THESIS

INTERPRETATION OF CONTENT FROM PAYMENT DOCUMENTS USING MACHINE LEARNING

Mathematical Engineering

14 March 2022

Roger Ruiz Salvat

615345

Table of contents

1. Management Summary

A functionality which scans payment documents in order to extract information from them, presents efficacy and scalability issues. In this thesis, an alternative solution based on machine learning is exposed. The project is approached as a binary classification problem in which numeric figures from payment documents are positively classified if they represent the total price of the invoice.

Following the steps from a data mining process, a dataset is generated from the content of the documents uploaded by users. The documents are obtained from a cloud storage and their content is extracted using an OCR algorithm. Then each numeric figure is converted into a dataset instance. The factors describe the situation of the figure in the document relating it with a specific key word. The key word has a semantic meaning that indicates the total price of the payment.

Data cleaning techniques have been applied to facilitate the training process of the machine learning algorithms. Undersampling was applied to obtain a balanced dataset. Then, outlier values from predictors were removed to reduce the bias in the dataset. With correlation analysis, it was ensured that no factors were dependent on each other. Lastly, the predictors were normalised to scale them into the same range of values.

The algorithms used in this thesis were the logistic regression, the K nearest neighbours, the decision trees, and the random forest. They were chosen based on theoretical research considering the characteristics of the dataset, their mapping function, and the expected training times.

The algorithm hyperparameters have been optimised using a random search method to save computing efforts, and stratified K-fold cross-validation to avoid optimising on overfitted models while evaluating on balanced datasets. A random forest model with 81 estimators provided the best performance results in recall. Other performance measurements and visualisation methods provided veracity to the evaluation results.

2. Introduction

Small businesses often need to do paperwork and keep track of their financial activity. Without a dedicated department, those tasks might become heavy and complex by non-financial experts who are running a business.

Gekko is an information technology company which provides financial tools to freelancers and small-sized companies. Such tools allow users to manage their hour registration, kilometre tracking, payslips, etc. All these functionalities can be either used from a mobile application or from a website platform. There are two subscription plans which users can be subscripted to. A free basic plan with limited functionalities, and a premium plan with more features available and unlimited functionalities.

Besides the mobile and web applications, Gekko's informatic system is organised into several software components and a database which stores user data. As an information technology company, Gekko keeps in the agenda the maintainability and scalability of the system while providing quality products and improving user experience. That means that some features need to be updated or sometimes even reimplemented from scratch.

This thesis updates one specific feature in order to improve its efficacy and provide a better user experience. It is intended to use machine learning to achieve such an objective and other benefits which might originate, such as solving scalability and maintainability issues.

a. Context

A functionality developed by Gekko involves the transaction from data in a text format into machine encoded data. In other words, when a user is registering a payment document as a cost, it is possible to scan the document and extract some information from it. This information is used to autocomplete some of the fields from the digital form where users register their costs. One of these fields is the total price of the payment.

The described functionality is called SmartScan. It is activated when a user uploads an image or PDF into the application. It has two main steps. First, it sends the uploaded file to an external API which would recognise characters from the payment. The API returns a list of string values which were taken from the image or document, together with the position in the document. Then, an algorithm processes the API response and selects figures to autocomplete the form.

i. Problem statement

Currently, SmartScan detects the total price from a payment document executing a JavaScript method which is run at the end-user device. The method uses regular expressions and geometrical patterns to extract the total price. This makes the functionality inflexible when handling

different document structures or languages. Moreover, the software is difficult to scale and maintain since there is just one method which is responsible for the whole process.

ii. Goals and objectives

The goal of this thesis is to create a predictive model which is able to classify figures extracted from a payment. By using machine learning, it is intended to classify numeric figures in order to predict whether they are the total price or not. It is expected that such a solution would overcome the mentioned problems. In addition, by slightly modifying the process used to generate the model, it should be possible to generate an analogous model which would be applied to determine whether the figure is the VAT or not. However, in this thesis, just the first implementation of the model is presented.

b. Main research questions and subquestions

This project is structured with the following main question and subquestions.

How can a predictive model, which binary classifies figures extracted from a payment document, be generated?

a. What data is available?

b. Which machine learning algorithms can be used to train the model?

c. Which preprocessing steps are required to generate a dataset?

d. Which set of hyperparameters are the optimal for each of the machine learning algorithms?

e. How can the performance of the predictive model be measured?

3. Methodology

By answering the approached subquestions, the main research question of the project is answered. As a machine learning project, the methodologies contain data mining techniques such as the extraction, integration, transformation, and cleaning of data together with modelling, pattern detection, and the evaluation of the results [1].

a. What data is available?

Digital storages from online platforms keep not only user's information, but also their actions. However, most of the user's personal data is fragmented across multiple sources [2].

To answer this subquestion, the organisation's data storage must be studied. Because of the diversity of information that Gekko is storing, the scope of research must be narrowed. In particular, data containing the information from user registered payments is researched.

During the research process, it is also analysed which elements from the available data can be used to form the dataset. Additionally, the sources from where the data can be extracted are obtained and later used at the data extraction step.

Data extraction is a preliminary process in data mining which unifies the fragmented data extracted from different sources for better indexing and querying [3].

b. Which machine learning algorithms can be used to train the model?

There are plenty of machine learning algorithms that are able to generate predictive models. In this project, a selection of those is used to be part of the mining process. The selected algorithms must fit into the requirements of the project. They are implemented using the machine learning library Scikit Learn [4]. Such a library also includes evaluation and parameter grid search methods.

It is known beforehand that just binary classifiers which can handle continuous numeric variables are going to be used. However, there are other considerations which are important when selecting the machine learning algorithms.

i. The size of the training data

Depending on the size of the dataset, some algorithms might perform better than others. For example, neural networks work well with large datasets and are able to detect hidden patterns and tends [5]. By the size of the dataset is meant, not just the number of instances, but also the number of features. Some algorithms like the support vector machines (SVM) work well on high dimensional datasets [6].

From the results obtained in the subquestion A, an estimation of the size of the dataset is obtained. That aspect may influence the selection of the algorithms.

ii. Restrictiveness of the mapping function

The mapping function refers to the transformation of the data inputs into the output. Some algorithms are less restrictive than others because they can adapt to a wider range of shapes. The K Nearest Neighbours is an example of a flexible algorithm which works well with non-parametric data [7]. On the other hand, some algorithms gain interpretability at the cost of having a more restrictiveness in the mapping function. For example, linear methods like the

linear SVM or the logistic regression would give less accurate predictions if the shape of the data points against the class attribute is non-linear because they can just generate linear functions [8].

iii. Training time

Depending on the complexity of the mapping function, some algorithms take longer to train models than others. Moreover, the size of the dataset might also influence the training time of the algorithm [9]. Depending on how the model is integrated in the system, some training times could be affordable and some others not.

c. Which preprocessing steps are required to generate a dataset?

To train a supervised machine learning model, it is necessary to have a dataset containing rows, columns of predictors, and a class attribute. This subquestion is answered by the data extraction, transformation, and cleaning methods used to obtain a dataset appropriate for training a supervised model.

From answering the subquestion A, it is obtained the sources from where the available data can be extracted. However, due to the big amount of information that the organisation stores, and computing limitations, a selection of data must be made. Such a selection must be narrowed to the scope of interest (payments registered by users), and must be representative of the whole set. Hypothesis testing is used as a technique to prove the representativeness of the selected data.

Once the data extraction process is completed, data transformation steps follow in order to obtain the dataset. The data is converted from the original format (payment documents as images or PDFs), into the dataset format with a class attribute. This conversion is done programmatically using scripts which use dedicated data mining libraries such as Pandas [10].

To deal with inconsistencies, noisy data or oversampling, data cleaning techniques must be applied to achieve better mining results [11]. The data cleaning methods applicable in this project are described below.

i. Undersampling

In binary classification problems, classifiers tend to work well with balanced datasets. Nevertheless, when one of the classes is underrepresented, machine learning models find it difficult to make accurate predictions on a rare event. Furthermore, evaluating a model with an imbalanced dataset might not be appropriate due to the bias towards the majority class. To overcome this problem, undersampling randomly resamples the dataset reducing the number of instances of the majority class [12].

ii. Outlier removal

Datasets often contain some values which are significantly different from the majority. Those values are called outliers. Outliers might be caused by measuring errors or they could have been measured under exceptional circumstances. Therefore, they might not fit well with the model which is being trained. By removing them, machine learning algorithms are able to provide less biased predictive models [13].

