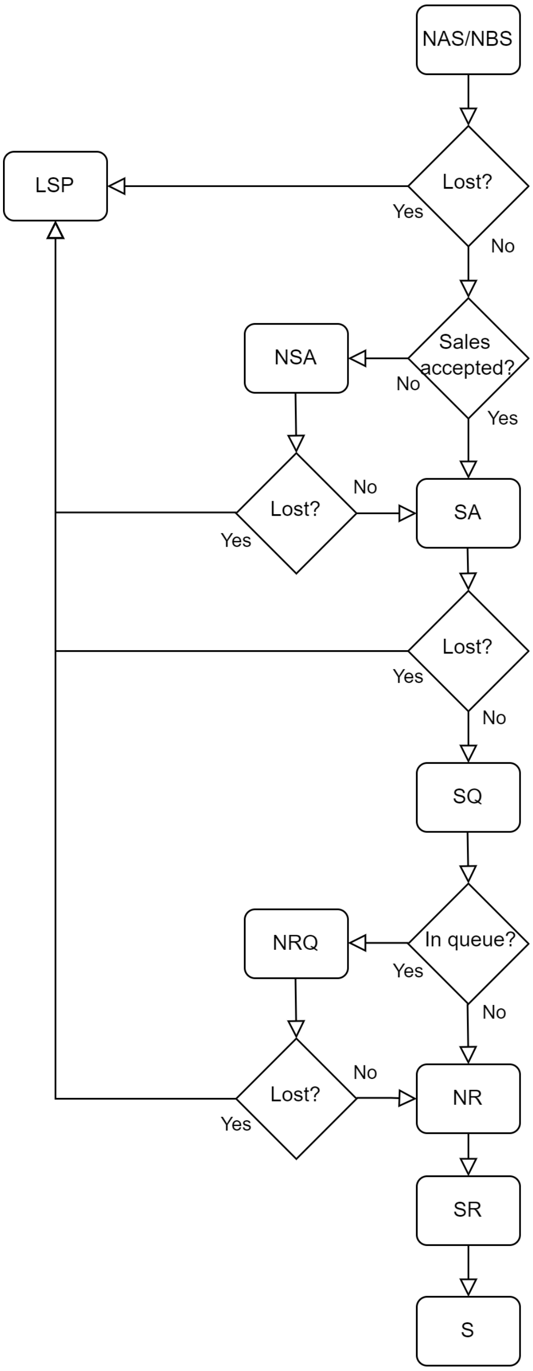

Figure 4. The plausible scenarios for a customer in the SP.

The CRM system also contains information about the specific units[8] and their connected project. Information about the projects such as municipalities, average price per unit and average price (per square meters) per unit are available.

4.2 Eloqua data

Eloqua is a Software as a Service (SaaS), a marketing automation tool for targeting customer groups with different campaign approaches (Griffith, 2013, p. 23). With this platform, data such as customers' responses and web activities can be recorded and used as the basis for modifying the campaigns to increase the probability of achieving new existing customers. The overall goal with using Eloqua is to maximize the understanding of prospects and existing customers to better satisfy these customer groups (ibid., p. 43). Eloqua can be used as a marketing tool independently of other information systems but it can also be integrated to a CRM database, which enables it to store more information about the customers (ibid., pp. 125–126).

The company uses an Eloqua database which covers information about the interaction between the company and its customers. Information about a customer's SP is not recorded here, instead it covers, e.g., a customer's web activity on the company's web site. Data regarding the actions (on the company's web site) of the customer in this database is solely included. A customer action is defined as: sending a registration of interest; accepting a project newsletter, a general newsletter or the company to reach the customer through SMS; sending a registration of interest; or browsing on the web site by clicking on web pages (Table 2).

Table 2. An example of the activity of three web users visiting the web site.

Web user ID	Web page URL	Timestamp
1	https://www.company.se/bostad	2019-07-29, 15:07:43
1	https://www.company.se/nyproduktion	2019-07-29, 15:05:34
1	https://www.company.se/bostad/stockholm/nacka	2019-07-29, 15:05:41
2	https://www.company.de/immobilien	2019-08-1, 10:24:26
2	https://www.company.de/konto/interessen	2019-08-01, 10:29:31
1	https://www.company.se/bostad	2019-08-01, 12:03:23
1	https://www.company.se/bostad/stockholm/nacka/tollare	2019-08-01, 12:04:10
3	https://www.company.no/bolig	2019-08-02, 19:32:39

[8] However, since a unit often is associated with a SP during NR. In addition, customers tend to send their registration of interests to projects rather than specific units. Thus, information regarding the projects is only utilized.

Moreover, to record the visitor's behavior on the web site, cookies are stored on the client machine during a specific period (Bonava, n.d.). According to the company, the web visitor can choose which types of cookies that will collect information about the visitor. The first option is to activate or deactivate "functionality cookies". The data generated through these are either collected by the company or a third-party vendor. These cookies are, e.g., used to improve the functionality of the web site. Another option is to accept or decline "marketing cookies". If these are accepted, third party vendors can store information about the web user, such as: visits on either the company's web site or other web sites that do not belong to the company, reactions on advertisements and emails. The company and the third-party vendors can use this information to personalize the advertisements, also known as "interest-based marketing". The visitor can also choose to activate or deactivate "analytics cookies"; this enables the company to analyze the performance of its web site. Besides the optional cookies, the web visitor must accept the "strictly necessary cookies". If these cookies are blocked or deleted, the web user cannot access the whole web site, since some of these are personal web pages which require identification of the visitor (ibid.).

When the marketing cookies on the web site are active, the company utilizes Eloqua to retrieve information, such as the web history, of the visitors. The company will also collect this information when the web user is considered as a "company customer", and this occurs when the web user either has subscribed to one of the newsletters or sent a registration of interest to a specific project (ibid.). Hereinafter, a customer is only referred to a company customer.

4.3 Pseudonymization

In CRM and Eloqua, the ID of the customer is a hashed sequence of the person's email address. This hash string is then used as the primary key – or the pseudonym – of the customer. By doing so, personal information connected to the customer was discarded in the data sets. In this thesis, the pseudonym is the only information that is connected to a specific customer: this to preserve the uniqueness of the customers. The projects and units were pseudonymized as well.

5. Method

In this chapter, the utilized software tools are first presented followed by a description about the developed ML framework. Next, the data filtering process in this framework is thereafter explored. The data cleaning section is thereafter introduced followed by the utilized data resampling techniques. Lastly, the considered hyperparameters for all the classifiers are presented.

5.1 Software tools

Different frameworks and libraries have been utilized for the constructed ML framework. U-SQL (Scott, 2017) has been used in the data filtering phase, whereas Python (Drake and Van Rossum, 2009) was applied during the rest of the process by using: Pandas (The pandas development team, 2021) for managing the structured data sets; NumPY (Abbasi et al., 2020) for computing and modifying data structures; Imbalanced-learn (Aridas et al., 2017) for applying the data resampling technique SMOTE on the training data; Scikit-learn (Blondel et al., 2011) for applying the ML models on the data; and Matplotlib (Hunter, 2007) for visualizations.

5.2 The machine learning framework

The ML framework (Figure 5) began with data filtering as the first step, to include the relevant customers in each generated data set. The next step was to preprocess the plausible features and clean the data for increasing the generalization capability. Then, each data set was split into a training and test data set, where data resampling was applied on the former data set to reach the same class ratio. Subsequently, the features were filtered based on the MI scores received on the training data sets (the MI threshold was set to 0.05). Afterwards, all models (LR, k-NN, DT, RF and MLP) were trained and evaluated by the F_1 score, which was the average validation result using the k-fold cross validation technique (where k=10). The training and evaluation process was repeated for each model since hyperparameter optimization was utilized, an approach to recursively evaluate all the considered hyperparameter values to identify the optimal ones for each model. Finally, each classifier was applied on the test data sets, which contained the same features that were selected in the training data sets.

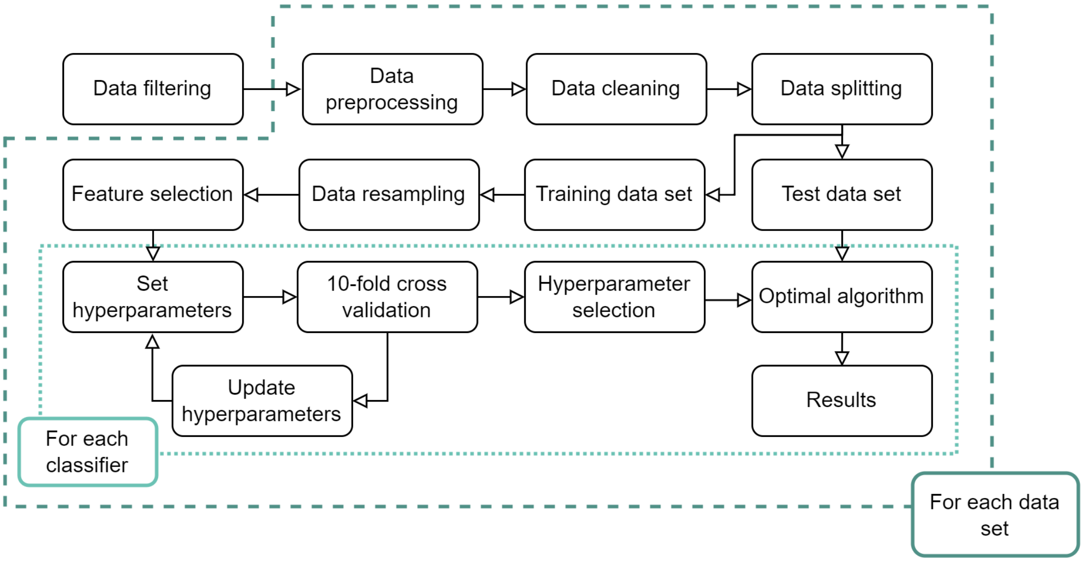

Figure 5. The ML framework.

Furthermore, regarding the web usage mining process (Figure 1), the data preparation phase occurred, to begin with, when the data was filtered and preprocessed in the developed ML framework, where the construction of the user transactions was done in the data preprocessing step. The data preparation phase also covered the data cleaning and splitting as well as data resampling and feature selection on the training data sets. The pattern discovery phase covered in part the training of the ML algorithms, partly the evaluation of the models on the test data sets. Finally, the pattern analysis step covered the reasoning about the results.

5.3 Data filtering

In this section, the filtering process of selecting the customers is presented. The process of defining the buyers and non-buyers was first done by solely using the data from the CRM database (Section 5.3.1). The next step was to consider the web activity of the customers; the Eloqua data connected to each customer was then utilized (Section 5.3.2). Lastly, the method of generating the data sets used in the ML framework is explored.

5.3.1 Defining buyers and non-buyers

Only two customer groups were of relevance: existing customers (buyers) and defectors (non-buyers). The prospects were discarded since these were not categorized in a customer group and could, thus, not be used by the classifiers.

