VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS MASTER'S STUDY PROGRAMME

Master's thesis

Įeinančių mokėjimų klasifikavimas

Classification of Incoming Payments

Julija Kolesnikova

Supervisor Doc. Dr. Vytautas Kazakevičius

Vilnius, 2021

Abstract

Classification of the incoming payments to the private customers of the bank is essential for customer behaviour and income analysis. However, more often banks classify the outgoing transactions and only specific categories of customer incomes.

This thesis proposes possible categories of incoming transactions and investigates how do different algorithms perform at classification.

There were manually obtained 21 new class of customer incomes. The Support Vector Machine, Naïve Bayes and Gradient Boosting classification algorithms show promising results and can be implemented in the bank instead of rule-based approach.

The main focus in the future work is to reduce a noise in the textual data by automatic correction of mistakes in the transaction description.

Key words: incoming payments, machine learning, classification, text analysis, Natural Language Processing

Santrauka

Įeinančių mokėjimų privatiems banko klientams klasifikavimas yra būtinas klientų elgsenos ir pajamų analizei atlikti. Tačiau bankai dažniau klasifikuoja išeinančius mokėjimus ir tik tam tikras klientų pajamų kategorijas.

Šio darbo tikslas yra pasiūlyti galimas įeinančių mokėjimų kategorijas ir palyginti, kaip skirtingi algoritmai veikia klasifikuojant.

Rankiniu būdu buvo gauta 21 nauja klientų pajamų klasė. Atraminių Vektorių Mašinos, Naivus Bajeso ir Gradiento stiprinimo klasifikavimo algoritmai rodo žadančius rezultatus ir gali būti naudojami banke vietoje raktažodžių metodo.

Pagrindinis dėmesys būsimame darbe yra sumažinti triukšmą tekstiniuose duomenyse, automatiškai ištaisant mokėjimo aprašymo klaidas.

Raktiniai žodžiai: įeinantys mokėjimai, mašininis mokymasis, klasifikavimas, teksto analizė, Natūralios kalbos apdorojimas

Contents

Chapter 1

Introduction

1.1 Motivation

Currently bank, which is based in Northern-European countries, is using a rule-based approach to identify private customer incomes. This approach is not covering all incoming transactions, but only specific classes such as salary, pension, grants and etc. The class of a transaction is recognized by predefined keywords and rules. This approach is quite precise, but it is needed to update the rules frequently just to stay up to date. Instead of the rule-based approach there can be used the machine learning model, which is cheaper and easier to train and adjust.

The payment recognition is needed to know the customer better and to be aware of income amount. Based on that the bank can improve services by providing appropriate products, credits and etc. The benefits of income classification also can be used in the anti-money laundering area to identify the unusual behaviour in the incoming cash flow.

1.2 Problem

The incoming payment classification problem is not so trivial, as it may seem at first glance. When we are talking about the outgoing payments, it is more important to know who is on

the "another side" – payment receiver, because when you are spending money in a grocery store, most probably you are buying a food and when you are using your card in a gas station you are buying a gasoline. When it comes to the incoming payments, it is crucial to know the reason of payment, i.e. analyse the text description of transaction. Here is the point where the main issue comes, as payment description is a free-form and its length is limited by 140 characters. The majority of payment descriptions from legal entities to private customers contains valuable textual information as "salary", "dividends", "insurance", "scholarship" and etc. But what concerns private-to-private transactions, most of them are uninformative. This problem can be solved in three main stages. Firstly, as data set provides partly-labelled data, it is needed to determine some possible payment classes, and if succeeded, to label the unlabelled instances. Secondly, it is needed to analyze transaction description – preprocess text and make it suitable for Machine Learning algorithms. And finally, to train and test Machine Learning model on the fully-labelled data set.

1.3 Research questions

The aim of this master's thesis is to answer the following research questions:

1. If there are any payment classes other than already defined by the bank?

   Since the rule-based approach is covering not all customer incomes, we wish to identify the other possible classes manually.

2. If such limited data as incoming payments, is suitable for classification?

   All related works that we found on transaction analysis, were conducted on the *fully labelled outgoing transactions from private to legal entities*, and none of them contained the *incoming payments to private customers from both private and legal entities.* Taking that into consideration the first question is, if such data can give valuable information for the classification task?

3. How do different classification algorithms, such as Naïve Bayes, Support Vector Machines, Generalized Linear Models and Gradient Boosting, compare in the payment

classification?

Considering that we have fully labelled data set, we want to compare the most popular models for classification, keeping in mind that transaction text is extremely short.

4. Can the banks' rule-based approach be substituted by the Machine Learning model?

   The goal of this thesis is not only detection of the new classes of incoming transactions, but also the proposal to use the Machine Learning model.

1.4 Limitations

All experiments were conducted on the average capability computer, using R&Rstudio software. Taking this into account the experiment was limited with computational resources and available R packages. Computational efficiency and model training time therefore are not included into result comparison. However, training time is available in the overall model evaluation metrics.

1.5 Structure of the thesis

The structure of the thesis is as follows. Chapter 2 gives a theoretical foundation and background needed for the experiment. Chapter 3 contains a short overview of the related works. In Chapter 4 we describe available data and give a structured plan of the experiment. Chapter 5 contains all the experiment results. The thesis is summarized in the last Chapter 6, by discussing obtained results and providing possible recommendations for similar tasks. Appendix includes supplementary materials.

Chapter 2

Background

2.1 Natural language processing

All the data can be divided into two groups - the structured and the unstructured. Everything people can express in a written or a verbal form, has a different complexity, meaning, slang and interpretation which makes text a good example of unstructured data. Unstructured data doesn't fit into standard row-column structure of the relational databases or tables. However, with a growth of technologies there has risen a separate field in the data science - Natural Language Processing. Natural Language Processing, or NLP for short, is defined as an automatic processing of the natural language, like speech and text, by software.