A common technique to detect outliers from a dataset is by setting a threshold based on a Z-score value. The Z-score is a statistical centrality measure, and it represents the number of standard deviations above or below the mean that a value is [14]. It is calculated with the formula 3.3.1.

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

Formula 3.3.1 Z score

In this project the rows containing outlier values, for at least one predictor, are removed.

iii. Correlation analysis

There is the possibility of having two or more variables that are correlated amongst each other. In other words, it could be that a predictor describes another predictor or a set of other predictors. This might cause problems when training a model due to the amount of shared information. Moreover, any small variance in a predictor value may have a huge impact on the training process and make the model unstable. This phenomenon is called collinearity [15].

A way to detect collinearity is by calculating correlations between variables. If two variables show a significant correlation, one should be removed from the dataset. There are many different correlations which can be used for correlation analysis. The correlation used in this thesis is chosen depending on the distribution of the variables in the dataset.

Statistical programming libraries like Scipy [16], are used to execute the correlation analysis calculations.

iv. Normalisation

Most likely, a dataset would contain columns whose values are scaled differently. Such a dataset could introduce bias when fitting the model. To avoid that, normalisation scales data into a certain range of values equally for all the predictors in the dataset. A common normalisation method is the min-max normalisation, which rescales values based on the minimum and maximum values found in the column of the dataset [17]. It is calculated using the formula 3.3.2.

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

Formula 3.3.2 Min-Max normalisation

d. Which set of hyperparameters are the optimal for each of the machine learning algorithms?

A machine learning algorithm is built using a training set and is parameterised with a set of hyperparameters [18]. An example of a set of hyperparameters is, for the K Nearest Neighbours algorithm, the number of neighbours and the method used to calculate the distance between the data points. The process of finding the optimal hyperparameters that minimises the error rate in the predictions is called hyperparameter tuning.

Usually, hyperparameter tuning is accomplished by testing on a predefined grid of parameters. However, when the number of combinations of hyperparameters is high, it might be computationally infeasible to find the most optimal set of parameters. To overcome this problem, randomised search algorithms randomly select a subset of parameter combinations to find a close optimal solution. Furthermore, they have other practical benefits, and they are recommended when the number of hyperparameters to tune is high [19].

In addition to the random search, the method used to evaluate the models for the different combinations of parameters is stratified K-fold cross-validation. The metric used is explained at section 3.e.

i. Stratified K-fold cross-validation

Cross-validation is a data resampling method used to prevent overfitting. An overfitted model would be perfectly adapted to the training dataset but unable to generalise well on yet unseen data [20].

K-fold cross-validation makes partitions of the learning set into k disjoint subsets of approximately the same size. The model is trained with $k - 1$ subsets and evaluated with the last subset called the validation set. This process is repeated until all the subsets have been used once, and only once, as the validation set. The average of the measured performance would serve as the cross-validated performance [21].

Stratified K-fold cross-validation is a variation of K-fold cross-validation which creates sets containing approximately the same percentage of each target class [22]. This ensures training and evaluating with balanced sets and gives reliability to the chosen evaluation metric (recall).

e. How can the performance of the predictive model be measured?

To answer this subquestion several aspects must be covered. First by choosing the evaluation metric which is more convenient to fulfil the purpose of the predictors. Then by defining the evaluation strategy in terms of what set of data can be used for testing. Lastly, by applying any kind of visualisation tool which might help to complement the results.

i. Evaluation metric

To evaluate the performance of the machine learning model, an evaluation metric must be chosen. The evaluation metric indicates the quality of the model. To choose the evaluation metric, the purpose of the predictions made by the model must be considered.

In this context, the figures extracted from a payment are classified as positive if they represent the total price. While a perfect model would classify as positive just the figures indicating the total price, in practice, some of the positive classified figures might be false positives. When implementing the model, from all the figures extracted from a payment, the positive classified figures would be potential candidates to be the total price. Negative classified figures are considered to not be the total price of the payment.

Then, from all the positive classified figures taken from the document, an algorithm chooses one to become the final prediction. Such an algorithm is simple and does not use machine learning to make the selection of the figure. Its implementation can be found at the appendix A.

According to the mentioned context, the machine learning model which is the most optimal in this scenario is that one which is able to find the biggest number of true positive instances and minimise the number of false negatives.

The reason why this is preferred instead of minimising the number of false positives, is because while a false positive can still be filtered out by the algorithm exposed at appendix A, a false negative would no longer be processed, and therefore, there are greater chances to obtain an incorrect prediction after the selection done by the algorithm.

The figure 3.5.1 illustrates an example of a model which would minimise the number or false negatives and maximise the detection of positives. It can be observed that the algorithm is able to provide the correct total price even if some of the positively classified figures are false positives.

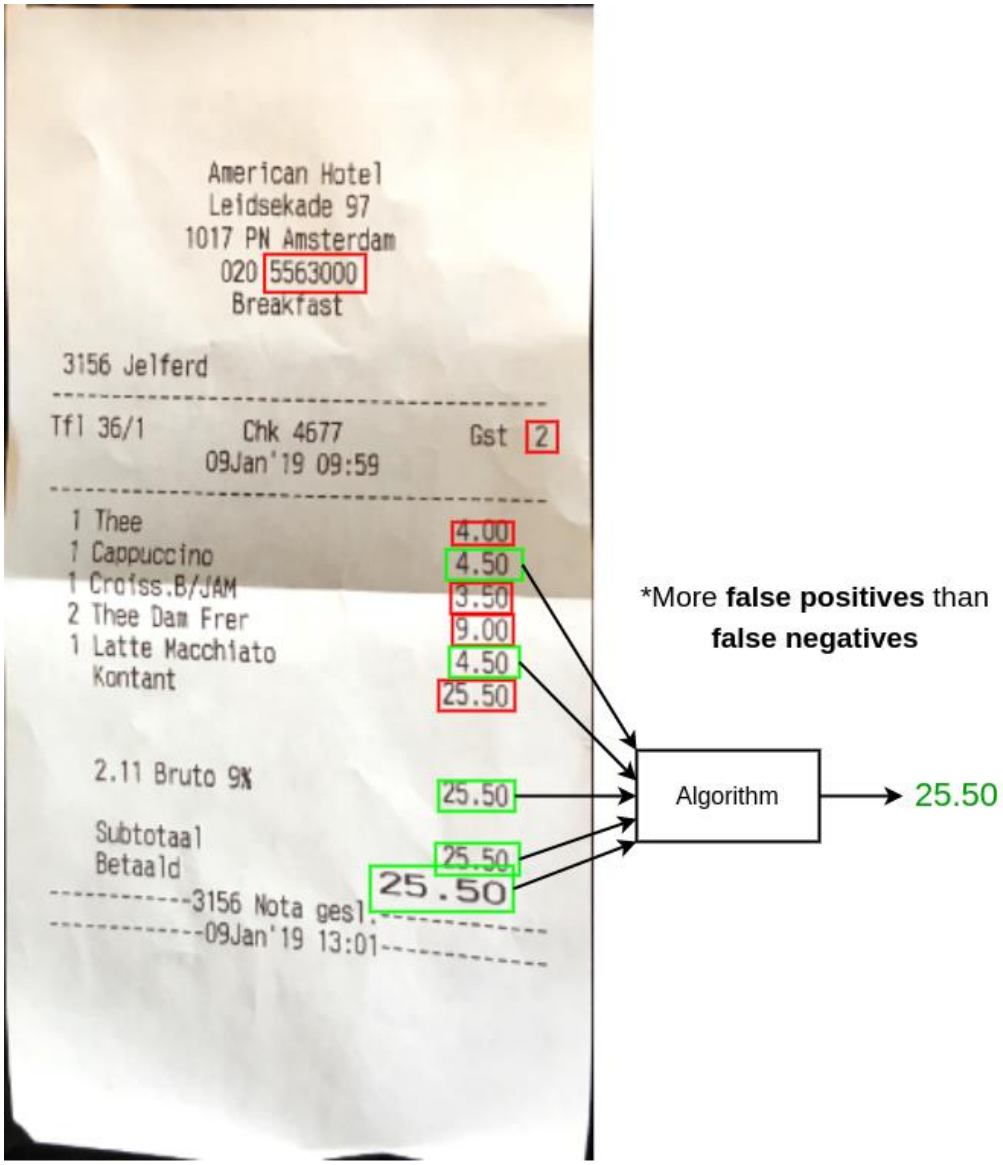

Figure 3.5.1 Model which minimises the number of false negatives

Recall is the evaluation metric that indicates the ability of the classifier to find the positive instances, and it is calculated using the formula 3.5.1 [23].

$$Recall = \frac{tp}{tp + fn}$$

Formula 3.5.1 Recall

However, recall has a weakness. If a model would predict no negative values, it would have a recall of 1 (perfect score). That would not be representative of how well the model classifies instances. To avoid that, having a balanced dataset is extremely important in order to make the model classify both positive and negative values [24]. Additionally, the ratio of positive classified instances can provide reliability to the evaluation metric.

ii. Dataset split

To calculate the performance of the generated models, the dataset must be split into a train dataset and a test dataset. When comparing the performance of the models, they need to be evaluated with the same test set containing unseen data by the models [25]. When running the search algorithm to find the optimal hyperparameters for each model, just the training set is used to ensure that the test set remains unseen. The figure 3.5.2 illustrates the split for both phases of the evaluation.