To delimit the customers, additional assumptions have been made. To begin with, the customers could only have the intentions of purchase units for private purposes. Therefore, investors, such as companies, were not included. Furthermore, it is possible that a customer has been involved in more than one SP; a buyer could have purchased one or more units, meanwhile a non-buyer could have lost one or more SPes. A buyer was however only referred to a customer that had purchased only one unit, where that SP was connected to the customer. Customers that have bought more than one unit were discarded, since this would result in several connected SPes to one person (and thus cause duplicate customers). To maintain unique people in the other customer group, the latest lost SP was only considered.

In addition, since the company is a spin-off from another company since 2016, data about the historical customers have been migrated to the company where the time spent in a SP was not recorded before. When there was no existing time span of the SPes available for buyers and non-buyers, these customers could not be utilized, and it was found that a substantial proportion of the customers did not have recorded SP before June 2019. These were, thus, discarded. Finally, customers in Russia have not been included, since the SP in that country differs from all the other countries. The SP in Russia does not only involve the customer and the company; authorities are also included. A SP in Russia was thus deemed to contain an inertia, which made it difficult to compare this with SPes in other countries.

5.3.2 Defining customers based on their web activity

Another restriction is that a customer could only have been browsing on the company's web site, i.e., the customer, besides from being registered in the CRM database, also needed to be registered in the company's Eloqua database. If not, the customer was discarded. Also, a customer needed to have been active on the web site before the SP or during the SP at a certain point in time. Any web activity after the SP was discarded.

Since the company is established in several countries in Europe, it has a web site for each country, where these web sites share a common web site structure (Figure 6). Web pages ($p = 1, 2, ..., n$) related to the *Project search* page were of certain interest, since interacting with those pages was assumed to indicate a customer who was potentially interested in purchasing a unit. When a web site of a country did not contain a Project search page, the customers in that country were not considered. The web site for the customers in Denmark did not contain this page and, thus, those customers were discarded.

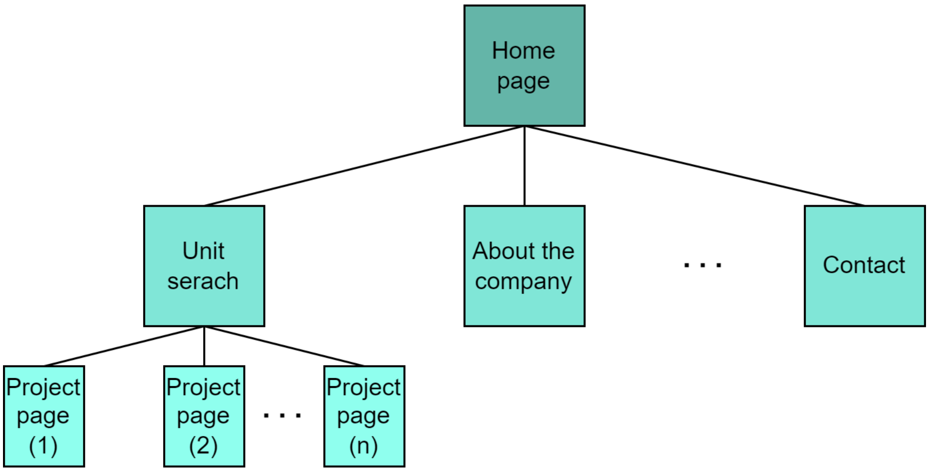

Figure 6. A simplified tree structure of the web site.

In total, customers from seven out of nine countries had a web activity either before or during the SP (Table 3). The majority of the customers, approximately 76%, were from Germany, and the other six countries constituted together approximately 24% of the customers, where the proportion of customers from Finland and Sweden were about 12% and 8%, respectively. The proportion of customers from the Baltic countries and Norway was approximately 4%.

Table 3. The total number, and the proportion, of customers per country.

Country	# of customers	% of customers
Estonia	193	1.30
Finland	1717	11.58
Germany	11317	76.32
Latvia	166	1.12
Lithuania	4	0.03
Norway	213	1.44
Sweden	1218	8.21

5.3.3 Generating the data sets

The generation of data sets was the last step in the data filtering process. A data set represents the number of available customers on a specific day D (the total number of days after their respective SP started), and it was crucial to identify when to apply the ML algorithms during the SPes. If a customer was included in a data set at day D, this would indicate that the customer would either be purchasing a unit or losing a SP in the future. However, since the purchase intentions were already known, the customers were separated into their respective customer groups – this to ensure that as many future buyers as possible were included in a data set.

In total, 1741 buyers and 11628 non-buyers (13369 customers in total) were considered in this thesis, which is an IR of approximately 1:7. In addition, 98.1% of the buyers had purchased a unit while 99.7% of the non-buyers lost a SP during a year. The other customers purchased and lost their respective SPes after that time period. The half-life, $t_{1/2}$, of customers transpiring into buyers and non-buyers were 49 and 46 days, respectively (Figure 7); more than half of the customers (in each customer group) either purchased a unit or lost a SP within 50 days of a SP.

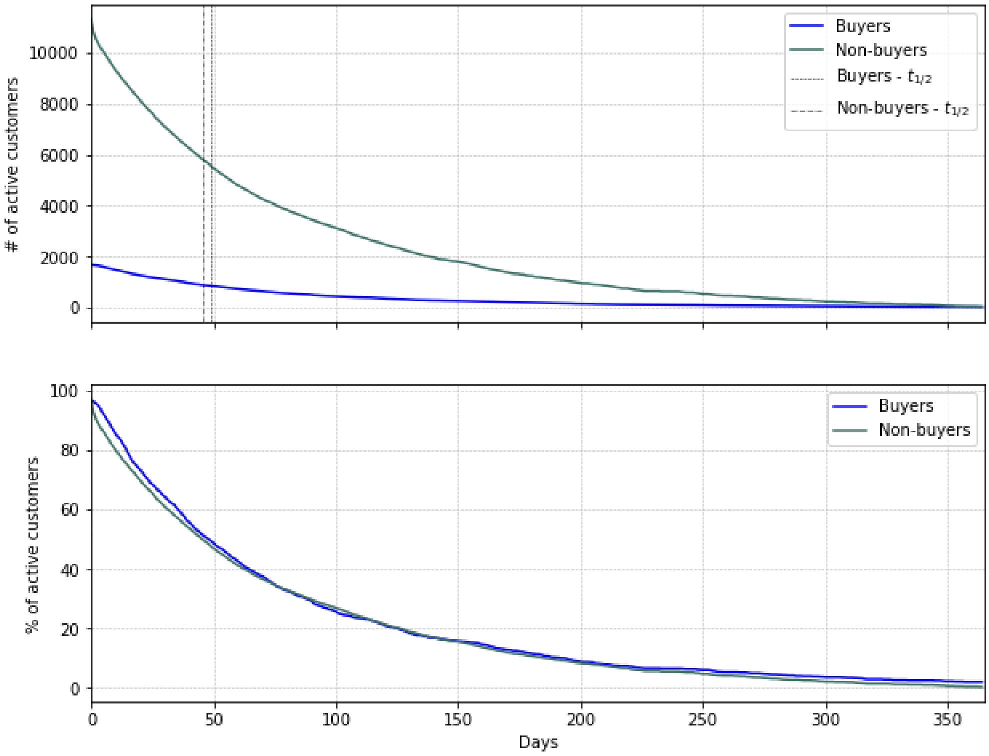

Figure 7. The number and proportion of customers that purchased units and lost SPes each day. The customers that remained along the x-axis were still involved in their SPes (active customers). (Top) shows the number of active customers in each customer group on different days. (Bottom) displays the proportion of the active buyers and non-buyers at certain points in time.

However, the customers' web activity was not considered in Figure 7. To include this, it was assumed that a customer must have been active on the company's web site 180 days before the SP started or until a certain, D, during the SP (Figure 8). If a customer had only been active on the company's web site during the SP, but after day D, the customer would not be included in a data set at day D. Also, a customer would be discarded if the web activity occurred more than 180 days before the SP (given that the customer did not browse on the company's web site during the SP).

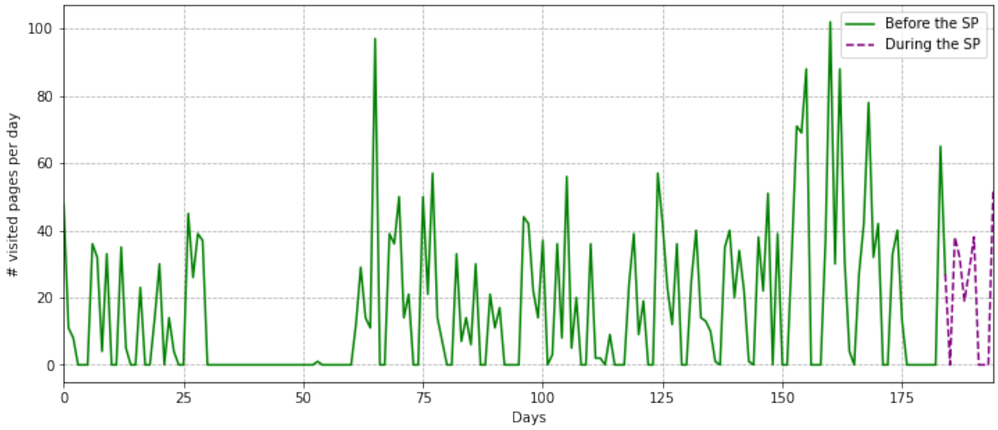

Figure 8. The web activity of one of the customers, which was both before and during the SP. This customer was then selected in a data set.

When the web activity was considered for each customer, the number of customers in both customer groups decreased (Figure 9). The customers that did not appear in Figure 9 either browsed on the company's web site 180 days before their respective SP started or visited the web site after their respective SP ended. In addition, the number of active customers increased for each customer group in the beginning of the SPes; this occurred since some customers had no web activity registered before the start of the SPes, and they began to browse on the company's web site during their respective SP. A difference between the future buyers and non-buyers could also be seen on the first day ($D = 0$): 35% of the customers who would lose a SP had an been browsing on the company's web site 180 days before the SPes started, whereas the corresponding proportion of the customers who would purchase a unit was 51%.