Text mining is a process that uses natural language processing for making large amounts of unstructured data into organised one. Transforming text into structured data makes it understandable for machines, so it can be used, for instance, for classification or clustering tasks. There are a few basic stages in the text processing, which help to create a larger body of organized text known as a text corpus or a collection of documents.

The main stages of the text pre-processing are: making all text lowercase, removing punctuation, numbers, extra spaces and tabs, excluding specific "stop" words, stemming, identifying the part of speech, spelling, correction of mistypes or mistakes and tokenization. The last step of NLP is representation of the data, for instance, using the *"Bag of Words"*.

Based on specific task and concrete textual data, analyst should decide which steps can be applied on the data and which cannot be used, as some very important information can be lost.

2.1.1 Text preprocessing

Almost all steps of text preprocessing are devoted to noise reduction, which helps to remove meaningless information and reduce dimensionality.
The main text preprocessing steps are:

- Making all text lowercase

  If we are not interested in finding proper nouns or named entities, the text should be lowered. For example : "John Johns goes to Starbucks, which is placed on Main ave. 23" -> "john johns goes to starbucks, which is placed on main ave. 23"

- Removing punctuation and numbers

  Dropping punctuation and numbers is one of the main steps of text cleaning, as this more often have no useful information. Taking the same example: "John Johns goes to Starbucks which is placed on Main ave 23" -> "John Johns goes to Starbucks which is placed on Main ave"

- Removing extra spaces and tabs

  Sometimes the text has extra spaces, and to prevent this to be counted as separate character or word, it is needed to keep a single space. For instance : "Black coffee" -> "Black coffee"

- Removing specific “stop” words

  Removal of specific words is a crucial step, as this helps to drop meaningless information, which is contained in almost all documents. Standard English stop words are “the”, “he”, “a” etc. However, depending on data set, an analyst should adjust list of stop words manually.

- Stemming

  The stemming is a step of dropping suffixes and prefixes of words (e.g. Transforming -> Transform, Processed -> process), which helps to reduce a dimensionality of vocabulary, without loosing the meaning.

- Tokenization

  Tokenization is a natural language processing feature that divides each document into separate word units or letters. In the Natural Language processing a word is considered as one token. [1]

  Example : "This is a cat" -> "This" "is" "a" "cat"

- Spelling correction, correction of mistypes or mistakes
  These are additional steps for correction of a possible mistakes in the document. But it should be noted, that this process is automated and mistypes can be corrected in improper way, which leads to wrong interpretation and can spoil the data for the further analysis.

2.1.2 Bag of Words

The Bag of words is a procedure of counting and encoding each word of the document. The word order and the grammatical word type are not considered in this text mining technique.[15] The Bag of Words fits the machine learning frameworks because it converts an unstructured data into an organized matrix. The main benefit of this approach is that it is generally not computationally expensive. However it has a drawback - if a dictionary is very large, the final matrix may grow to a huge sizes, since each distinct word creates a new column. The following table shows an example a Bag of Words of the three sentences:

- This payment is rejected.
- Cat is sitting on the table.
- This table is wooden.

Sentence	This	payment	is	rejected	Cat	sitting	on	the	table	wooden
1	1	1	1	1	0	0	0	0	0	0
2	0	0	1	0	1	1	1	1	1	0
3	1	0	1	0	0	0	0	0	1	1

Table 2.1: Bag of Words example

2.2 One-hot Encoding

The data may contain categorical values, for example, feature "Customer risk" can have values "High", "Medium" and "Low". Some algorithms, like hierarchical or tree-based machine learning models, can handle such data type, but most of the algorithms require only numerical values to achieve the desired results. There are a few ways to convert categorical values into numerical ones, one of them is One-Hot Encoding.

This technique converts each categorical value vector with N categories, into new N columns and assigns a 1 or 0 (true or false) value to the column. However, it can cause the number of columns to expand greatly if there are many unique values in a category column.

Customer Nbr	Risk Segment
1	High
2	Medium
3	Low
4	High

Customer Nbr	High	Medium	Low
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

Table 2.2: One-hot encoding example

2.3 Feature standardization

The distance algorithms like KNN, K-means or SVM are sensitive to the range of the feature, because they are using distances between data points to determine their similarity. In order to give for a model the appropriate data it is needed to standardize it. Standardization is a scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

The formula for standardization:

$$\hat{x} = \frac{x - \mu}{\sigma}, \tag{2.1}$$

here μ is the mean of the feature values and σ is the standard deviation of the feature values.

2.4 Supervised machine learning

The machine learning has two major branches - supervised learning and unsupervised learning. In the supervised learning, the labels of each instance are provided, which improves a performance of the model. Also, it is easier to check the model adequacy and errors. In the unsupervised learning, the labels are not available, so performance cannot be so easily measured.