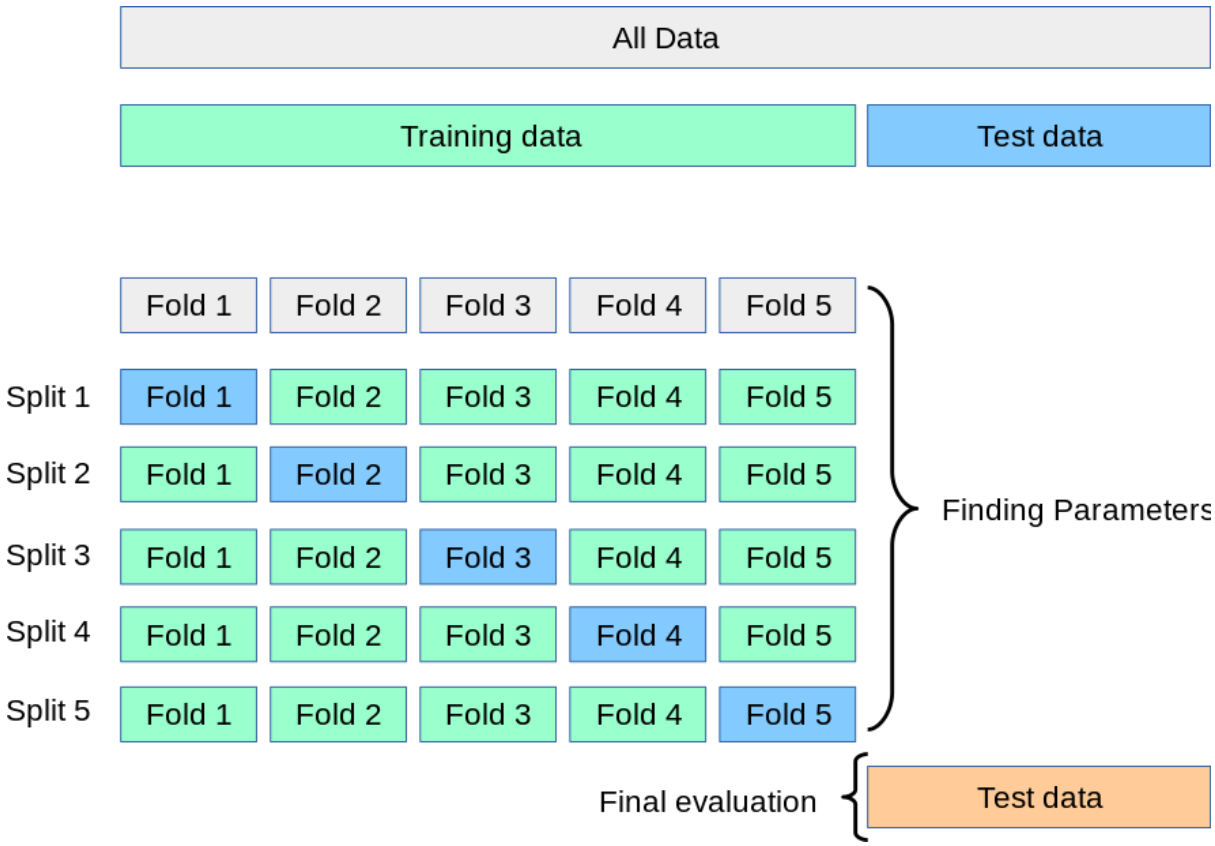

Figure 3.5.2 Data split diagram from
https://scikit-learn.org/stable/modules/cross_validation.html [4]

iii. Visualisation methods

Additionally, to complement the evaluation results, visualisation methods can be used to help with the understanding of how well the selected machine learning model performs. In this thesis, because of the binary classification problem evaluated with recall, a confusion matrix and a ROC curve are presented.

A confusion matrix is a square matrix in which its rows represent the actual class of the instances and the columns their predicted values [26]. It is a useful tool to directly visualise all the performance rates in a single matrix.

A ROC curve, or receiver operating characteristic, is a graphical representation of the relationship between false positive and true positive rates. Such a

relationship is plotted by the true positive rate in the y-axis against the false positive rate in the x-axis. This often results in a curve located in the upper left triangle above the $y = x$ line. A curve which would indicate that the model is not able to make distinctions between classes, would be located in the $y = x$ line. By calculating the area under the curve (AUC), it is obtained a score which indicates how well the model discriminates [27]. It can be seen as a numeric summary of the ROC curve.

4. Results and analysis

a. What data is available?

Gekko stores user data in a Postgres database installed in a cloud server. Such a database contains diverse user data including credentials, forms, metadata, user actions, etc. There is a specific table which contains the payments registered as costs. A column in that table indicates the total price of the payment. Additionally, the table contains the date when the payment was issued, the payment type, the taxes included in the payment, the country of origin, and metadata like the user id. Another column contains the file path of the payment uploaded by the users. Both fields, the price of the payment and the file path, might be empty. The payment document is stored in a cloud storage provided by Amazon.

The total price is manually inputted by the user, or, for those users using SmartScan, reviewed before the payment is registered. Therefore, it is assumed that the field is correctly inputted, and is actually representing the total price of the payment.

The data available for analysing in this project is the content of the documents. To obtain such a content, an OCR model must be used. OCR stands for optical character recognition and is an artificial intelligence based algorithm which is able to recognise sets of characters from images or PDFs [28]. Gekko uses Google Cloud Vision as a service provider for the OCR algorithm.

By processing the response obtained from the OCR algorithm, it is possible to match the user registered total price with the strings of characters obtained from the documents. By further analysing the payment documents, it can be observed some relevant factors from each of the numeric figures which appear in the document. These factors are:

- The font height of the figure
- The number of times that the figure appears in the document
- The height in the document where the figure appears

Furthermore, there are some key words which semantically indicate the total price from a payment. These key words are:

- Totaal
- Totaalbedrag
- Pin
- Maestro
- Subtotaal

By relating a key word with a numeric figure, more factors can be obtained. These factors are:

- The angle formed from the figure, the key word, and the horizontal axis
- The distance between the figure and the word
- The key word

The figure 4.1.1 illustrates the factors which are available from payment documents.