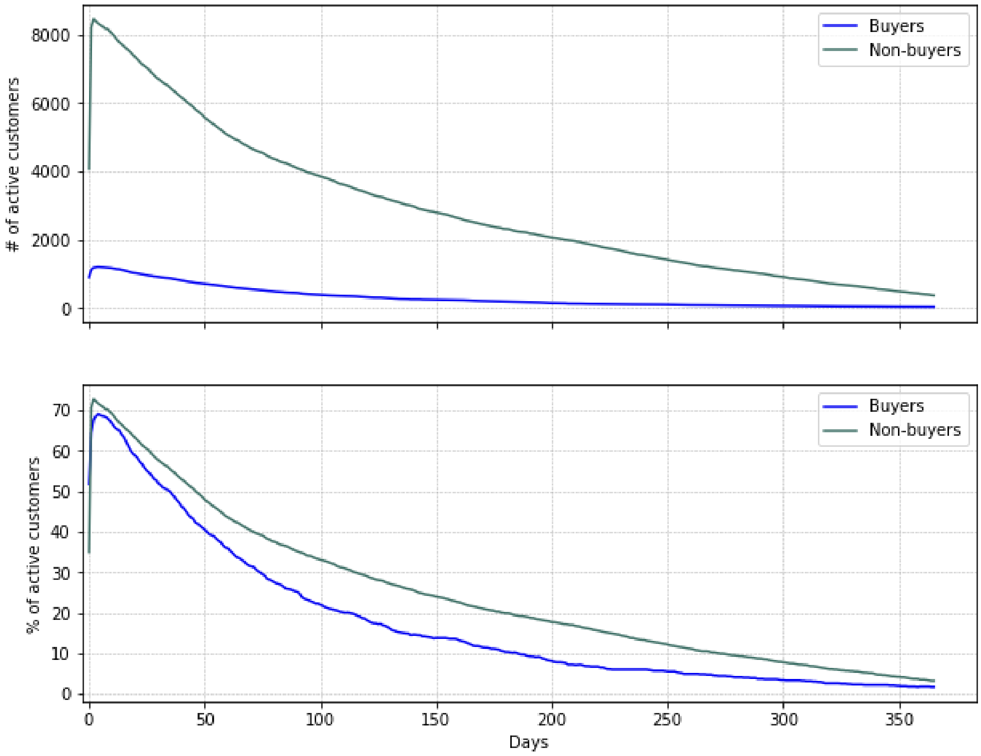

Figure 9. The number and proportion of customers that purchased units and lost SPes (regarding their web activity) each day, where the axis represents the number of active customers. (Top) shows the number of active customers on different days. (Bottom) displays the proportion of the active buyers and non-buyers at certain points in time.

Three data sets were generated in total. To begin with, two data sets included all the active customers 10 and 20 days after the SPes started. These were chosen before the number of available customers in each customer group reached their respective $t_{1/2}$. Furthermore, the company was interested to see whether the predictions of buyers and non-buyers could have been achieved on the first day. Because of this, a data set which included all the active customers on that day was generated. Thus, the considered days were $D = [0, 10, 20]$ (Table 4). These data sets represented the active customers at day 0, 10 and 20, and what they had been browsing on before those time points.

Table 4. The total number, and the proportion of, customers per country in each data set.

	# of customers			% of customers		
Country	$D = 0$	$D = 10$	$D = 20$	$D = 0$	$D = 10$	$D = 20$
Estonia	114	127	103	2.29	1.38	1.22
Finland	1010	1064	902	20.30	11.53	10.73
Germany	3055	7122	6673	61.41	77.19	79.35
Latvia	121	105	87	2.43	1.14	1.03
Lithuania	1	2	1	0.02	0.02	0.01
Norway	74	82	67	1.49	0.89	0.80
Sweden	600	725	577	12.06	7.85	6.86

In each data set, training and test data sets were generated. The number of records in each data set were split into these two sets: 70% and 30% of the customers were distributed to the training and test data set, respectively (Table 5); this to decrease the bias in the test data sets and, thus, to have a fair evaluation of the classifiers (Lindholm et al., 2021, p. 237).

Table 5. The total number of buyers and non-buyers in both training and test data sets.

	Training data sets			Test data sets		
	$D = 0$	$D = 10$	$D = 20$	$D = 0$	$D = 10$	$D = 20$
# of buyers	647	799	721	255	364	306
# of non-buyers	2835	5659	5166	1238	2405	2217
# of customers	3482	6458	5887	1493	2769	2523
Prevalence	0.23	0.14	0.14	0.21	0.15	0.14

5.4 Data preprocessing

The web activity, or the pageviews, for each customer occurred during different points in time. It was therefore necessary to represent these pageviews as user transactions for the classifiers. Thus, aggregation of the data has been applied using different methods depending on the data type of the feature.

5.4.1 Categorical features

The first considered input variables for the ML models were the nominal input variables: *Building type* and *Building category*. The former contained building types (Not available value (NA), Block of flats, Row house, Semi-detached house and Single family house), meanwhile the latter – which categorized the building type in high level categories – consisted of the nominal values: NA, Multi family, Single family. Both features were aggregated by using the nominal value associated with the project which was the most frequently visited project on the company's web site by a customer. Furthermore, the most frequently visited project on the web site did not necessarily mean the most frequent web page by the customer. If a customer did not watch a web page about projects, the nominal value was set to NA. In comparison, the building category and type of the most watched project was found in more than 45% to 55% of the buyers' activity, whereas the corresponding proportion for the non-buyers were approximately 30% in all data sets (Figure 10).

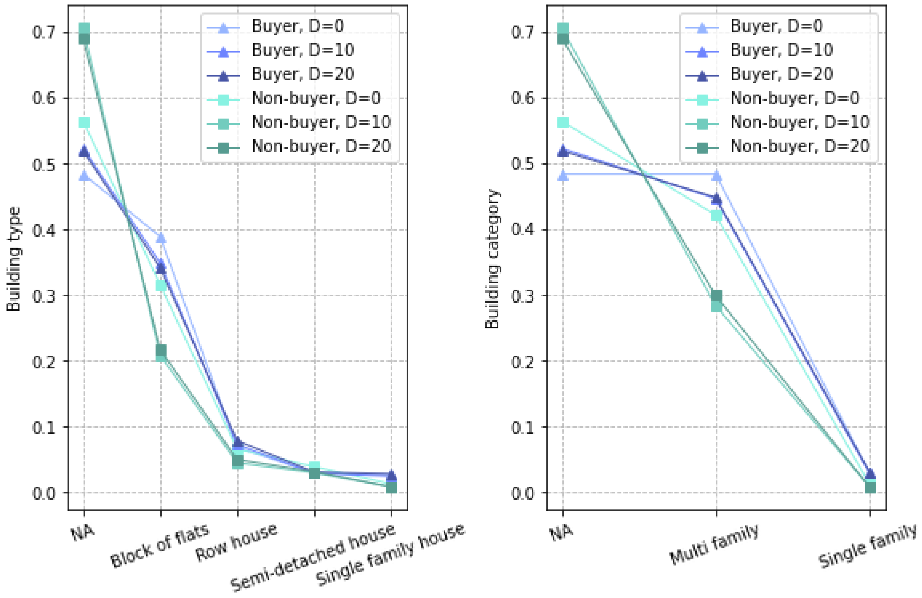

Figure 10. The proportion of each nominal value for each customer group in the data sets D = [0, 10, 20].

The other categorical variable type, binary features, was considered as well. The first input variable, *Most frequent location search* (MF location search), stated whether the location of the most frequently visited project on the company's web site was the same as the location of the project in the connected SP. The next input variable, *Most frequent project search* (MF project search), was developed in the same manner as the previous one, instead the category described whether the most frequently watched project on the company's web site was the same as the project which the customer had a connected SP to. Both features were developed in the same manner as the nominal values; the most frequently watched web page would not necessarily mean a Project search page. Moreover, the most frequently watched location was not the same as the location in the SP for the majority of the customer groups in the three data sets: between 55% and 65% of the buyers, and more than 70% of the non-buyers (Figure 11). However, the most frequently watched project was not the same as the project in the SP for almost all customers. The other three input variables, *Project newsletter*, *General newsletter*, and *SMS* were the actions the customers had taken on the company's web site. The input variables project and general newsletter described whether a customer had accepted to receive information via emails about the company's projects and general information, e.g., tip or inspiration regarding accommodations. The last feature, SMS, stated if the customer had accepted to receive information via SMS. Regarding the buyers, more than 60% did not accept to receive project newsletters, general newsletters and SMS. With respect to the other customer group, the corresponding proportion was more than 75%. In addition, all the binary features contained the boolean values True or False.

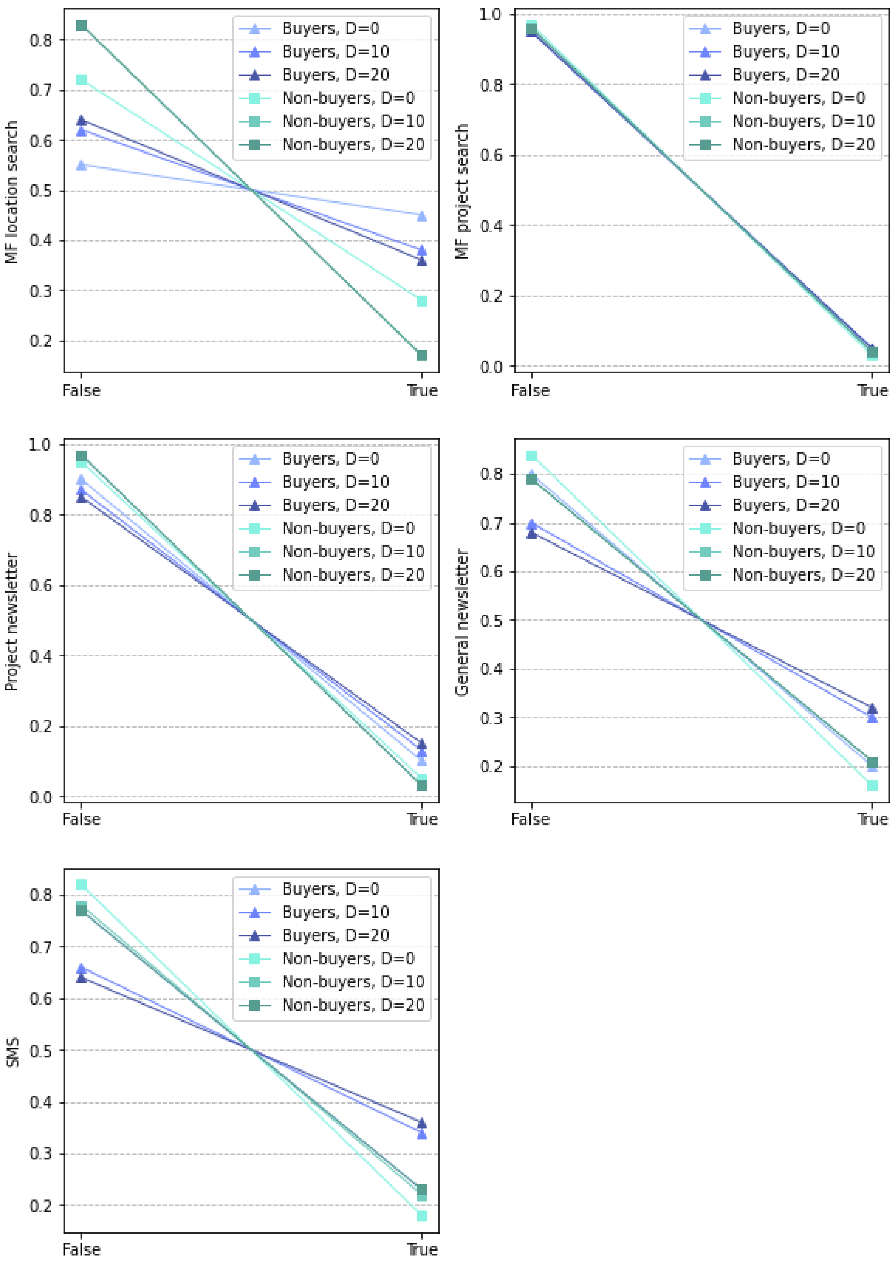

Figure 11. The proportion of each binary value for each customer group in the data sets D = [0, 10, 20].