2.4.1 Naïve Bayes Model

Naïve Bayes is a probabilistic supervised machine learning model, which is known by its accuracy and extra high speed. It can handle large data sets having high number of dimensions, also it shows a good performance on the sparse data, so it makes Naïve Bayes model suitable for text classification. The idea of Naïve Bayes can be expressed as probability:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.2}$$

Which formally means we can find the probability of A happening, given that B has occurred. It is called naïve because we are making presence that one particular feature does not reflect other - they are independent. With regards to multi-class data, Bayes' theorem can be applied in following way:

$$P(y_j|X) = \frac{P(X|y_j)P(y_j)}{P(X)}, \tag{2.3}$$

here $y_j = (y_1, y_2, y_3, ..., y_k)$ is multi-class variable, where k is a total number of classes, X is a feature vector $X = (x_1, x_2, x_3, ..., x_n)$, so the equation can be stated as following:

$$P(y_j|x_1, x_2, x_3, ..., x_n) = \frac{P(x_1|y_j)P(x_2|y_j)...P(x_n|y_j)P(y_j)}{P(x_1)P(x_2)...P(x_n)}, \tag{2.4}$$

which can be expressed as:

$$P(y_j|x_1, x_2, x_3, ..., x_n) = \frac{P(y_j)\prod_{i=1}^{n} P(x_i|y_j)}{P(x_1)P(x_2)...P(x_n)}, \tag{2.5}$$

As the denominator remains constant for a given input, that term can be removed:

$$P(y_j|x_1, x_2, x_3, ..., x_n) \propto P(y_j)\prod_{i=1}^{n} P(x_i|y_j) \tag{2.6}$$

For the classification model it is needed to find the probability of given set of inputs for all possible values of the class variable y_j and pick up the output with maximum probability. A Bayes classifier is a function that assigns a class label $\hat{y} = y_j$ for some k and can be expressed as:

$$\hat{y} = \underset{\text{j}\in\{1,\ldots,\text{k}\}}{\operatorname{argmax}} P(y_j) \prod_{i=1}^{n} P(x_i|y_j) \tag{2.7}$$

2.4.2 Generalized Linear Model

Generalized lineal model (GLM) is a linear approach to modelling the relationship between a response and one or more explanatory variables. GLM is more flexible than ordinary linear regression, because it allows for response variable errors to be non-normally distributed. Generalized linear models were formulated by John Nelder and Robert Wedderburn in 1972. GLM has the structure :

$$g(\mu_i) = X_i\beta, \tag{2.8}$$

where $\mu_i \equiv E(Y_i), g$ is a smooth monotonic "link function", X_i is the i^{th} row of a data matrix X, and β is a vector of unknown parameters.[14]

The Generalized linear model consists of three elements:

1. A linear predictor $\eta = X\beta$. It is a quantity which incorporates the information about independent variables into the model.

2. A link function g, which describes how the mean of the process Y depends on the linear predictor : $\mathrm{E}(Y) = \mu = g^{-1}(\eta)$

3. A variance function V and dispersion parameter ϕ : $\mathrm{Var}(Y) = \phi\,\mathrm{V}(\mu) = \phi\,\mathrm{V}(g^{-1}(X\beta))$, which describes how variance of Y depends on the mean. However, in the most cases this property is assumed as probability distribution on dependent variable Y, which can be Normal, exponential, gamma, Poisson, Bernoulli, Binomial, categorical, multinomial and etc. [6]

The most common link function for both binary data and multinomial is *Logit* :

$$g(p) = \ln(\frac{p}{1-p}) \tag{2.9}$$

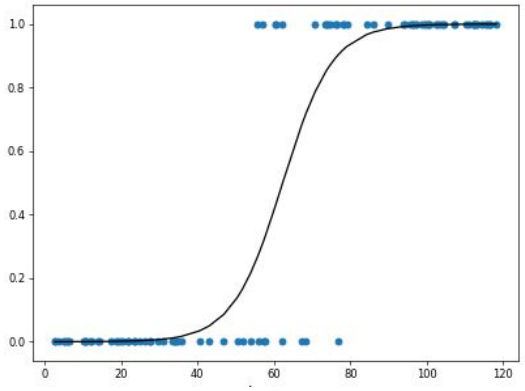

Figure 2.1: Logit link function

2.4.3 Gradient Boosting Model

Gradient boosting is a supervised machine learning technique, which was developed by Jerome H. Friedman.[2]. It creates a prediction model in the form of an ensemble of weak models, which are usually a Decision Trees.

A Decision Tree is a tree-based structure, which can be represented graphically as following:

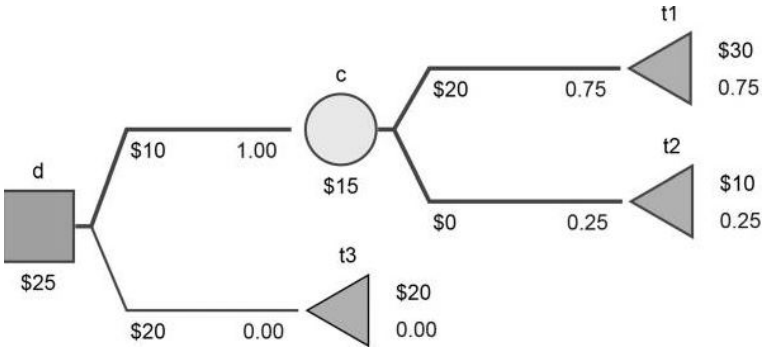

Figure 2.2: Decision Tree example

A Decision (or Root) nodes is represented by squares, a Chance nodes - by circles and a

Terminal (or Leaf) nodes - by triangles. "The paths from a Decision node to Leaf represent classification rules. In a decision node, the decision maker selects an action, i.e. one of the edges stemming from this node (one of the edges having the node in question as the parent). In a chance node, one of the edges stemming from it (a reaction) is selected randomly. Terminal nodes represent the end of a sequence of actions/reactions in the decision." [17]

The Gradient Boosting uses a Decision Tree ensembles, i.e. model consists of a set of classification and regression trees (CART), which is nothing else than Random Forests. The prediction scores of each tree are summed up to get the final score and prediction. The model takes form [12]:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F} \tag{2.10}$$

here K is the number of trees, f is a function in the functional space of all possible classification and regression trees $\mathcal{F}$, defined as $f(x) = w_{q(x)}, w \in R^T, q : R^d \to \{1, 2, ..., T\}$. Here w is the vector of scores, q is a function which assigns data points to separate leaves and T is the number of leaves. The objective function which should be optimized has a form :