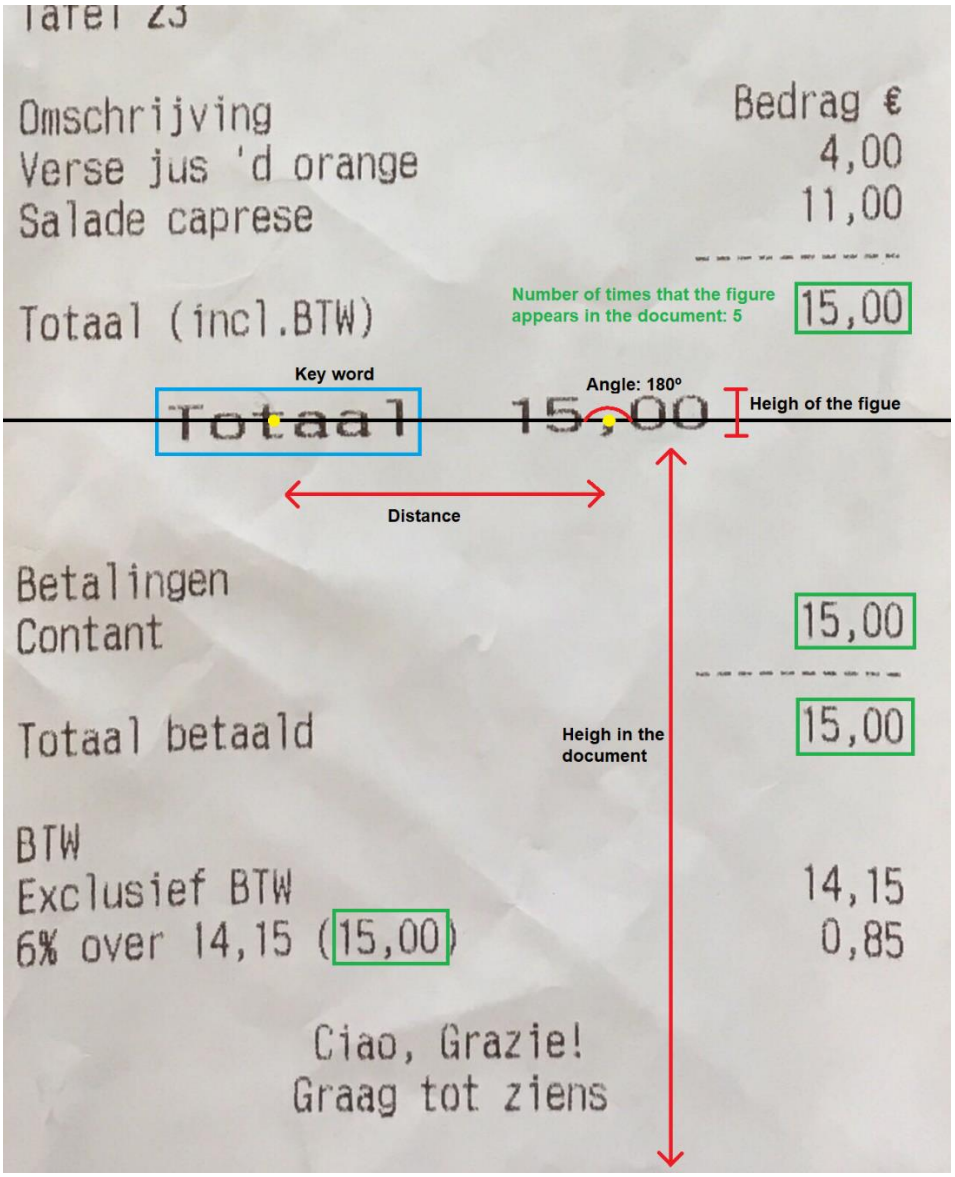

Figure 4.1.1 Available data from a payment document

While other factors could be used to train a machine learning model, these have been considered to be the ones which can provide the information needed to make distinctions between classes. Moreover, some other factors like the issuer of the payment, cannot be used because there is no data available to relate and identify strings of characters with companies. Other information available from the payment documents, such as reference numbers, phone numbers or addresses, do not provide any relevant information to determine whether a figure is the total price of the payment or not. Furthermore, an approach based on natural language processing has been rejected because of the impossibility to make distinctions between figures from the same document.

b. Which machine learning algorithms can be used to train the model?

In this section, a theoretical selection of some machine learning algorithms is made. It is known from the approached research problem that the resulting model must be a binary classifier. Moreover, from the results obtained answering the subquestion A, it is known that the training dataset does not have a high-dimensionality and contains continuous numeric variables.

From the factors available in payment documents, it can be observed that they do not follow a normal distribution. The histograms for each of the numeric factors can be found at the appendix B. Thus, machine learning algorithms which do not assume data being normally distributed must be selected.

From the correlation analysis exposed at the section 4.c.iv, it can be assumed the independence of the predictors. When considering the training time of the algorithm, because the model is being implemented at the back-end of the organisation's system, it is convenient to use an efficient algorithm to avoid latency problems if the server needs to be restarted.

Consequently, neural networks are excluded from the data mining process because of their inefficient training times [29].

Support vector machines were excluded because, it could not be proven an optimal linear separation between classes, as the dataset generated is low dimensional. On the other hand, non-linear SVMs are more suitable for high dimensional datasets [30].

The naive bayes classifier is excluded because it is better suited for datasets with categorical inputs and low entropy feature distributions (level of uncertainty in the variables). It could help to overcome dependencies between features but this is not a problem as it can be observed from the correlation analysis at section 4.c.iv. [31].

According to the mentioned context, the chosen algorithms are the logistic regression, the K nearest neighbours, the decision trees, and the random forest.

The logistic regression is chosen because it is well suited for low dimensional datasets with continuous variables. However, it is a restrictive algorithm and might not perform well if the data points are not linearly separable [32].

On the other hand, non-parametric algorithms such as decision trees and the K nearest neighbours (KNN) can adapt better to other data point shapes and they do not require any data distribution [33].

The KNN is well-suited for low dimensional datasets, but it can get slower training times if the number of instances is high [34]. Nevertheless, such training time is expected to be affordable for the organisation's system.

Decision trees seem to be efficient and tend to work well with low dimensional datasets. However, they can get unstable with small variations in the data and have the risk of training an overfitted model [35].

To overcome this problem, the random forest algorithm is an ensemble method which combines many decision trees with a depth limit in order to avoid training a single overfitted decision tree [36]. The disadvantage from random forests is that they can be slower at training the model, though it is expected to be an affordable training time.

c. Which preprocessing steps are required to generate a dataset?

To generate a dataset that can be used to train the machine learning models, an ETL process is executed. ETL stands for extract, transform, and load, and is the process for extracting data from multiple sources, transforming it into a usable format, and load it into a target database or storage [37].

i. Data extraction

There are two sources from which the data must be extracted (a database and a file storage). The total volume of storage which Gekko currently has is over 1,5 terabytes. For computing limitations, just a selection of this data is used for this project. Then, the selection is proven to be representative at section 4.c.ii.

To make the selection of which files need to be extracted, first, the database is cloned into a localhost. By querying the database, it is obtained the file path of the payment document and its total price. Many costs registered by users cannot be used for further analysis because of inconsistencies in the data. About **8,4%** of the costs do not have a

document attached, or they contain a document with an incorrect file format, or they have either a negative or a null price. To avoid retrieving these costs, a condition is added to the query in order to gain data quality. Such a condition is that the costs must be registered by active users which have already registered more than 1000 costs. Moreover, the file path must be non-null and the total price non-null and non-zero.

After running this query, **395484** costs from a total of **1866725,** were selected. To narrow more the selection of data, from the retrieved costs, **10000** were randomly selected. The randomisation method is obtained from the python package "random" which is able to generate pseudorandom numbers for various distributions [38].

Then, by using the file paths from the retrieved costs, the documents are downloaded using a script which connects to the cloud server.

ii. Representativeness of the selected data

To prove that the sampled dataset is representative of the whole group of costs, a hypothesis test is conducted. The test evaluates the total price of the payments and is approached as a hypothesis test regarding the population mean. The compared populations are:

- All the payments which have a non-null, non-zero, and non-negative registered total price.
- And the 10000 selected payments.

It is set as a two-tailed test with a level of significance of $\alpha = 0.05$.

$$H_0: \mu = \mu_0$$
$$H_1: \mu \neq \mu_0$$

Note that by rejecting the null hypothesis, the sample dataset would be proven to not be representative of the whole group.

Before calculating the test statistic, it must be stated that the sample is randomly obtained, its size is bigger than 30, and that the values are independent from each other [39].

After conducting the hypothesis test the obtained p-value is **0.84381**. Therefore, the null hypothesis is not rejected which means that the sample dataset is representative of the whole group.

iii. Data transformation

To transform the extracted data into a usable format, the content from the documents needs to be converted into machine encoded data. To do that, each document is sent to Google Cloud Vision via an API call

which returns the content in a string format. The returned string is formatted as a JSON object from which attributes represent a string of characters found in the document and their values are the coordinates where the characters are found. More details about the API can be found at https://cloud.google.com/vision/docs/ocr.

Then, the JSON object is processed to generate dataset instances with the factors presented at the subquestion A. To generate the class attribute, the process iterates through all the attributes from the JSON object. If an attribute is a numeric figure, it is further processed, and a dataset instance is generated from it. The class value is obtained by matching the value of the numeric figure with the user registered total price. If there is a match, that instance is classified as true. Otherwise, it is classified as false.