5.4.2 Numerical features

The other feature type considered in this thesis were the numerical input variables. To begin with, the first numerical features were the continuous input variables (Figure 12) *Average price* and *Average price [m^2]*, which was the cost (in euros) of the average and the average per square meters of a unit, respectively, connected to the most frequently visited project by the customer on the company's web site. The activity was similar between the buyers and non-buyers: more than 80% of each customer group frequently visited a project on the web site which had an average cost between 0 to 400522 euros in total and 0 to 5917 euros per square meter. Additionally, if a customer did not visit a web page regarding a project, both features would contain the value 0. Furthermore, *Total duration* was the final continuous input variable, which represented the total time spent (in seconds) on the web site. The time spent on the web site was similar between the customer groups.

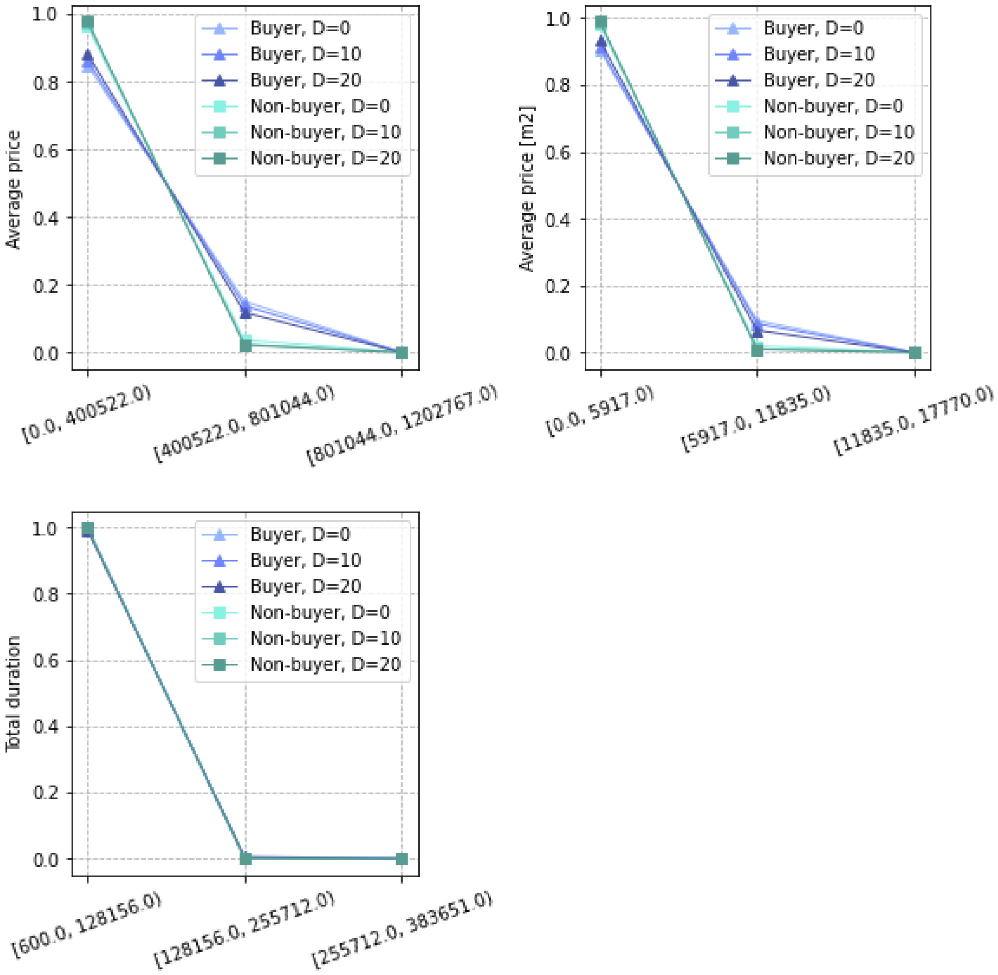

Figure 12. The proportion of continuous value for each customer group in the data sets D = [0, 10, 20].

The last input variables contained discrete values which was the total sum of the value for each feature. These were (Figure 13): *Sessions* – the total number of sessions; *Pages* –

the total number of visited web pages; *Interests* – the total number registration of interests sent; *Project search* – the total number of watched web pages regarding information about the projects; and *Location search* – the total number of watched web pages regarding information about the location connected to a project. The majority of the customers (more than 90% of each customer group in all data sets) had: visited less than ten projects, three locations and 631 web pages; started less than 135 sessions; and sent less than three registration of interests.

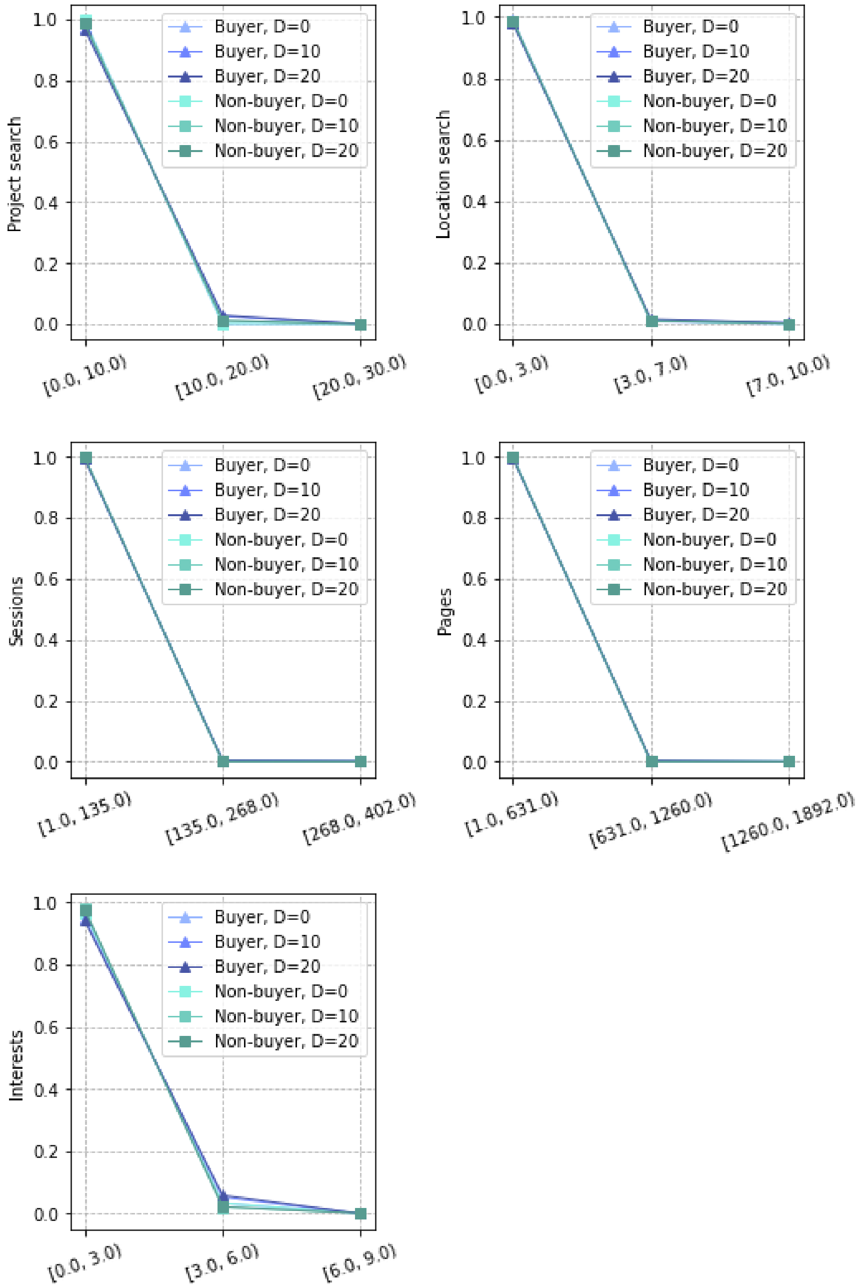

Figure 13. The proportion of discrete values for each customer group in the data sets D = [0, 10, 20].

The maximum and minimum value for each numerical feature with its standard deviation (SD) for each day $D = [0, 10, 20]$ is visualized in Table 6.

Table 6. The maximum and minimum value for each numerical feature with their respective SD in each data set D = [0, 10, 20].

	$D = 0$			$D = 10$			$D = 20$		
Feature	Min. value	Max. value	SD	Min. value	Max. value	SD	Min. value	Max. value	SD
Average price	0.0	$1.2 \cdot 10^6$	$1.3 \cdot 10^5$	0.0	$1.2 \cdot 10^6$	$1.1 \cdot 10^5$	0.0	$1.2 \cdot 10^6$	$1.0 \cdot 10^5$
Average price [m^2]	0.0	$1.8 \cdot 10^4$	$1.5 \cdot 10^3$	0.0	$1.8 \cdot 10^4$	$1.3 \cdot 10^3$	0.0	$1.8 \cdot 10^4$	$1.2 \cdot 10^3$
Total duration	600.0	$5.3 \cdot 10^5$	$1.7 \cdot 10^4$	600.0	$5.8 \cdot 10^5$	$1.3 \cdot 10^4$	600.0	$3.8 \cdot 10^5$	$1.3 \cdot 10^4$
Sessions	1	459	16.0	1	506	12.3	1	402	12.1
Pages	1	2917	75.2	1	3167	61.0	1	1890	55.5
Interests	0	8	0.8	0	9	0.8	0	9	0.8
Project search	0	64	2.7	0	29	2.1	0	30	2.2
Location search	0	14	1.0	0	10	0.9	0	10	0.9

5.5 Data cleaning

The numerical features were normalized using Z-score normalization (Table 7).[9] The lowest value for each numerical feature was close to the mean value for the same feature, whereas all the maximum values for all the input variables were considered to be relatively high; the lowest maximum value of all the features, Average price ($D = 0$), was 8.6 SD from its mean, whereas the largest minimum value of all the features, Location search ($D = 0$), was 0.9 SD from its mean.

Table 7. The numerical features after Z-score normalization with SD = 1 in each data set D = [0, 10, 20].