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.11}$$

here $\Omega(f_k)$ is the complexity of the tree, defined as $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$. The difference between Random Forest model and Boosted Trees is that how they are trained. In the Boosted Trees the objective function to be optimized :

$$obj = \sum_{i}^{n} l(y_i, \hat{y_i}^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \tag{2.12}$$

here t is a training loss. The main idea of the model is to use an additive approach : fix what have been learned and add a new tree at a time. Then the prediction value at step t as $\hat{y_i}^{(t)}$ has the form :

$$\hat{y_i}^{(t)} = \sum_{i=1}^{t} f_k(x_i) = \hat{y_i}^{(t-1)} + f_t(x_i) \tag{2.13}$$

To define which tree is essential at each step it is needed to add the one that optimizes objective, consider using mean squared error (MSE) as a loss function. Taking *Taylor expansion of the loss function up to the second order* the specific objective at step t becomes:

$$\sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t), \tag{2.14}$$

here $g_i = \partial_{\hat{y_i}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$ and $h_i = \partial^2_{\hat{y_i}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$

After re-formulating the tree model, the objective value with the t-th tree can be expressed as :

$$obj^{(t)} \approx \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2}h_i w^2_{q(x_i)}] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 =$$
$$\sum_{j=1}^{T}[(\sum_{i\in I_j} g_j)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T$$

here $I_j = \{i|q(x_i) = j\}$ is the instance set of leaf j. In this equation w_j are independent with respect to each other. Compressing the expression by defining $G_j = \sum_{i\in I_j} g_j$ and $H_j = \sum_{i\in I_j} h_j$ we will get :

$$obj^{(t)} = \sum_{j=1}^{T}[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2] + \gamma T \tag{2.15}$$

As far as the form $G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2$ is quadratic and the best w_j for a given structure $q(x)$ is $w_j^* = -\frac{G_j}{H_j+\lambda}$. Therefore, the the best objective reduction, which measures quality of a tree structure $q(x)$ is :

$$obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \tag{2.16}$$

Assume that I_L and I_R are the scores on left and right leaves, then the loss reduction is given by [12] :

$$\alpha = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{(G_R+G_L)^2}{H_R+H_L+\lambda}\right] - \gamma \tag{2.17}$$

2.4.4 Support Vector Machines

Support Vector Machine is a supervised machine learning model used for both classification and regression tasks. The original SVM approach by Boser et al. (1992) was derived from the generalized portrait algorithm invented earlier by Vapnik and Lerner (1963). [5] The main goal of the Support Vector Machine algorithm to find a line in 2-dimensional space or hyperplane in N-dimensional space, where dimensions are defined by number of features of dataset. There are many possible hyperplanes that could be chosen to separate two classes of the data points. SVM objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes:

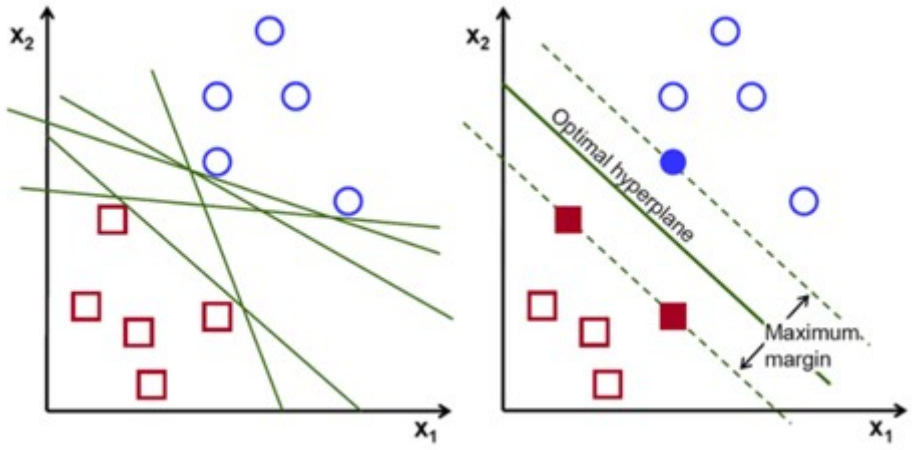

Figure 2.3: Support vectors and decision hyperplane

The kernel function is an implicit mapping returning the inner product $\langle \Phi(x), \Phi(x^{'}) \rangle$ between two data points $x, x^{'}$ in the n-dimensional feature space. This function computes all data points located in the feature space without memorizing the coordinates. Such function is called "kernel trick" and is used by SVM algorithm. Kernel trick reduces a computing performance, which is a big advantage for text classification problems, because it can work in spaces of any dimension without any significant additional computational cost, i.e. without memorizing the coordinates of data points. There are few types of kernel functions, such as linear, linear splines in one dimension, polynomial, Gaussian Radial Basis an other. [4]

Support vector machines separate different classes of data by a hyperplane [4]:

$$\langle w, \Phi(x) \rangle + b = 0 \tag{2.18}$$

corresponding to the decision function:

$$f(x) = \text{ sign}\langle (w, \Phi(x) \rangle + b), \tag{2.19}$$

here w is a weight vector, $\Phi(x)$ is a training set and b is a constant

Decision function corresponds to a positive and negative hyperplanes. They separates different classes of the data so that curves of the separated areas do not include any object from the training set. The distance should be maximized and optimization problem takes the form:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{2.20}$$

The hyperplane can be found by the equation:

$$\sum_{i=1}^{m} \alpha_i y^{(i)} \mathcal{K}(x^{(i)}, x) + b, \tag{2.21}$$

here α_i and b are parameters of the optimal hyperplane, $y_i = \pm 1$ and $\mathcal{K}$ is a kernel function. [4]

Originally SVM can only solve binary classification problems. However, there is a possibility to allow a multi-class classification. For multi-class problem should be used the one-against-one and one-against-all technique by fitting all binary subclassifiers and finding the correct class by a voting mechanism.