Note that, for each document processed, several dataset instances are generated (one for each figure found in the document). That results in a dataset containing **161346** instances generated from the **10000** selected documents. In the next section 4.c.iv, data reduction steps are applied in order to obtain a balanced dataset and ensure that each instance is independent from each other. The dataset is stored in a CSV format in a localhost.

iv. Data cleaning

To keep the independence among all the instances from the dataset, they all must be taken from different documents. To do that, a script removes rows which are generated from the same files (and keeping just one). The script uses the document file name as an identifier. Note that, a payment document might contain many numeric figures, if just one is kept, it means that a significant number of rows are removed from the dataset.

Furthermore, the script also ensures that the resulting dataset has the same number of positive and negative instances (providing a balanced dataset). In total, **154671** instances were removed resulting in a dataset with **6675** instances.

For each of the numeric predictors, outlier values were detected using the z-score formula. A value is considered to be an outlier value or an extreme value if the absolute value of the calculated z-score is above 3. After running this process, **362** rows were removed, resulting in a dataset with **6313** instances. In spite of this reduction, the balance between classes is kept.

Following the same methodology from the section 4.c.ii, the dataset is proven to be representative of the whole group. This time the comparison is made between the registered prices of all the payments,

and the resulting dataset with 6313 instances. The p-value obtained is **0.65104**.

Then correlation analysis is executed in order to detect possible dependencies between the predictors and avoid collinearity. Because the predictors are non-normally distributed, the Spearman's correlation assesses monotonic relationships (linear or nonlinear) between ranked variables and its values are encompassed in the range [-1, 1] [40]. A correlation below -0.6 or above 0.6 is considered to be strong. It is calculated using the formula 4.3.1.

$$r_s = \rho_{R(X),R(Y)} = \frac{cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}$$

Formula 4.3.1 Spearman's correlation

The table 4.3.1 shows all the correlations between the predictors from the dataset.

	Height	Angle	Distance	Figure Count	Position
Height	1	0.13857	0.20132	0.0465	-0.09504
Angle		1	-0.12847	0.22589	-0.17439
Distance			1	-0.05521	0.09522
Figure Count				1	0.01718
Position					1

Table 4.3.1 Spearman's correlation matrix

Because no strong correlations have been detected, all the attributes from the dataset are kept.

Finally, in order to scale all the predictor values within the same range, Min-Max normalisation has been applied. The resulting dataset has **6313** rows, and **7** columns being **5** numeric predictors and **1** that is nominal. Note that the nominal predictors are converted into numeric labels. The class attribute is a binary value, and all the predictors have a value between 0 and 1.

d. Which set of hyperparameters are the optimal for each of the machine learning algorithms?

For each of the machine learning algorithms selected at section 4.b, a set of hyperparameters optimal for the performance of the model is searched. The method used to find the best parameters is the random search method with stratified K-fold cross validation.

In this section, the set of optimal hyperparameters for the random forest algorithm are presented. A random forest model is the resulting best model as it is stated at section 4.e.

The random forest algorithm is based on training many decision tree models and making predictions by averaging their output values. The number of decision trees which the algorithm generates is a hyperparameter. Another hyperparameter is the maximum depth that the decision trees can have. This parameter prevents them from being overfitted [41]. Moreover, at each node split of the trees, a limit can be set to the number of features which can be considered. There are other parameters which can be tuned although they do not have as much impact on the performance of the model [42]. In this thesis, just the above mentioned are searched for the random forest algorithm.

As a result, the optimal number of trees is **81**, their maximum depth **108**, and the maximum number of features to be considered in each node split is equal to the square root of the total number of features from the dataset.

e. How can the performance of the predictive model be measured?

Once the optimal hyperparameters are found for each of the machine learning algorithms, new models are trained and evaluated with yet unseen data. From the evaluation results, a single model is selected.

The test data used at this point is obtained from splitting the dataset into a training set and a test set. The test set represents 40% of the whole dataset. No cross validation is used because it is convenient to obtain the performance of a single model which can be implemented at Gekko's system.

The performance results from the hyperparameter tuning with cross validation are no longer considered because, for model selection, it is required to provide the performance results of the model which is finally implemented. That also reduces the probability of selecting a model which performs well when searching the hyperparameters, but that is unable to provide accurate predictions with unseen data. The reason why this would happen is because the score obtained from optimising the hyperparameters is biased and not a good estimator of the generalisation [41].

The table 4.5.1 shows the recall obtained for each of the generated machine learning models.

	Logistic Regression	K Nearest Neighbours	Decision Tree	**Random Forest**
Recall	0.86071	0.84759	0.86071	**0.89777**

Table 4.5.1 Performance results

As it can be observed, the random forest model was the model with the best performance results. Therefore, the random forest is the finally selected one.

In order to provide veracity to the presented performance results, the rate of positively classified figures is calculated. Due to the weakness of recall mentioned at section 3.e.i, this rate is calculated to prove that the model is able to predict both positive and negative classes. The calculated rate of positive predictions is **0.48812**. Thus, the obtained performance in recall has veracity and is a fair evaluation of how well the model performs.

More evaluation rates can be seen broken down into a confusion matrix at the figure 4.5.1.

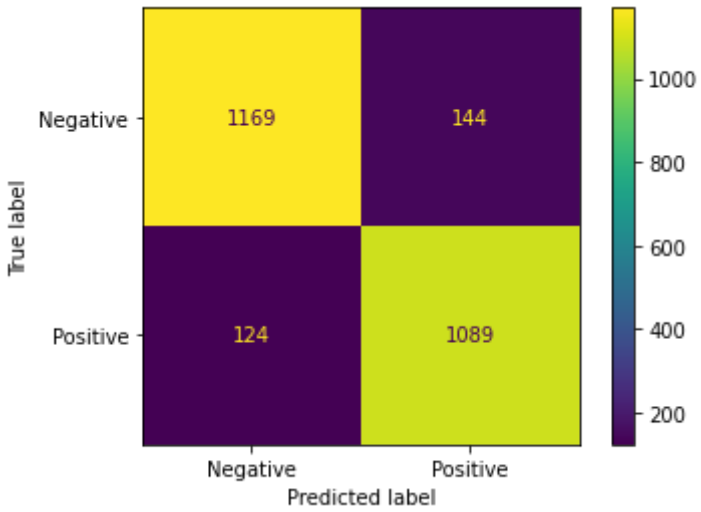

Figure 4.5.1 Confusion matrix

Moreover, from the ROC curve at the figure 4.5.2, it can be observed that the model is able to make clear distinctions between the classes. The calculated area under the curve is **0.955**, being a satisfactory score for model performance.

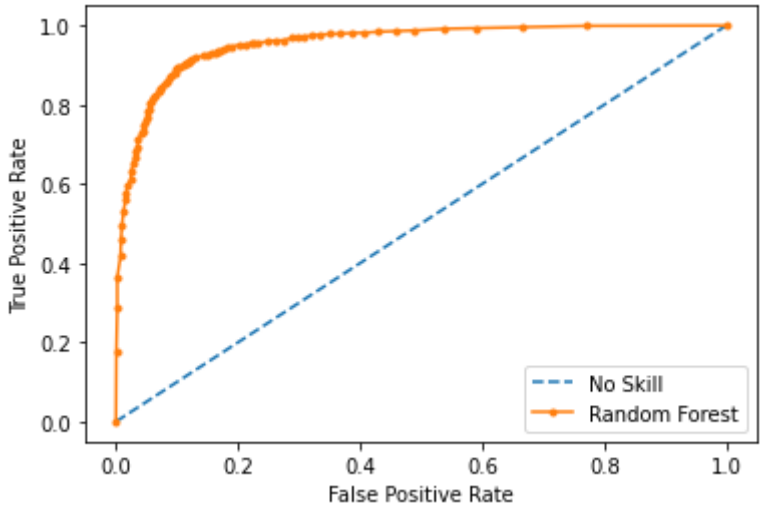

Figure 4.5.2 ROC curve

5. Conclusion

By answering the first subquestion, it is concluded that the data available is distributed over different sources at the organisation's system. From all the available data, the scope of interest for this thesis is only the payments registered by users. The sources from which data can be extracted are a database and a cloud file storage. The database contains records of costs registered by users where the total price of the payment can be obtained. Moreover, it can also be found a file path pointing to the cloud file storage where the payment documents are stored.

The factors which can be used to train a supervised binary classifier are based on the analysis of document content. These factors describe each of the numeric figures found in a document, which can be positively classified if they do represent the total amount of the payment. Some of the factors relate certain key words with the figure. Some other factors are not considered because they do not provide relevant information in order to make distinctions between figures.

To find which algorithms could be used to train a supervised binary classifier, it was taken into consideration the size of the training data, the restrictiveness of the mapping function of the algorithm, and the training time of the algorithm.