	$D = 0$		$D = 10$		$D = 20$	
Feature	Min. value	Max. value	Min. value	Max. value	Min. value	Max. value
Average price	-0.4	8.6	-0.3	10.6	-0.3	11.3
Average price [m^2]	-0.4	10.9	-0.3	13.7	-0.3	14.7
Total duration	-0.3	31.0	-0.3	44.1	-0.3	30.3
Sessions	-0.3	28.4	-0.3	40.9	-0.3	32.8
Pages	-0.3	38.4	-0.3	51.7	-0.3	33.8
Interests	-0.7	8.8	-0.7	10.7	-0.7	10.6
Project search	-0.6	23.0	-0.5	13.2	-0.5	13.2
Location search	-0.9	12.8	-0.7	10.6	-0.5	13.2

Also, one-hot encoding was applied on the categorical variables. Building type and Building category were split into five and three new features, respectively. The categorical input

[9] See Appendix A.

variables (MF location search, MF project search, Project newsletter, General newsletter and SMS) have only been binarized.[10]

5.6 Data resampling

The customer groups in the training data sets $D = [0, 10, 20]$ were imbalanced, where the negative class (non-buyers) constituted the majority of the customer groups (Table 5). To solve this, data resampling techniques – first undersampling and then SMOTE – were utilized to increase the IR in the training data sets. Before the data resampling techniques were applied, the IR of non-buyers and buyers in each data set ($D = [0, 10, 20]$) were approximately 1:4, 1:7 and 1:7, respectively (Table 8).

Table 8. The total number of customers in each customer group and the respective IR in all training data sets before data resampling.

Data set	# of buyers	# of non-buyers	IR
0	647	2835	$\approx 1 : 4$
10	799	5659	$\approx 1 : 7$
20	721	5166	$\approx 1 : 7$

When the data sampling techniques were applied, the IR reached 1:1 between the classes in the training data sets (Table 9).

Table 9. The total number of customers in each customer group and the respective IR in all training data sets after data resampling.

Data set	# of buyers	# of non-buyers	IR
0	2444	2444	$1 : 1$
10	4035	4035	$1 : 1$
20	3625	3625	$1 : 1$

5.7 Hyperparameter optimization

To optimize each classifier, hyperparameters with different values were considered for each model (Table 10). The LR consisted of either a penalty term in the cost function (L_1 or L_2) or the base model itself. Two hyperparameters were evaluated for the next algorithm (k-NN): how many neighbors k that were included (1 to 60), and weights that either handled the influence of neighbors uniformly or by their distance. The next model, the DT, was assessed by using a different number of maximum depths (from 2 to 10 with a step

[10] See Appendix A.

length of 2), minimum customers for splitting an internal node and minimum customers containing in a leaf node (from 5 to 30 with step length 5). The same hyperparameters were considered for the RF with the additional hyperparameter, which stated how many DTs that were included in the algorithm. Finally, the MLP consisted of a single hidden layer with either 10, 20 or 30 neurons. The activation functions could either be logistic or ReLU.

Table 10. The hyperparameters and the considered values for each classifier.

Model	Hyperparameter	Vector
LR	Penalty term ρ	[None, L_1, L_2] [0.001, 0.01, 0.1, 1, 10, 100, 1000]
k-NN	Weight k	[Uniform, Distance] [1,2, ..., 60]
DT	Max. depth Min. samples split Min. samples leaf	[2, 4, 6, 8, 10] [5, 10, 15, 20, 25, 30] [5, 10, 15, 20, 25, 30]
RF	Max. depth Min. samples split Min. samples leaf DTs	[2, 4, 6, 8, 10] [5, 10, 15, 20, 25, 30] [5, 10, 15, 20, 25, 30] [100, 200]
MLP	Neurons Activation function	[10, 20, 30] [Logistic, ReLU]

The optimal model for each classifier was found by recursively assess each combination of hyperparameters on each data set using the grid search technique (Lindholm et al., 2021, p. 110), so the total number of evaluated models ranged from 6 (MLP) and 360 (RF) on each training data set. The optimal hyperparameters of a classifier were chosen based on the model with the highest F_1 score (to increase the performance of predicting the positive class) which was the average validation result using the 10-fold cross validation technique on the training data. The optimal algorithm of each classifier was then applied on the test data.

6. Results

In this chapter, the results are presented. In the first section, the features' MI scores and the selected ones in each data set are considered. Further, the hyperparameters for each classifier and the performance of all these ML algorithms are explored. Additionally, the optimal classifiers applied on each test data set – which achieved the highest F_1 score – are introduced, where their corresponding hyperparameters are presented together with their performances.

6.1 Mutual information scores

In the first data set, $D = 0$, the numerical features with the lowest and highest MI scores (Figure 14) were Interests ($I(Interests; Class) = 0.04$) and Pages ($I(Pages; Class) = 0.25$). The corresponding features in the next data set, $D = 10$, were Interests ($I(Interests; Class) = 0.06$) together with Location search ($I(Location\ search; Class) = 0.06$), and Pages ($I(Pages; Class) = 0.22$). Lastly, the analogous features in the data set $D = 20$ were Location search ($I(Location\ search; Class) = 0.06$) and Pages ($I(Pages; Class) = 0.21$).

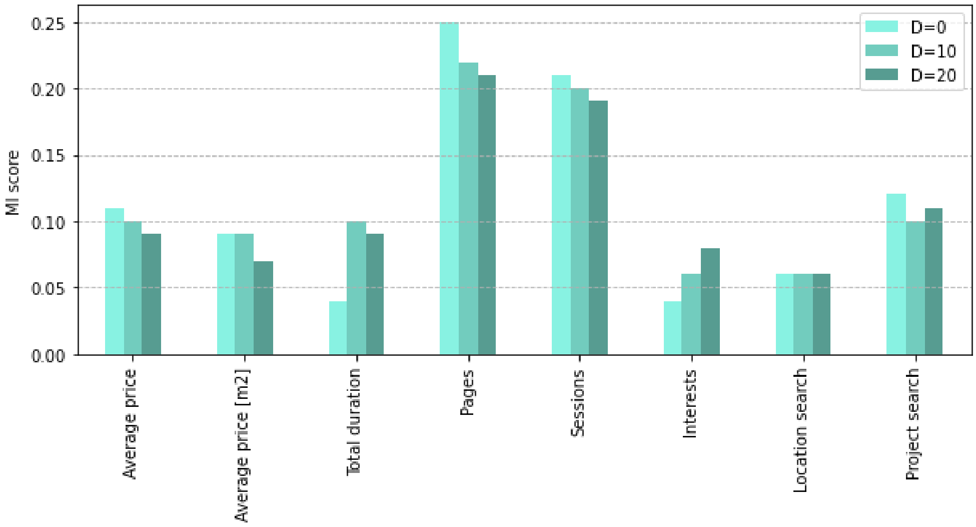

Figure 14. The MI score for each numerical feature in the data sets D = [0, 10, 20].

Regarding the categorical input variables, the lowest and highest MI scores (Figure 15) in the first data set, $D = 0$, were SMS ($I(SMS; Class) = 0$) together with Building type 2 ($I(Building\ type\ 2; Class) = 0$) and General newsletter ($I(General\ newsletter; Class) = 0$), and MF location search ($I(MF\ location\ search; Class) = 0.014$), respectively. In the next data set, $D = 10$, the analogous features were MF project search ($I(MF\ project\ search; Class) = 0$) and Building type 0 ($I(Building\ type\ 0; Class) = 0.028$) together with Building category 0 ($I(Building\ category\ 0; Class) = 0.028$). Finally, the lowest and highest values in the last data set, $D = 20$, were MF project search

($I(MF\ project\ search; Class) = 0$) and Building type 0 ($I(Building\ type\ 0; Class) = 0.019$) together with Building category 0 ($I(Building\ category\ 0; Class) = 0.019$), respectively.

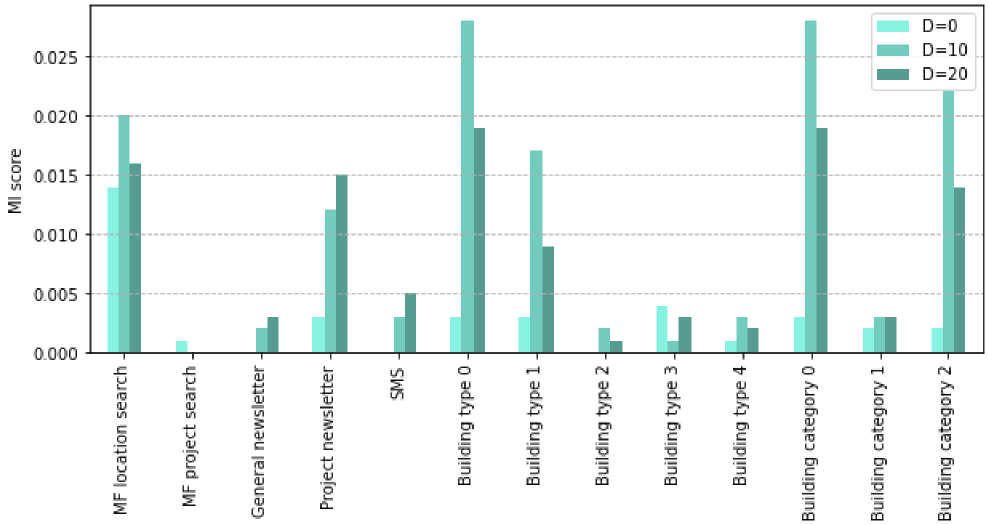

Figure 15. The MI score for each categorical feature in the data sets D = [0, 10, 20].

The MI score threshold was set to 0.05 and the features above, or equal to that value were included in the data sets (Table 11). Pages, Sessions, Project search, Average price, Average price [m^2] and Location search constituted all the input variables for data set $D = 0$; these features were also included in the final two data sets ($D = 10$ and $D = 20$). In addition, Total duration and Interests were included in these two data sets. All the categorical input variables were discarded.

Table 11. The considered features in each data set with their respective MI score.

$D = 0$		$D = 10$		$D = 20$	
Feature	MI	Feature	MI	Feature	MI
Pages	0.25	Pages	0.22	Pages	0.21
Sessions	0.21	Sessions	0.20	Sessions	0.19
Project search	0.12	Total duration	0.10	Project search	0.11
Average price	0.11	Average price	0.10	Total duration	0.09
Average price [m^2]	0.09	Project search	0.10	Average price	0.09
Location search	0.06	Average price [m^2]	0.09	Average price [m^2]	0.08
		Location search	0.06	Interests	0.08
		Interests	0.06	Location search	0.06

6.2 The optimal hyperparameters and performance of the classifiers

The first considered algorithm, LR, had the same hyperparameter values for all the two optimal models applied on data set $D = 10$ and $D = 20$. These utilized L_1 regularization with a regularization parameter of $\rho = 0.01$, whereas the LR applied on $D = 0$ utilized L_1 regularization with a regularization parameter of $\rho = 10$. The hyperparameters regarding the next model, k-NN, were the same on all the data sets $D = [0, 10, 20]$: the number of neighbors were $k = 2$, and weighted distances were utilized. Next, the three optimal DTs only shared the maximum depth, 10 nodes in a branch, on all the data sets $D = [0, 10, 20]$. The DTs applied on the second and third data set ($D = 10$ and $D = 20$) also shared the same minimum number of records in a leaf node (20 records), whereas the DT applied on the first data set ($D = 0$) utilized 10 samples in a leaf node. The number of minimum samples in a split were 5, 15 and 25 for the DTs applied on the data sets $D = 0$, $D = 10$ and $D = 20$, respectively. Moreover, the optimal RFs all shared the maximum depth of a tree (10 nodes in a branch) and minimum samples in a split (5 records). The RFs applied on the first ($D = 0$) and the last data set ($D = 20$) utilized the same number of minimum samples in a leaf node (5 records), whereas the optimal RF applied on $D = 10$ used 10 records instead. Also, the RFs applied on $D = 10$ and $D = 20$ both utilized 200 ensemble members; 100 more trees in comparison with the RT on the first data set $D = 0$. The last model, MLP, used the same activation function, ReLU, on all data sets. The MLPs applied on $D = 0$ and $D = 20$, consisted each of 30 neurons, while the MLP applied on the second data set, $D = 10$, utilized 10 neurons.