In the one-against-all method k binary SVM classifiers are trained, where k is the number of classes. Each of k SVM are trained to distinguish one class from the rest. Then all SVM classifiers are compared by the highest decision value for each class.

In the one-against-one classification method, which is also known as pairwise classification, $\binom{k}{2}$ classifiers are trained on data from two classes. The prediction of the class is conducted by voting. Each SVM prediction is counted and wins the most frequent class. This method suggests a higher number of support vector machines to train the overall CPU time used is less compared to the one-against-all method since the problems are smaller and the SVM optimization problem scales super-linearly.[4]

2.5 Evaluation metrics

It is very important to evaluate performance of the model, since model can underfit, or conversely, overfit the data. To avoid this, it is needed to evaluate the model, on both training and test subsets.

Before describing main evaluation metrics, let give an example of four basic measures:

1. **True-positive (TP):** means instance were correctly classified, i.e. actual label was "1" and predicted "1"

2. **True-negatives (TN):** means that class of instance was correctly rejected, i.e. instance with actual label "2" was not classified as "1"

3. **False-positives (FP):** refers to instance falsely classified to a given label, i.e. instance with actual label "2" classified as "1"

4. **False-negatives (FN):** refers to instance falsely rejected from given label, i.e. instance with actual label "1" not classified as "1"

The following table, known as confusion matrix, represents basic measures for binary response:

Predicted value \ Actual value	1	2
1	TP	FP
2	FN	TN

Table 2.3: Confusion matrix for binary response

Confusion matrix for multi-class response:

*	$c_1, .., c_{k-1}$	c_k	$c_{k+1}, .., c_n$
$c_{k+1}, .., c_n$	TN	FP	TN
c_k	FN	TP	FN
$c_1, .., c_{k-1}$	TN	FP	TN

Table 2.4: Confusion matrix for multi-class response

Accuracy

Accuracy shows the number of correct predictions made to the total number of instances. Which can be expressed as :

$$\frac{\sum^{k} TP_k}{N} \tag{2.22}$$

where k is the total number of classes, TP_k is the number of TP for class k and N is the total number of instances.

However, this evaluation metric should be used very carefully, because it suitable only for balanced data sets. In case of imbalanced data set it can falsely show high accuracy on a very weak model. For instance :

Actual value / Predicted value	1	2
1	700 (TP)	80 (FP)
2	200 (FN)	20 (TN)

Table 2.5: Confusion matrix example

Accuracy for such data is 72%, which can say that model is quite good, but if we will look at actual value "2" - the only 20% of it predicted accurately.

Recall

Recall is also known as sensitivity, or TPR (True Positive Rate). Recall shows the number of correctly classified instances to a number of instances for each class separately. It can be expressed as following:

$$TPR = \frac{TP}{TP + FN} \tag{2.23}$$

Precision

Precision, or Positive Predictive value (PPV), shows the proportion of positive identifications that was actually correct. As well as Recall, this measure is calculated for each class separately and expressed as :

$$PPV = \frac{TP}{TP + FP} \tag{2.24}$$

F-score

There can be some problem where higher Recall take precedence over a higher Precision and conversely. For some cases it is useful to use both Precision and Recall at the same time, i.e. calculate its weighted average:

$$F_1 = \left(\frac{2}{recall^{-1} + precision^{-1}} \right) = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.25}$$

Receiver Characteristic Operator

The area under the ROC (Receiver Characteristic Operator) curve, or the equivalent Gini index, is a widely used measure of performance of supervised classification rules. [3] ROC is a probability curve that plots the $TPR = \frac{TP}{TP+FN}$ against $FPR = \frac{FP}{FP+TN}$ at different classification thresholds and separates the "signal" from the "noise". In a multi-class models ROC is calculated for each class separately, i.e. one class vs all the other. The area under the ROC curve (AUC) is most common measure for classification model assessment which shows the ability of classifier to distinguish between classes. The higher AUC value to 1l the better performance of the model.

AUC and ROC is shown in the graph below:

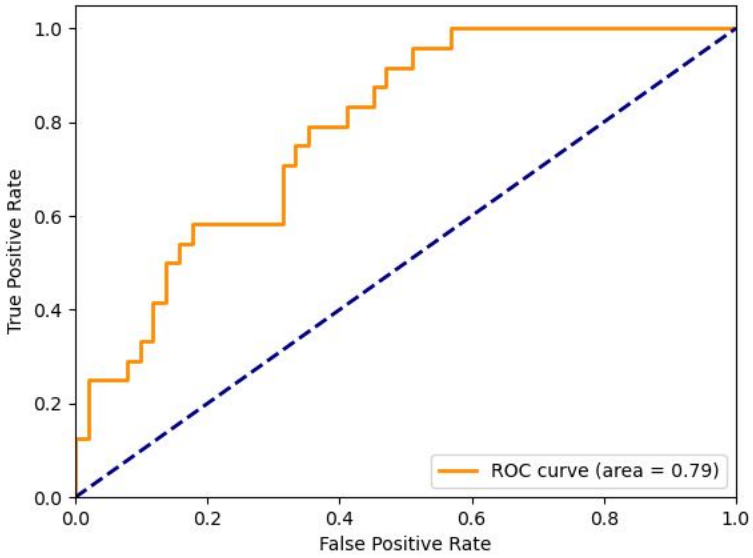

Figure 2.4: ROC example

Chapter 3

Related works

Text mining became very popular in past 15 years and is still ongoing topic of research. It is a process of extraction valuable information from unstructured data and transforming it to structured one, that can be further analyzed by different methodologies like Machine Learning. In the related work [11] authors are analysing multilingual Twitter text in a context of sentiment analysis, the goal of sentiment analysis is to extract opinions, emotions, or attitudes towards different objects of interest. Authors constructed and evaluated six different classification models for each labelled language dataset. Comparing the classifiers' turns out that there is no statistically significant difference between most of classifiers. However, authors selected SVM classifier based on most relevant evaluation measures.