The selection of algorithms excluded neural networks to avoid high training times, the support vector machines because it was not clear if the data shapes could satisfy an optimal separation between classes, and the naive bayes because it is not suitable for the dataset characteristics. On the other hand, the logistic regression, the K nearest neighbours, and decision trees, were selected because they are appropriate for low dimensional datasets with numerical features. Being the K nearest neighbours and the decision tree more flexible, adapting to a wider range of data point shapes. However, those algorithms also have some disadvantages, and the ensemble method random forest was also selected to overcome some of the problems presented by the rest of the algorithms.

From the subquestion c, it was stated that the preprocessing steps required to form a dataset consisted of data extraction, data transformation, and data cleaning. Thus, an ETL process was executed.

The data extraction went through a preliminary selection of data to work with smaller storage amounts and favour data quality. The selection was proven to be representative of the whole group of costs registered by users. Such a proof was done via statistical inference.

The transformation of data into a usable format was substantiated on the content of the payment documents. By using an OCR algorithm, it was possible to get the content from the documents in a string format. By processing those contents, dataset instances were generated. Each instance represents a numeric figure from a document. Moreover, a figure is classified as positive if the figure is the total price of the payment. The class attribute is obtained by comparing the value of the figure with the user's registered total price.

To obtain a dataset with independent instances, just instances from different documents were kept. Then, the data cleaning steps were executed, beginning with the undersampling technique. Undersampling reduced the size of the original dataset providing a balanced dataset and ensuring the independence among the instances. The balanced dataset removed the possible bias towards the majority class and the underrepresentation of the positive instances. Then, an outlier removal technique was applied. By using the centrality measure z-score, many rows of the dataset were removed but keeping the balance between classes. Continuously, the conducted correlation analysis did not show any strong correlation between predictors. Therefore, all the predictors from the dataset were kept. Note that, it is also concluded that the appropriate correlation to be used is the Spearman’s rank correlation due to the non-normal distribution of some of the predictors. Finally, the predictor values were normalised in order to scale them in the same range.

The method used to search the optimal hyperparameters of the machine learning algorithms was the random search with stratified K-fold cross-validation. The random search tried different combinations of parameters, instead of trying all of them, to save computational efforts. Stratified K-fold cross-validation avoided optimising the hyperparameters with overfitted models while evaluating on balanced datasets for each of the folds. It is concluded that, the optimal set of hyperparameters found for the random forest algorithm sets the number of decision trees to 81, the maximum depth for each tree to 108, and the maximum number of features to be considered at each node split is set to be equal to the square root of the number of features from the dataset.

The evaluation metric that was considered the most appropriate, according to the context in which the model is implemented, is the recall. Optimising the model parameters, and doing model selection based on recall, minimised the false negative rate of predictions. For the model selection phase, it is concluded that evaluating on a single test dataset is more convenient to obtain the performance results of the model

which is going to be implemented. Using 40% of the dataset for testing and the remaining set for training the models, it was concluded that the model with the best performance is a random forest with a recall of 0.8977 and 48.8% of positive classifications. Furthermore, the presented ROC curve, and the calculated area under the curve (AUC=0.955), suggest that the random forest model is able to discriminate between classes.

By answering the main research question, it is concluded that, to generate a predictive model to classify figures from a payment document, several data mining techniques must be applied. Starting by identifying what data is available, narrowing it to the scope of interest, and selecting which factors can be used to train a model. Secondly, preselecting some machine learning algorithms based on theoretical knowledge. After that, the data to be used for training is extracted from the sources. This extracted data is a selection of the whole set and must be proved to be representative by statistical inference. From its raw format, the data must be transformed into a dataset with a binary class attribute. Then, data cleaning steps favour the training process to fit a model which is able to make clearer distinctions and generalise well. This is achieved by applying correlation analysis, removing outlier values, normalising the predictors and undersampling the dataset to avoid the underrepresentation of one of the classes. Continuously, the hyperparameters of the machine learning algorithms can be optimised with a random search method. The random search might be used with stratified K-fold cross validation in order to prevent from overfitting and validate with balanced sets. The finding of optimal hyperparameters and the model selection can be based on recall to generate a model which has the ability to find the biggest number of positive instances. Finally, the performance results of the model can be complemented with visualisation methods like the ROC curve to provide veracity to the results.

6. Discussion

a. Training set reductions

During the preprocessing phase, three data reduction steps are applied. The first step is applied by the selection of payments from all the payments registered by users. At that phase, no datasets have been generated yet. For this reason, the only value that could be used to prove that the selection of data is representative of the whole set is the total price of the payments (manually registered by users). Other factors from the final dataset are not tested, and therefore, the representativeness of the selection of documents with respect to these factors cannot be proved.

The second data reduction step takes place when applying the undersampling technique. The resulting dataset has an equal number of positive and negative classified instances, while the original dataset is imbalanced. That is what the undersampling technique wants to achieve, which means that by nature, the resulting dataset is not representative. The last data reduction step is applied when removing outlier values from the predictors.

In spite of the difficulties proving that the finally obtained dataset is representative, the same experiment done at section 4.c.ii was executed again with satisfactory results.

b. Evaluation strategy

Different evaluation techniques have been implemented during the data mining process. There are two evaluation steps. One applied to the hyperparameter tuning phase, and the other one applied at the model selection.

For each evaluation, the chosen technique is considered to be the most appropriate according to what the evaluation wants to achieve. At the hyperparameter tuning phase, the technique chosen is the stratified K-fold cross-validation. As the goal is to find the optimal parameters for the models, an evaluation technique which prevents overfitting is required to avoid obtaining parameters which are not able to generalise well on unseen data. However, there are many different variants of cross-validation.

Stratified K-fold cross-validation was chosen to test on a balanced dataset. Another possibility was dividing the dataset into many training and testing sets, and obtaining the performance results from averaging the evaluations. This would try to improve cross-validation by avoiding training the models on repeated training data. Nevertheless, cross-validation was considered to be sufficient for finding the optimal parameters as it never tests on repeated data.

Furthermore, the evaluation strategy implemented in this thesis project follows the official guidelines from the machine learning library used [43]. Such a library is one of the most used libraries for machine learning, and its documentation and guidelines were considered as a reliable source of information.

c. Scalability of the product

When analysing the scalability of the new version of the functionality, it can be stated that more models which clarify other figure types, can be generated using the same techniques. The data extraction process does not need to be executed again, and by reusing the same scripts it is possible to generate new datasets adapted to the figure type which is intended to be classified. The only change which the new models would require is the list of key words that identify the figure. For example, a model which classifies figures which indicate the VAT of the payment would have the following key words (VAT, BTW, IVA, …).

Consequently, the models could be updated allowing more languages by adding the translated key words to the list of words. This is a significant improvement from the previous version of the functionality as it required development efforts and technical knowledge to update the scope of the predictions.

7. References

1. Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.

2. Lentini, Salvatore, Enrico Grosso, and Giovanni L. Masala. "A comparison of data fragmentation techniques in cloud servers." International Conference on Emerging Internetworking, Data & Web Technologies. Springer, Cham, 2018.

3. Vianna, Daniela, et al. "A tool for personal data extraction." 2014 IEEE 30th International Conference on Data Engineering Workshops. IEEE, 2014.

4. Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

5. Kaikhah, Khosrow, and Sandesh Doddameti. "Discovering trends in large datasets using neural networks." Applied Intelligence 24.1 (2006): 51-60.

6. Braun, Andreas Ch, Uwe Weidner, and Stefan Hinz. "Classification in high-dimensional feature spaces—Assessment using SVM, IVM and RVM with focus on simulated EnMAP data." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 5.2 (2012): 436-443.

7. Chirici, Gherardo, et al. "K-NN FOREST: a software for the non-parametric prediction and mapping of environmental variables by the k-Nearest Neighbors algorithm." European Journal of Remote Sensing 45.1 (2012): 433-442.

8. Schmitt, Erick J., and Hossein Jula. "On the limitations of linear models in predicting travel times." 2007 IEEE Intelligent Transportation Systems Conference. IEEE, 2007.

9. Prusa, Joseph, Taghi M. Khoshgoftaar, and Naeem Seliya. "The effect of dataset size on training tweet sentiment classifiers." 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015.

10. McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

11. Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches." IEEE Data Eng. Bull. 23.4 (2000): 3-13.

12. Yap, Bee Wah, et al. "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets." Proceedings of the first

international conference on advanced data and information engineering (DaEng-2013). Springer, Singapore, 2014.