In the first data set, $D = 0$, the ACC for all classifiers ranged between 0.693 (MLP) and 0.794 (LR) (Table 12). The minimum and maximum value of the TPR were 0.341 (k-NN) and 0.635 (MLP), respectively, while the corresponding PPV scores were 0.319 (k-NN) and 0.391 (LR). The minimum F_1 score was 0.330 (k-NN) and the maximum value was 0.414 (MLP). Furthermore, the performance of predicting the negative class was, in general, higher in comparison with the positive class. The range of the TNR scores were 0.705 (MLP) and 0.880 (LR), while the lowest and highest NPV values were 0.862 (k-NN) and 0.904 (MLP), respectively.

Table 12. The performance of the classifiers on each data set D =[0,10,20], where the highest F_1 score achieved on each test data set is marked.

Day	Model	ACC	F_1	TPR	PPV	TNR	NPV
	LR	0.794	0.382	0.373	0.391	0.880	0.872
	k-NN	0.763	0.330	0.341	0.319	0.850	0.862
0	DT	0.746	0.374	0.443	0.323	0.809	0.876
	RF	0.752	0.397	0.478	0.339	0.808	0.883
	MLP	0.693	0.414	0.635	0.307	0.705	0.904
	LR	0.728	0.327	0.503	0.243	0.763	0.910
	k-NN	0.740	0.309	0.442	0.237	0.785	0.903
10	DT	0.770	0.356	0.484	0.281	0.813	0.912
	RF	0.798	0.413	0.541	0.334	0.837	0.923
	MLP	0.748	0.382	0.591	0.282	0.772	0.926
	LR	0.709	0.294	0.500	0.208	0.737	0.914
	k-NN	0.750	0.285	0.412	0.218	0.797	0.908
20	DT	0.786	0.323	0.422	0.262	0.836	0.913
	RF	0.801	0.369	0.480	0.300	0.845	0.922
	MLP	0.763	0.346	0.516	0.260	0.797	0.923

In the second data set, $D = 10$, the ACC range was 0.728 (LR) and 0.798 (RF). The performance of predicting the positive class was low for each classifier here as well; the minimum and maximum values of TPR, PPV, and F_1 scores were: 0.442 (k-NN) and 0.591 (MLP), 0.237 (k-NN) and 0.334 (RF), 0.309 (k-NN) and 0.413 (RF), respectively. The classifiers applied on this data set had, however, in general high performance of predicting the negative class; the TNR scores ranged from 0.772 (MLP) and 0.837 (RF), whereas the minimum and maximum NPV scores were 0.903 (k-NN) and 0.926 (MLP).

Lastly, in the data set, $D = 20$, the ACC ranged from 0.709 (LR) and 0.801 (RF). The minimum and maximum TPR score were 0.412 (k-NN) and 0.516 (MLP), and the corresponding values for the PPV score were 0.218 (k-NN) and 0.300 (RF), respectively. The F_1 score ranged from 0.285 (k-NN) and 0.369 (RF). On the contrary, the classifiers for this data set had a higher performance in predicting the negative class as well. The minimum and maximum TNR were 0.737 (LR) and 0.845 (RF), respectively, and the corresponding values for the NPV scores were 0.908 (k-NN) and 0.923 (MLP).

Regarding the AUC-ROC scores in $D = 0$, the ML algorithms LR, k-NN, DT, RF, and MLP received AUC-ROC scores of 0.638, 0.630, 0.656, 0.694 and 0.715, respectively (Figure 16). The PR scores for the classifiers were lower in comparison with their respective AUC-ROC values: 0.307, 0.358, 0.297, 0.376 and 0.350.

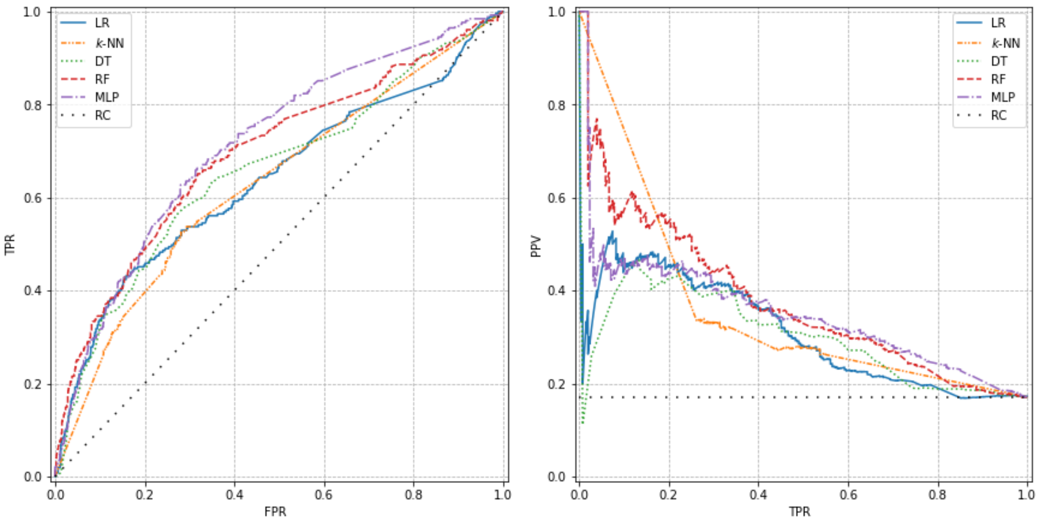

Figure 16. The ROC curve (left) and PR curve (right) for the classifiers using the test data set D = 0.

In the second the data set, $D = 10$, the AUC-ROC scores were 0.699, 0.652, 0.683, 0.734, and 0.738 for the LR, k-NN, DT, RF and MLP, respectively, with the corresponding AUC-PR scores 0.281, 0.348, 0.302, 0.373 and 0.345 (Figure 17).

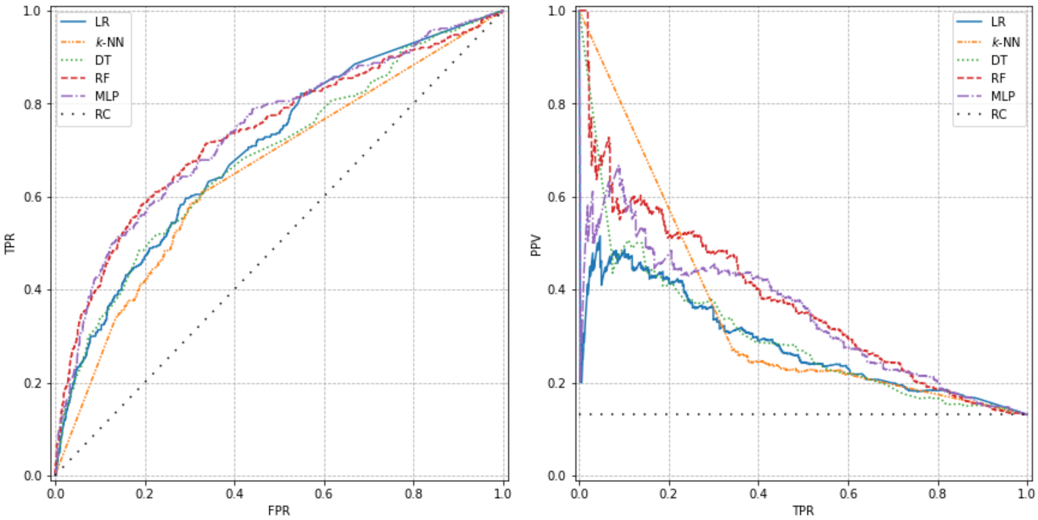

Figure 17. The ROC curve (left) and PR curve (right) for the classifiers using the test data set D = 10.

Finally, the AUC-ROC scores regarding LR, k-NN, DT, RF and MLP in the data set $D = 20$ were 0.694, 0.636, 0.679, 0.722 and 0.708, respectively. In addition, the AUC-PR scores for the corresponding models were 0.298, 0.308, 0.284, 0.330 and 0.320 (Figure 18).

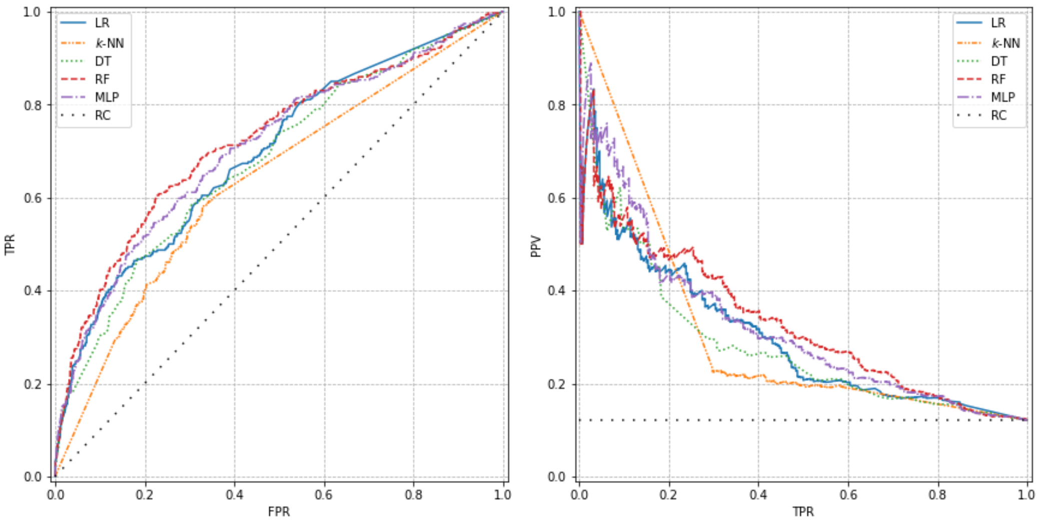

Figure 18. The ROC curve (left) and PR curve (right) for the classifiers using the test data set D = 20.