Another work based on sentiment analysis [8]. In this paper Authors propose a new method for feature selection which is based on the probabilistic topic model. The proposed approach uses a structured vector of features, composed of weighted pairs of words. The proposed vector of features is automatically learned, given a set of documents. The terms are extracted based on the Latent Dirichlet Allocation implemented as the Probabilistic Topic Model. Based on F-score, the performance of such approach is on average higher than of SVM.

In the paper [16] Authors are presenting a hybrid approach of rule-based and machine learning classifier XGBoost, applying it on labelled outgoing transaction data, which has two

subsets – wire transfers and card payments. Hybrid approach improved the coverage of the first subset by 11% and second subset by 0.6%, compared to fully rule-based approach.

Several works [9], [10] are the most related to this thesis. The Authors analyzed outgoing labelled transactions, classifying them with Neural Networks, SVM, Decision Trees and Naïve Bayes algorithms. The results show that transaction data is suitable for machine learning. Support Vector Machines and Decision Tree models gave the best results.

One more advanced work on transaction classification [13]. Authors are using outgoing transactions for classification. The data contains only payment description and some supplementary information obtained from external resources. Results show that using external semantic resources to supplement the classification model provides a significant improvement to the overall accuracy of the system.

Chapter 4

Implementation

4.1 Technical notes

The experiment is completed on laptop (Intel core i5, 4 cores, 16 GB RAM), since it is not a Big Data related experiment, but the main goal of the thesis is to create possible solution for classification of transactions. For data preprocessing and modelling we chose the statistical software R&RStudio. R has a wide choose of libraries for Natural Language Processing tasks and building a machine learning models.

4.2 Data

The data is extracted to csv files from the internal database of the bank. The data set contains 69980 incoming transactions which are randomly selected within the time interval from 2020 January to 2020 June, 29019 transactions are labelled and 40961 unlabelled. For the modelling part, data set was divided into training 70% and test 30% subset.

There are 9 transaction classes which are labelled by bank, for instance, "salary", "social security payment", "dividends", "pension" and other. In the unlabelled subset some transaction

descriptions do not contain any values, as customers have option to leave it blank.

Available data fields are following:

- Customer name
- Contraparty Individual organization code (private person or legal entity)
- Contraparty name
- Payment text description (unstructured free-form text payment description, maximum length is 140 characters)
- Payment amount in Eur
- Payment activity Type Code (technical attribute, for instance, "SEPA payment", "payment between bank customers", etc.)
- Currency
- Same contraparty indicator
- Same surname (indicates if customer and contraparty has the same surname)
- IN between 1 year (indicates if number of incoming transactions from contraparty is more than 1)
- OUT between 1 year (indicates if number of outgoing transactions to contraparty is more than 0)

Data example

In the following table is presented artificial data example with the main fields:

Customer name	Jonas Jonaitis	Petras Petraitis	Ona Petraitytė	Jonas Jonaitis	Ona Petraitytė
Transaction description	sūnui	atlyginimas už 2020.01	stipendija	hi	pervedimas
Amount in Eur	10	1500	58.5	4.5	50
Original currency	EUR	EUR	EUR	USD	EUR
Payment activity Type	10	3	19	4	10
Same contraparty ind	N	N	N	N	Y
Same surname	Y	N	N	N	Y
Contraparty Name	Jona Jonaitienė	UAB Darbas	Vilniaus Universitetas	Paulius Paulauskas	Ona Petraitytė
Contraparty Individual organization code	2	1	1	2	2

Table 4.1: Data example

4.3 Experiment plan

The experiment were divided into 4 main steps:

Step1. The manual labelling of unlabelled data subset

The first step includes manual labelling of unlabelled data subset, which is the most

time consuming part of data mining.

Step2. Data preparation for the modelling

The transaction description will be processed based on Natural Language processing procedures and the rest variables will be one-hot encoded or standardized.

Step3. Fitting the ML models

For classification part were selected Naïve Bayes, Support Vector Machines, Generalized Linear and Gradient Boosting models.

Step4. Model assessment and result comparison

The last step includes comparison of the models based on the common evaluation metrics for classification tasks.

Chapter 5

Experiment results

5.1 Labelling the data

The most time consuming step of data mining is the manual data labelling. The subset of unlabelled transactions with 40961 instances were labelled manually. Based on transaction description and other variables there were obtained 21 new possible label.