13. Rousseeuw, Peter J., and Mia Hubert. "Robust statistics for outlier detection." Wiley interdisciplinary reviews: Data mining and knowledge discovery 1.1 (2011): 73-79.

14. Thompson, Glenn L. "An SPSS implementation of the nonrecursive outlier deletion procedure with shifting z score criterion (Van Selst & Jolicoeur, 1994)." Behavior Research Methods 38.2 (2006): 344-352.

15. Dormann, Carsten F., et al. "Collinearity: a review of methods to deal with it and a simulation study evaluating their performance." Ecography 36.1 (2013): 27-46.

16. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

17. Jayalakshmi, T., and A. Santhakumaran. "Statistical normalization and back propagation for classification." International Journal of Computer Theory and Engineering 3.1 (2011): 1793-8201.

18. Claesen, Marc, and Bart De Moor. "Hyperparameter search in machine learning." arXiv preprint arXiv:1502.02127 (2015).

19. Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of machine learning research 13.2 (2012).

20. Berrar, Daniel. "Cross-Validation." (2019): 542-545.

21. Rodriguez, Juan D., Aritz Perez, and Jose A. Lozano. "Sensitivity analysis of k-fold cross validation in prediction error estimation." IEEE transactions on pattern analysis and machine intelligence 32.3 (2009): 569-575.

22. Purushotham, Swarnalatha, and B. K. Tripathy. "Evaluation of classifier models using stratified tenfold cross validation techniques." International Conference on Computing and Communication Systems. Springer, Berlin, Heidelberg, 2011.

23. Vakili, Meysam, Mohammad Ghamsari, and Masoumeh Rezaei. "Performance analysis and comparison of machine and deep learning algorithms for iot data classification." arXiv preprint arXiv:2001.09636 (2020).

24. Burns, Nicola, et al. "Sentiment analysis of customer reviews: Balanced versus unbalanced datasets." International Conference on Knowledge-Based and

Intelligent Information and Engineering Systems. Springer, Berlin, Heidelberg, 2011.

25. Xu, Yun, and Royston Goodacre. "On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning." Journal of analysis and testing 2.3 (2018): 249-262.

26. Caelen, Olivier. "A Bayesian interpretation of the confusion matrix." Annals of Mathematics and Artificial Intelligence 81.3 (2017): 429-450.

27. Hoo, Zhe Hui, Jane Candlish, and Dawn Teare. "What is an ROC curve?." Emergency Medicine Journal 34.6 (2017): 357-359.

28. Mithe, Ravina, Supriya Indalkar, and Nilam Divekar. "Optical character recognition." International journal of recent technology and engineering (IJRTE) 2.1 (2013): 72-75.

29. Bhavsar, Hetal, and Amit Ganatra. "A comparative study of training algorithms for supervised machine learning." International Journal of Soft Computing and Engineering (IJSCE) 2.4 (2012): 2231-2307.

30. Pradhan, Ashis. "Support vector machine-a survey." International Journal of Emerging Technology and Advanced Engineering 2.8 (2012): 82-85.

31. Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.

32. Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. Applied logistic regression. Vol. 398. John Wiley & Sons, 2013.

33. Liu, Bao, et al. "Nonparametric machine learning for mapping forest cover and exploring influential factors." Landscape ecology 35.7 (2020): 1683-1699.

34. Fayed, Hatem A., and Amir F. Atiya. "A novel template reduction approach for the $ K $-nearest neighbor method." IEEE Transactions on Neural Networks 20.5 (2009): 890-896.

35. Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction." Shanghai archives of psychiatry 27.2 (2015): 130.

36. Biau, Gérard, and Erwan Scornet. "A random forest guided tour." Test 25.2 (2016): 197-227.

37. Gour, Vishal, et al. "Improve performance of extract, transform and load (ETL) in data warehouse." International Journal on Computer Science and Engineering 2.3 (2010): 786-789.

38. Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.

39. Sullivan, Michael. Statistics: Informed decisions using data. Prentice Hall/Pearson, 2018.

40. Astivia, Oscar L. Olvera, and Bruno D. Zumbo. "Population models and simulation methods: The case of the Spearman rank correlation." British Journal of Mathematical and Statistical Psychology 70.3 (2017): 347-367.

41. Patil, Dipti D., V. M. Wadhai, and J. A. Gokhale. "Evaluation of decision tree pruning algorithms for complexity and classification accuracy." International Journal of Computer Applications 11.2 (2010): 23-30.

42. Probst, Philipp, Marvin N. Wright, and Anne-Laure Boulesteix. "Hyperparameters and tuning strategies for random forest." Wiley Interdisciplinary Reviews: data mining and knowledge discovery 9.3 (2019): e1301.

43. Cawley, Gavin C., and Nicola LC Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation." The Journal of Machine Learning Research 11 (2010): 2079-2107.

8. Appendix

a. Algorithm which selects positive classified figures

As it is explained at section 3.e.i, the prediction model presented in this thesis classifies all the numeric figures found in a payment document. If a figure is classified as positive, it is a potential candidate for being the total price of the payment. Then, a simple algorithm selects a figure among all the positive classified figures.

Such an algorithm obtains as an input a list of figures that were positively classified. As an example, from the payment shown at figure 8.1.1, the following list of figures would be passed as an input to the method: [43.57, 43.57, 419741].

Figure 8.1.1 Example of inputs and output to the algorithm

There are two comparisons made to decide which is the finally selected figure. First, it is counted the number of times that a figure appears in the input list.

The figure which is repeated the most is returned as the finally selected one. The concept behind this comparison is that the figure which is repeated the most, is more likely to be the total price. This fact can also be observed from the histogram of the number of occurrences of figures at appendix B (figure 8.2.3).

If two figures are repeated the same number of times, the figure with the greatest value is selected as the total price of the payment. This comparison is made expecting that the greatest positively classified figure from a payment document is more likely to be the total price of the payment.

While these comparisons would make logical sense, they can be interpreted as arbitrary, and they could cause some prediction errors. However, the purpose of the algorithm is to simply select one figure among all the candidates while most of the work is being done by the classifier.

b. Histograms of predictors

In this appendix section, the histograms for each of the numerical predictors from the dataset are presented. They show the data distributions for both classes (figures which indicate the total price of the payment, and figures which do not indicate the total price of the payment). These histograms helped to have an overview of the data distribution of the factors used for modelling and make decisions based on that. Find them below at figures 8.2.1, 8.2.2, 8.2.3, 8.2.4, and 8.2.5.