6.3 The performance of the optimal classifiers applied on each data set

By using F_1 score as measure of the performance, a MLP was the optimal model on the first data set ($D = 0$), whereas a RF achieved the best results on the second ($D = 10$) and last data set ($D = 20$). The hyperparameters of the MLP consisted of a hidden layer size of 30 neurons with ReLU as activation functions. The RF utilized the same hyperparameters on $D = 10$ and $D = 20$ except for the minimum samples in a leaf node. The RFs used the maximum depth of 10 nodes in a branch, 5 records in the minimum split and 200 estimators in total. The minimum samples in the leaf nodes were 10 and 5 for the RT applied on the data sets $D = 10$ and $D = 20$, respectively.

The positive class performance, using the F_1 measure, of the MLP was 0.414 on the test data set. Meanwhile, the performance of predicting the negative class was higher: the TNR and NPV were 0.705 and 0.904, respectively. Since 255 buyers and 1238 non-buyers were considered in this test data set, the number of correctly classified records were 162 (TPs) and 873 (TNs). The number of FNs and FPs were 93 and 365, respectively (Table 13).

Table 13. Confusion matrix - MLP, D = 0.

	$\hat{y} = 1$	$\hat{y} = -1$
$y = 1$	162	93
$y = -1$	365	873

In the second data set, $D = 10$ (where 364 buyers and 2405 non-buyers are included), the RF was the optimal algorithm with a F_1 score 0.413, whereas the TNR and NPV were

0.837 and 0.923, respectively. The number of correctly classified records were 197 (TPs) and 2012 (TNs), whereas the number of incorrectly classified records were 167 (FNs) and 393 (FPs) (Table 14).

Table 14. Confusion matrix - RF, D = 10.

	$\hat{y} = 1$	$\hat{y} = -1$
$y = 1$	197	167
$y = -1$	393	2012

The last model, RF, achieved a F_1 score of 0.369 on the data set $D = 20$ with TNR and NPV scores of 0.845 and 0.922, respectively. In this data set, 306 buyers and 2217 non-buyers were considered. The number of TPs and TNs were 147 and 1874 records, respectively (Table 15). The number of misclassified were 159 (FNs) and 343 (FPs).

Table 15. Confusion matrix - RF, D = 20.

	$\hat{y} = 1$	$\hat{y} = -1$
$y = 1$	147	159
$y = -1$	343	1874

Although the number of customers differ between the data sets, the optimal MLP achieved the highest F_1 score. The RF applied on the second data set, $D = 10$, achieved the highest AUC-ROC and AUC-PR scores (0.734 and 0.373 respectively) in comparison with the other two optimal algorithms. The RF, on the second data set, did also achieve a higher AUC-ROC score (0.722) in comparison with the MLP (0.715). Meanwhile, the latter achieved a higher AUC-PR score (0.350) relative to the former (0.330).

7. Discussion

In this chapter, the results will be analyzed. First, the input variables are discussed followed by a section regarding the selection of applying the considered classifiers in a SP. Finally, the developed ML framework is evaluated regarding the GDPR.

7.1 The utilized input variables

A filtering approach was implemented to identify the optimal features in each considered data set $D = [0, 10, 20]$. The optimal features differed slightly between the data sets (Table 11): the classifiers utilized Pages, Sessions, Project search, Average price, Average price $[m^2]$ and Location search in the first data set $D = 0$. The same features were also included in the classifiers applied on data set $D = 10$ and $D = 20$. Also, Total duration and Interests were utilized in the second and third data set.

Moreover, as mentioned in Section 4.1, the SP is a social process, which means that the involved individuals also affect the actual outcome of a SP. Although the customer does not have the authority to affect the company's decisions, that individual can still *decide* to change an opinion: the project might not be as interesting as it was before at day $D = 0$. The choice of starting a SP and the purchase intention of a customer cannot be seen as static. This social process might be one reason why it is more challenging to predict the positive class compared to the negative class on the test data with the utilized hyperparameters in each classifier. The web data utilized in this thesis might not cover the complex behavior of buyers. However, it is argued that the web data can cover the purchase intentions of the negative class. A common denominator between non-buyers is the lack of interest in browsing on the company's web site regarding projects, which constituted approximately 70% of those customers (Figure 10). This could plausibly be a reason why the negative class was easier to predict. Thus, it is suggested that further features are needed to cover the purchase intentions for the buyers, whereas the utilized features in this study are useful for classifying the negative class.

7.2 The optimal time point for applying classification during sales processes

The MLP applied on the data set $D = 0$ has the highest F_1 score (0.414) and it can be seen as the best classifier for predicting the positive class. However, the total number of customers in the data set $D = 0$ was lower compared with the other data sets: the first data set, $D = 10$, and second data set, $D = 20$, contained approximately 85% and 69% more records, respectively. It is important to note that 13369 customers were considered, and the total number of customers were fewer in both the training and test data sets

$D = [0, 10, 20]$ (Table 4 and 5). Because of this, the number of customers could have affected the outcome, since, e.g., the identification of the optimal hyperparameter values could have been affected. To choose the optimal model among all classifiers applied on the test data sets to identify when it is suitable to apply classification during SPes, it is argued to only evaluate the classifiers on the second and third data set since these have been trained and evaluated on substantially more records.

Furthermore, it can be seen in the results (Table 12) that all the classifiers – with their set of hyperparameter values – had difficulties with predicting the positive class, meanwhile the classification of the negative class was better. It can be argued that this might be due to the low value prevalence in each data set, and this could, therefore, be another reason that it was easier to predict the negative class in general. Although, the classifiers were still able to predict TPs (Table 12, 13, 14 and 15).

Instead of using the F_1 score, it is argued that the NPV and TPR scores are essential for comparing the performance of the considered classifiers. When both NPV and TPR are maximized, the prediction of the negative class is maximized, and FNs are minimized. Based on this, it can therefore be stated the best classifier is the MLP on the data set $D = 10$: the TPR and NPV scores were 0.591 and 0.926, respectively (Table 12). This model has the highest performance, using these metrics, in comparison with all other models, and the number of customers in this data set was the highest compared to the other data sets. The number of FPs and TNs were greater than the corresponding records in the optimal RF model, (167 and 2012, respectively) on the test data set $D = 10$ (Table 14), which was based on the F_1 score. With respect to other metrics, the MLP on $D = 10$ achieved the highest AUC-ROC score among all the classifiers (0.738). The AUC-PR score (0.345) was however lower in comparison with other models (such as the RF on the same day). The PPV score, 0.282, (Table 12) and the ACC, 0.748, were also lower compared with other classifiers. Nonetheless, these scores do arguably not affect the final evaluation of the model. All classifiers have a challenge to predict the positive class, and the metrics for evaluating the performance of predicting the positive class, such as PPV or AUC-PR, can therefore not be seen as crucial metrics in this context. With regard to ACC, this does only describe the general performance of a model (and not the performance of predicting specific classes).

The optimal model, a MLP with 10 neurons in its hidden layer with ReLU as activation functions, was found by using the second data set: $D = 10$. This means that it is suggested to apply classification algorithms 10 days after a SP starts for a customer to predict the class. By using web data as the source for classifying the customer groups, it is not suggested to apply classification models on the first day $D = 0$ (due to the lack of customers). Also, it can be stated that predicting classes of the customers on the third data set, $D = 20$ is also not recommended.

The results of this ML framework shows that the purchase intention of a customer can be classified where the performance of predicting non-buyers is higher in comparison with buyers. If a MLP with the same hyperparameters (based on the same ML framework) was implemented in production 10 days after the start of SPes to classify purchase intentions of customers, it is suggested to utilize it as a method for identifying the non-buyers. Using a ML framework with a similar methodology in production would thus be a useful tool to gain more control of the TOM of units.

7.3 The machine learning framework with respect to GDPR

In the ML framework, customers have been classified to predefined categories, buyers and non-buyers, based on their behaviors on the company's web site. The data does not contain any sensitive information regarding the customers, their web activity has solely been used, and the customers have been pseudonymized. Moreover, this ML framework has been produced as an ad hoc analysis to understand the possibilities of predicting the customer groups by using web data. The ML framework has not been conducted to generate new profiles of the customers in the company's database.

It is argued that this ML framework has been developed accordingly with respect to the GDPR. The ML framework follows Recital 72 in the GDPR, since no sensitive data has been utilized in the classifiers. As encouraged in Article 4(5) in the GDPR, the customers have been pseudonymized to protect the privacy of the concerned customers in this thesis. In addition, it is argued that this ML framework cannot be viewed as profiling; it is rather an exploratory work which has evaluated the usage of web data as a basis for predicting customer groups in a test environment.

The methodology of this ML framework could, however, be utilized for profiling purposes. If the company wants to develop this further, a DPIA would be useful to conduct profiling fairly. For example, a plausible indirect bias might have occurred in the data sets used in the ML framework, which could have affected the results. Customers from Germany consisted of approximately 76% (Table 3) of all the considered customers, whereas the rest belonged to Estonia, Finland, Latvia, Lithuania, Norway and Sweden. Also, it is necessary to evaluate whether this in fact is an indirect bias as well as to identify other potential indirect biases. Regarding transparency, the company should provide meaningful information to the customers about the profiling process. According to Gutwirth et al. (2017), this should not be too technical, and, as interpreted by Powles and Selbst (2017), the actual algorithm(s) does not need to be described (to keep intellectual property rights). To implement profiling lawfully, human involvement in decision-making must occur. It is therefore argued that the results of a ML framework with this methodology should not be the only basis for decisions taken by the company in the SPes. The results should be evaluated and additional aspects should be considered as well. Furthermore, it is suggested that a ML framework that uses the same methodology should, at most, be used as a complementary tool in decision-making.

8. Conclusions

In this thesis, a ML framework has been developed to classify the customer groups, buyers and non-buyers, of customers, by using their web activity as input variables. The classifiers have been applied during different points in time of a SP to identify when it is suitable for predicting the customer groups. Three data sets, $D = [0, 10, 20]$ were constructed, which included the number of customers that were still active in a SP. In this ML framework, five classifiers – LR, k-NN, DT, RF and MLP – have been evaluated.

With regard to the first research question, *Which are the most relevant input variables in each data set?*, MI was used for filtering the input variables and to identify the most relevant ones. The total number of visited pages (Pages) and sessions (Sessions) were included in all three data sets, including the average price of a project in total (Average price) and per square meter (Average price $[m^2]$). Additionally, the total number of searched projects on the company's web site (Project search) and locations (Location search) were also included. The total duration spent on the company's web site (Total duration) and the total number of registration of interests sent (Interests) were only considered in the second and third data set ($D = 10$ and $D = 20$).

Regarding the second research question, *When is it recommended to classify the purchase intentions of customers during SPes?*, it was argued that using metrics for measuring the performance of predicting the positive class were not feasible as a basis for choosing the optimal model. Instead, the NPV and TPR scores of the classifiers were considered. By using these metrics, a MLP – with 10 neurons and ReLU as activation function – was suggested as the optimal classifier on the second data set ($D = 10$). It was therefore suggested to use this classifier to predict the purchase intentions of a customer 10 days after a SP starts.