The following table shows some transaction examples and their class, obtained manually:

Transaction text	Same surname	Contr Ind code	In between 1 year	Out between 1 year	Label
:]	Y	2	1	1	between family members
pervedimas	N	1	0	0	incoming cash flow from legal entity
ačiū	N	2	0	0	incoming cash flow from private - unknown
mokėjimas	N	2	1	1	incoming cash flow from private - known
pervedimas pagal sąskaitą-faktūrą nr1	N	1	0	0	Transfer by invoice
už konsultaciją	N	2	1	0	Services
už pristatyta pieną	N	1	1	0	Farmery
skola	Y	2	1	1	Debts
ūkinems išlaidoms	N	1	1	0	Income for household expenses
už būto nuomą	N	2	1	0	Rent / Accommodation
vykdomoji byla nr2	N	1	1	0	Payment by executive case

Table 5.1: Labelled data example

The next table presents all labels, the number of instances and the most frequent words in transaction description in each class :

Label Nbr	Label name	Nbr of instances	The most frequent words
0	Transfer between family members	9092	papildymas, sūnui, pavedimas, žmonai, pinigų
1	Refunds / orders	2509	užsakymas, grąžinimas, permokos, refund, pagal
2	Bet money withdrawal	317	išvedimas, pinigų, withdrawal, money, laimėjimai
3	Farmery	432	pienas, pristatytas, žaliava, pusė, susietoji
4	Damage compensation from insurance	45	Ind, atlyginimas, tuk, kdi, priemoka
5	Transfer by invoice	956	atsiskaitymas, pagal, išmoka, sąskaita, faktūra
6	Payments to soldiers	65	šauktiniams, mėn, buitinė, kareiviams, karo
7	Payments from credit institutions	82	didinimas, vartojimo, kreditas, sumos, numeris
8	Goods / clothes	886	prekes, vinted, suknelė, batai, džinsai
9	Payment by executive case	124	vbnr, byla, vykdomoji, skolininkas, vb
10	Rent / Accommodation	621	būtas, nuoma, auto, patalpų, mėn
11	Services	327	paslauga, konsultacija, suteiktas, reklamos, sutartis
12	Received gift	901	dovana, dovanelė, gimtadienio, dovanoju, myliu
13	Loans	174	paskola, bobutės, dengti, eur, busto
14	Debts	2979	skola, grąžinimas, skolinu, dolg, skolinimas
15	Income from taxi activities	101	food, balance, pid, įeinantis, payout
16	Payment from tax institution	909	gyventojų, pajamų, lietuvos, mokamas, nuolatinio
17	Pocket money	39	kišenpinigai, mėn, kisianpinigai, išmokėjimo, pagal
18	Incoming cash flow from legal entity	3666	tmp, įeinantis, pavedimas, pervedimas, papildymas
19	Income for household expenses	65	reikmėms, išlaidoms, ūkinėms, avansas, reikalams
20	Incoming cash flow from private - unknown	6591	papildymas, pavedimas, pervedimas, mobiliąją, programėlę
21	Incoming cash flow from private - known	7387	papildymas, ačiū, pinigų, mokėjimas, transfer
22	Salary payment	12021	atlyginimas, avansas, darbo, užmokestis, alga
23	Pension payment	147	pensija, įeinantis, for, pension, išmokėjimas
24	Family support payment	2712	išmoka, vaikams, vaikui, administracija, savivaldybė
25	Social security payment	12963	sodros, nurodymas, žiniaraščio, tikslinė, išmoka
26	Dividend payment	101	dividendai, užsienyje, eur, išskaiciuota, eurvnt
27	Grant payment	396	stipendija, sąskaita, atsiskaitymai, studentai, mėn
28	Alimony payment	552	alimentai, vaikui, elementai, išlaikymui, pinigai
29	Transfer between customer accounts	2820	pervedimas, mobiliąją, savo, sąskaitų, papildymas

Table 5.2: List of all labels

5.2 Data preparation

All the needed background for the data preparation is presented in the section 2.1. Some variables were omitted because of their statistical insignificance. The significance of the variables were identified by experimental approach - trying different combinations of variables and comparing the evaluation metrics of algorithms. So far, for the machine learning model were used : Payment text description, Contraparty Individual organization code, Currency, Payment activity Type Code, Same contraparty indicator, Same surname, IN between 1 year, OUT between 1 year. To prepare these variables for the ML models, the following steps of data preparation were performed:

- Transaction text were processed with the *tm* [1] package in R , which provides all needed functions for Natural Language Processing. The transaction text was lowered, extra spaces, punctuation and numbers were removed as it is mentioned in the section 2.1. Also we performed stemming for Lithuanian language and all the special Lithuanian language characters were substituted by regular : ą - a, ė - e, ų - u and etc. An example of the processed text is shown in the following table:

Raw text	Processed
Atlyginimas už 2020.01	atlyginim uz

Table 5.3: Processed data example

  In the next step text was represented using Bag of Words, as it is described in section 2.1.2. Since transaction texts are very short and contains the only few keywords, such simple approach as Bag of Words is enough. After all manipulations with the text there were left 16461 columns with distinct words.

- The variable "Amount in Eur" were standardized as it is described in section 2.3

- All the other categorical features as Contraparty Individual organization code, Cur-

[1] Online; accessed 4 January 2021 https://cran.r-project.org/web/packages/tm/tm.pdf

rency, Payment activity Type Code, Same contraparty identificator, Same surname, IN between 1 year, OUT between 1 year were One-hot encoded, as it is described in section 2.2.

5.3 Classification

Since all the labels are in place we can create an automated solution for assigning labels for a new transactions. For that purpose it is needed to train supervised machine learning models. While running an experiment there were selected four models based on the evaluation metrics. This section will show results of Naïve Bayes, Generalized Linear Models, Gradient Boosting and Support Vector Machines models.

5.3.1 Naïve Bayes Model

For classification with Naïve Bayes model were used a *naivebayes* package in R [2].