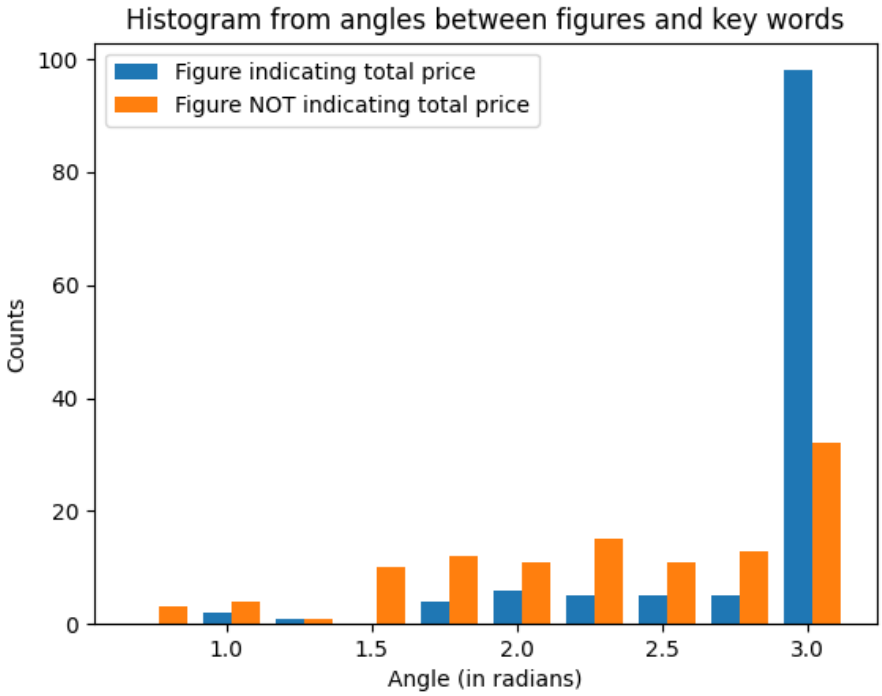

Figure 8.2.1 Angles between figures and key words

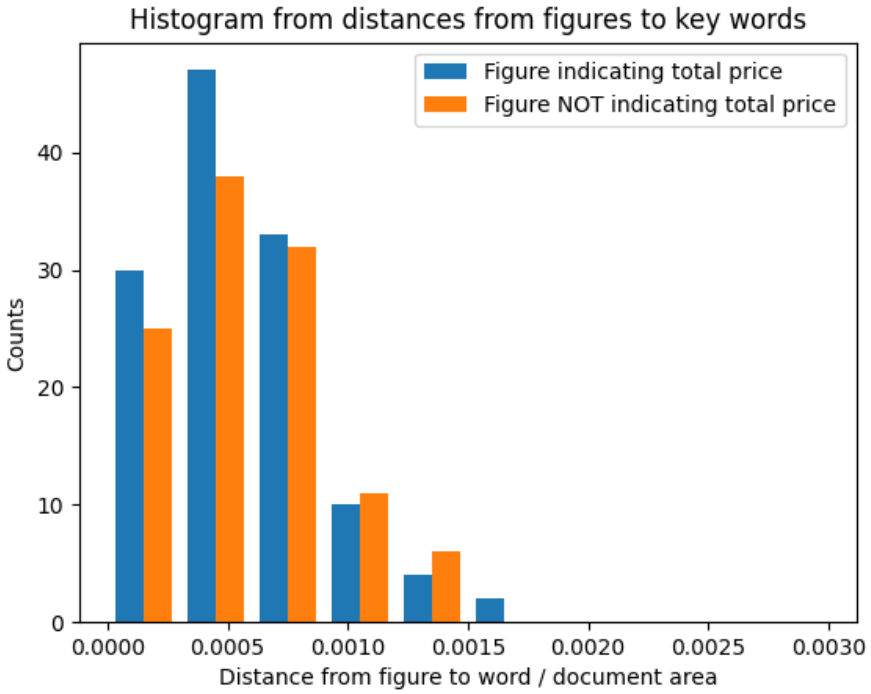

Figure 8.2.2 Distances from figures to key words

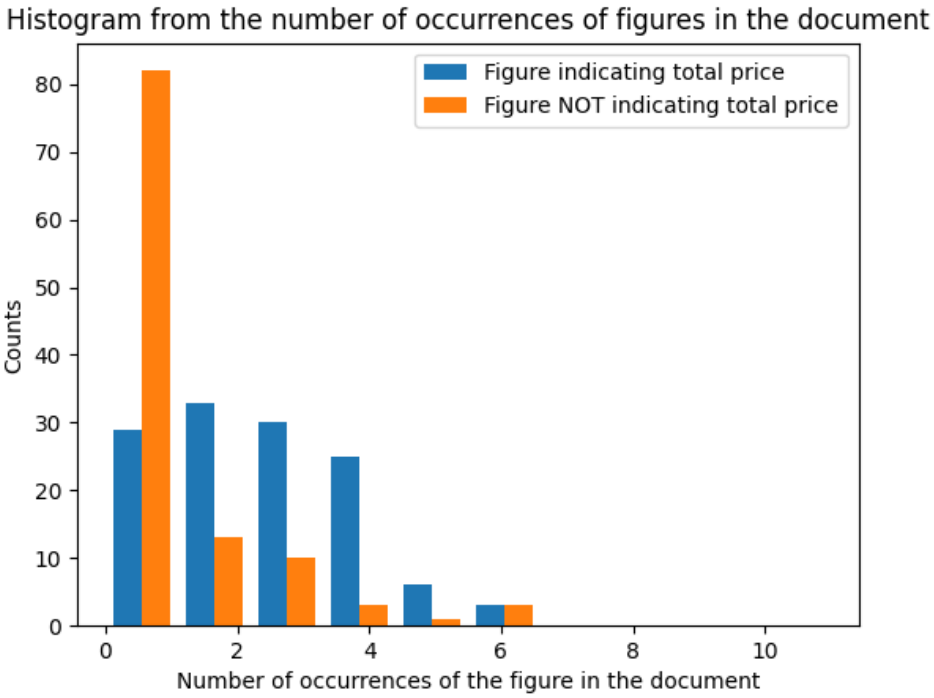

Figure 8.2.3 Number of occurrences of figures in the document

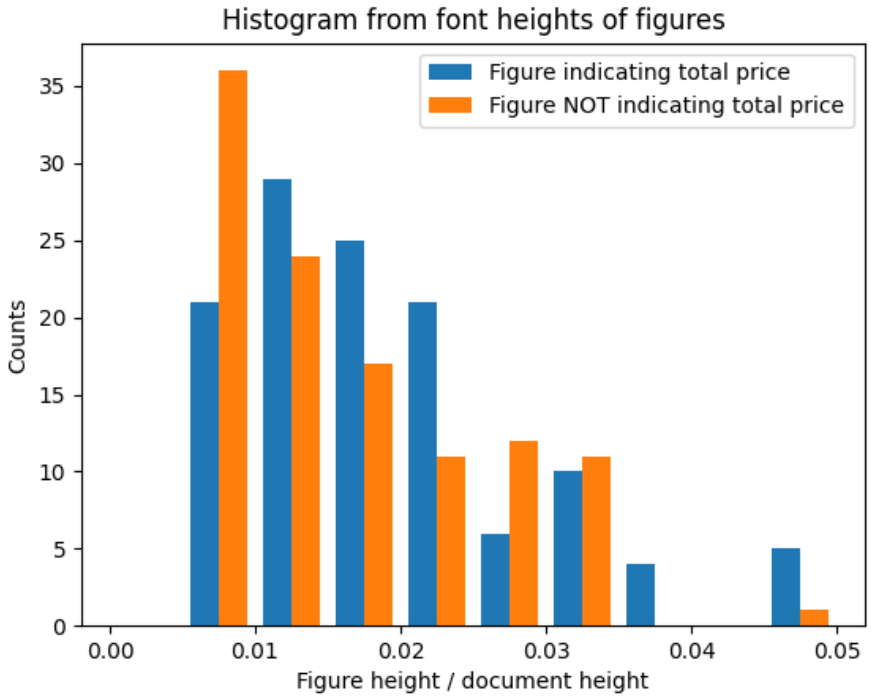

Figure 8.2.4 Font heights of figures

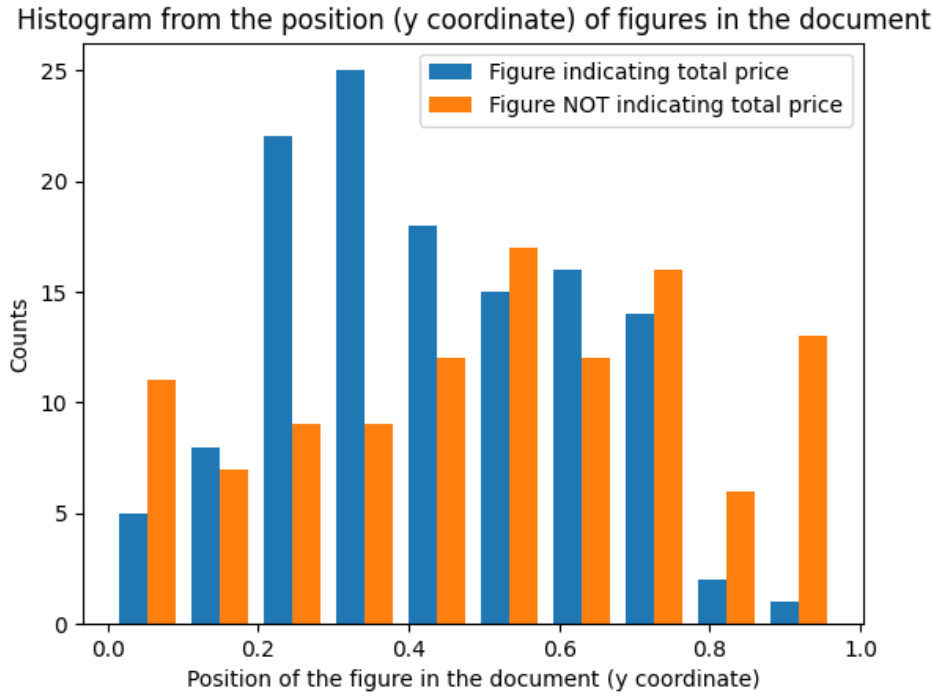

Figure 8.2.5 Position of figures in the document