Lastly, in respect to the third research question *Are there any challenges for this ML framework with respect to the GDPR?*, it was argued that the developed ML framework was in compliance with the GDPR. Further, if this methodology would be used in production for profiling purposes, it was suggested to conduct DPIA for assessing the potential indirect biases. Also, to be in compliance with the GDPR, this methodology was recommended to be used as, at most, a complementary tool in decision-making processes given that the customer is provided with meaningful information regarding the profiling process.

8.1 Future work

In the field of predicting purchase intentions of customers in real estate SPes by using ML, some ideas might be considered for future studies. To begin with, other definitions of the customer groups can be utilized to include more records, for example including customers that have purchased more than one unit can be one approach. Also, considering all the SPes connected to a certain customer would be another solution. However, this would most certainly violate the uniqueness of the customers in a data set and should be taken into consideration. Furthermore, since the company is located in several European countries, customer behaviors could plausibly be different between the countries. For cross-national companies, to identify whether this factor is an indirect bias or not is a critical step for understanding the purchase intentions of customers. Next, additional input variables which represent other activities on the web site can be useful to investigate the purchase intentions of customers. With regards to feature selection, other methods could be considered. Wrapper methods could be utilized instead of filter techniques to evaluate the importance of the input variables. Moreover, to tackle the class imbalance problem, other approaches might be useful. Instead of using data resampling on training data sets, utilizing algorithm-level techniques, e.g. modifying the decision thresholds of a classifier, could be implemented in the hyperparameter optimization step. This to avoid learning models on modified training data sets. Also, using $\beta > 1$ in F_β could be used for identifying the optimal hyperparameters, since the TPR was suggested to be one of the most important metrics to evaluate the performance of a classifier. Finally, dependency modeling (by using, e.g., markov models, bayesian belief networks or recurrent neural networks) could also be applied instead of classifying the purchase intentions by using the customer's navigational pattern on a company's web site.

References

Abbasi, H., Berg, S., Brett, M., Del Río, J.F., Cournapeau, D., Gérard-Marchant, P., Gohlke, C., Gommers, R., Haldane, A., Harris, C.R., Hoyer, S., Kerkwijk, M.H., Kern, R., Millman, K.J., Oliphant, T.E., Peterson, P., Picus, M., Reddy, T., Sheppard, K., Smith, N.J., Taylor, J., Virtanen, P., Walt, S.J, Weckesser, W., Wiebe, M., and Wieser, E. (2020), "Array programming with NumPy", Nature vol. 585, no. 7825, pp. 357–362.

Aridas, C.K., Lemaître, G., and Nogueira, F. (2017), "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning", Journal of Machine Learning Research vol. 18, no. 17, pp. 1–5, URL: `http://jmlr.org/papers/v18/16-365.html`.

Banaitis, A., Ferreira, F., Ferreira, J., Spahr, R., and Sunderman, M. (2016), "A learning-oriented decision-making process for real estate brokerage service evaluation", Service business vol. 11, no. 3, pp. 453–474.

Bishop, C.M. (2006), Pattern recognition and machine learning, New York: Springer.

Blondel, M., Brucher, M., Cournapeau, D., Dubourg, V., Duchesnay, É., Gramfort, A., Grisel, O., Michel, V., Passos, A., Pedregosa, F., Perrot, M., Prettenhofer, P., Thirion, B., Vanderplas, J., Varoquaux, G., and Weiss, R. (2011), "Scikit-learn: Machine Learning in Python", Journal of machine learning research.

Bonava (n.d.), Information om personuppgiftsbehandling, URL: `https://www.bonava.se/om-oss/information-om-personuppgiftsbehandling` (visited on 11/20/2021).

Bowyer, K.W., Chawla, N.V., Hall, L.O., and Kegelmeyer, W.P. (2002), "SMOTE: Synthetic Minority Over-sampling Technique", The Journal of artificial intelligence research vol. 16, pp. 321–357.

Cate, F.H., Kuner, C., Lynskey, O., Millard, C., and Svantesson, D. (2017), "Machine learning with personal data: is data protection law smart enough to meet the challenge?", International data privacy law vol. 7, no. 1, pp. 1–2.

Chen, K.K., Chou, P.H., Li, P.H., and Wu, M.J. (2010), "Integrating web mining and neural network for personalized e-commerce automatic service", Expert systems with applications vol. 37, no. 4, pp. 2898–2910.

Cheng, K., Li, J., Liu, H., Morstatter, F., Tang, J., Trevino, R., and Wang, S. (2018), "Feature Selection: A Data Perspective", ACM computing surveys vol. 50, no. 6, pp. 1–45.

Cheng, P., Lin, Z., and Liu, Y. (2008), "A Model of Time-on-Market and Real Estate Price Under Sequential Search with Recall", Real estate economics vol. 36, no. 4, pp. 813–843.

Cheng, P., Lin, Z., and Liu, Y. (2010), "Illiquidity, transaction cost, and optimal holding period for real estate: Theory and application", Journal of housing economics vol. 19, no. 2, pp. 109–118.

Cooley, R., Deshpande, M., Srivastava, J., and Tan, P.N. (2000), "Web usage mining: discovery and applications of usage patterns from Web data", SIGKDD explorations vol. 1, no. 2, pp. 12–23.

Cover, T.M. and Thomas, J.A. (1991), Elements of information theory, New York: Wiley.

Davis, J. and Goadrich, M. (2006), "The Relationship between Precision-Recall and ROC Curves", in: Proceedings of the 23rd International Conference on Machine Learning, Association for Computing Machinery, pp. 233–240.

Ding, C., Long, F., and Peng, H. (2005), "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy", IEEE transactions on pattern analysis and machine intelligence vol. 27, no. 8, pp. 1226–1238.

Dombrow, J. and Turnbull, G.K. (2007), "Individual Agents, Firms, and the Real Estate Brokerage Process", The journal of real estate finance and economics vol. 35, no. 1, pp. 57–76.

Drake, F.L. and Van Rossum, G. (2009), Python 3 Reference Manual, Scotts Valley: CreateSpace.

Esposito, C., Landrum, G.A., Schneider, N., Stiefl, N., and Riniker, S. (2021), "GHOST: Adjusting the Decision Threshold to Handle Imbalanced Data in Machine Learning", Journal of Chemical Information and Modeling vol. 61, no. 6, pp. 2623–2640.

European Union (2016), General Data Protection Regulation, URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

Fernández, A., García, S., Herrera, F., López, V., and Palade, V. (2013), "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics", Information Sciences vol. 250, pp. 113–141.

Ferreira, F. and Jalali, M. (2015), "Identifying key determinants of housing sales and time-on-the-market (TOM) using fuzzy cognitive mapping", International journal of strategic property management vol. 19, no. 3, pp. 235–244.

Friedman, J.H., Hastie, T., and Tibshirani, R. (2009), The elements of statistical learning: data mining, inference, and prediction, 2nd Edition, New York: Springer.

Griffith, B. (2013), Marketing automation with Eloqua, Birmingham: Packt Publishing.

Gutwirth, S., Hert, P.D., Leenes, R., and van Brakel, R. (2017), Data protection and privacy: the age of intelligent machines, vol. 10, Oregon, Oxford and Portland: Hart Publishing.

Han, J. and Kamber, M. (2006), Data mining: concepts and techniques, 2nd Edition, Burlington: Elsevier.

Hand, D.J., Mannila, H., and Smyth, P. (2001), Principles of data mining, 1st Edition, Cambridge: MIT Press.

Hunter, J.D. (2007), "Matplotlib: A 2D graphics environment", Computing in Science & Engineering vol. 9, no. 3, pp. 90–95.

Kastro, Y., Katircioglu, M., Polat, S.O., and Sakar, C.O. (2019), “Real-time prediction of online shoppers’ purchasing intention using multilayer perceptron and LSTM recurrent neural networks”, Neural computing & applications vol. 31, no. 10, pp. 6893–6908.

Kumar, V. and Reinartz, W. (2018), Customer Relationship Management: Concept, Strategy, and Tools, 3rd Edition, New York: Springer.

Kumar, V., Steinbach, M., and Tan, P.N. (2014), Introduction to data mining, 1st Edition, Harlow: Pearson.

Lindholm, A., Lindsten, F., Schön, T.B., and Wahlström, N. (2021), Machine Learning - A First Course for Engineers and Scientists, URL: `https://smlbook.org`.

Liu, B. (2011), Web data mining: exploring hyperlinks, contents, and usage data, 2nd Edition, Berlin and Heidelberg: Springer.

Powles, J. and Selbst, A.D. (2017), “Meaningful information and the right to explanation”, International data privacy law vol. 7, no. 4, pp. 233–242.

Ramesh, V. and Thushara, Y. (2016), “A Study of Web Mining Application on E-Commerce using Google Analytics Tool”, International journal of computer applications vol. 149, no. 11, pp. 21–26.

Scott, K. (2017), “U-SQL”, in: IoT Solutions in Microsoft’s Azure IoT Suite, Berkeley: Apress, pp. 173–190.

The pandas development team (2021), pandas-dev/pandas: Pandas, version 1.3.5, URL: `https://doi.org/10.5281/zenodo.3509134`.

Appendix A

Z-score normalization

Z-score normalization is a data transformation method to convert the range of values of an input variable (Han and Kamber, 2006, p. 72). It is defined as

$$z = \frac{x_{i,j} - \mu_{x_i}}{\sigma_{x_i}} \tag{35}$$

where $x_{i,j}$, μ_{x_i}, and σ_{x_i} represent the value of a record $(j = 1, 2, ..., n)$, the mean value of the input variables x_i $(i = 1, 2, ..., m)$, and the standard deviation of the input variable, respectively (ibid., p. 72).

One-hot encoding

When a categorical variable contains more than two values, it can be represented as a k-dimensional binary feature vector

$$\mathbf{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,k}] \tag{36}$$

where k is the number of values the input variable $\mathbf{x}_i$ contains (Lindholm et al., 2021, p. 43). This method is known as one-hot encoding; when one record contains a value of $\mathbf{x}_i$, the feature will be represented as 1 while the other features will be 0.

The one-hot encoded and binarized categorical features are presented in Table 16.

Table 16. The one-hot encoded and binarized categorical features.

Feature	Categorical value	Vector/value
Building type 0	NA	[1, 0, 0, 0, 0]
Building type 1	Block of flats	[0, 1, 0, 0, 0]
Building type 2	Row house	[0, 0, 1, 0, 0]
Building type 3	Semi-detached house	[0, 0, 0, 1, 0]
Building type 4	Single family house	[0, 0, 0, 0, 1]
Building category 0	NA	[1, 0, 0]
Building category 1	Multi family	[0, 1, 0]
Building category 2	Single family	[0, 0, 1]
MF location search	True	1
	False	0
MF project search	True	1
	False	0
Project newsletter	True	1
	False	0
General newsletter	True	1
	False	0
SMS	True	1
	False	0