The main averaged evaluation metrics for test and training subsets are presented in the following table:

Evaluation metric	Training subset	Test subset
Training time	0.734 sec	-
AUC	96.73	92.92
Accuracy	95.44	88.74
Recall	95.02	88.75
Precision	95.48	86.79
F-score	95.25	87.76

Table 5.4: Naïve Bayes model results

The most significant words for each class are presented in the Appendix. The basic statistics

[2]Online; accessed 4 January 2021 https://cran.r-project.org/web/packages/naivebayes/naivebayes.pdf

for each class separately is presented in the next table:

Label Nbr	Training Recall	Training Precision	Training F-score	Training Accurately predicted instances	Test Recall	Test Precision	Test F-score	Test Accurately predicted instances
0	97.35	91.72	94.45	6168 / 6336	91.36	84.81	87.97	2518 / 2756
1	99.05	95.15	97.06	1766 / 1783	95.87	89.23	92.43	696 / 726
2	100.00	97.37	98.67	222 / 222	100.00	95.00	97.44	95 / 95
3	100.00	98.72	99.36	309 / 309	100.00	95.35	97.62	123 / 123
4	90.91	100.00	95.24	30 / 33	91.67	100.00	95.65	11 / 12
5	90.10	82.31	86.03	619 / 687	74.35	58.82	65.68	200 / 269
6	81.63	100.00	89.89	40 / 49	100.00	94.12	96.97	16 / 16
7	98.15	100.00	99.07	53 / 54	96.43	96.43	96.43	27 / 28
8	98.53	89.45	93.77	602 / 611	91.64	79.75	85.28	252 / 275
9	93.98	93.98	93.98	78 / 83	46.34	61.29	52.78	19 / 41
10	98.39	85.92	91.73	427 / 434	91.98	71.37	80.37	172 / 187
11	95.09	89.12	92.01	213 / 224	79.61	75.23	77.36	82 / 103
12	99.36	96.86	98.09	617 / 621	94.64	82.56	88.19	265 / 280
13	97.79	93.66	95.68	133 / 136	89.47	82.93	86.08	34 / 38
14	99.61	97.32	98.45	2067 / 2075	95.24	93.28	94.25	861 / 904
15	100.00	98.72	99.35	77 / 77	100.00	88.89	94.12	24 / 24
16	100.00	99.24	99.62	651 / 651	100.00	98.85	99.42	258 / 258
17	90.48	100.00	95.00	19 / 21	77.78	100.00	87.50	14 / 18
18	76.77	94.49	84.71	1953 / 2544	57.40	73.02	64.27	644 / 1122
19	80.00	96.97	87.67	32 / 40	72.00	90.00	80.00	18 / 25
20	89.93	98.21	93.89	4171 / 4638	76.50	89.62	82.54	1494 / 1953
21	86.78	92.87	89.72	4478 / 5160	69.78	82.05	75.42	1554 / 2227
22	98.71	94.51	96.56	8310 / 8419	94.73	91.28	92.97	3412 / 3602
23	100.00	97.32	98.64	109 / 109	94.74	92.31	93.51	36 / 38
24	99.68	98.28	98.97	1882 / 1888	99.39	97.27	98.32	819 / 824
25	99.68	99.17	99.42	9055 / 9084	99.05	95.57	97.28	3842 / 3879
26	95.65	95.65	95.65	66 / 69	100.00	88.89	94.12	32 / 32
27	96.81	96.47	96.64	273 / 282	96.49	91.67	94.02	110 / 114
28	96.21	90.93	93.50	381 / 396	85.90	64.11	73.42	134 / 156
29	99.95	100.00	99.97	1951 / 1952	100.00	100.00	100.00	868 / 868

Table 5.5: Naïve Bayes model detailed results

5.3.2 Generalized Linear Model

The second fitted model is GLM. For this we used R package *glmnet* [3], which fits a generalized linear model via penalized maximum likelihood. There were conducted cross-validation for a multinomial model with 15 number of folds and alpha = 0, which means that model was fitted with ridge penalty. Ridge regression is known to shrink the coefficients of correlated predictors towards each other, allowing them to borrow strength from each other. [7]

The main averaged evaluation metrics for test and training subsets are presented in the following table:

Evaluation metric	Training subset	Test subset
Training time	24.6 min	-
AUC	81.94	82.6
Accuracy	93.78	88.86
Recall	69.53	63.84
Precision	94.83	89.09
F-score	80.23	74.38

Table 5.6: Generalized Linear Model results

The most significant words for each class are presented in the Appendix. The basic statistics for each class separately is presented in the next table:

[3] Online; accessed 4 January 2021 https://cran.r-project.org/web/packages/glmnet/glmnet.pdf

Label Nbr	Training Recall	Training Precision	Training F-score	Training Accurately predicted instances	Test Recall	Test Precision	Test F-score	Test Accurately predicted instances
0	99.57	92.21	95.75	6309 / 6336	97.53	89.75	93.48	2688 / 2756
1	93.61	99.64	96.53	1669 / 1783	86.92	95.46	90.99	631 / 726
2	63.96	99.30	77.81	142 / 222	100.00	98.96	99.48	95 / 95
3	81.88	100.00	90.04	253 / 309	90.24	99.11	94.47	111 / 123
4	39.39	100.00	56.52	13 / 33	41.67	100.00	58.82	5 / 12
5	63.76	97.55	77.11	438 / 687	47.21	83.55	60.33	127 / 269
6	55.10	100.00	71.05	27 / 49	56.25	90.00	69.23	9 / 16
7	83.33	100.00	90.91	45 / 54	89.29	100.00	94.34	25 / 28
8	60.56	99.73	75.36	370 / 611	45.82	95.45	61.92	126 / 275
9	61.45	100.00	76.12	51 / 83	4.88	66.67	9.09	2 / 41
10	59.22	100.00	74.38	257 / 434	47.06	91.67	62.19	88 / 187
11	37.50	98.82	54.37	84 / 224	13.59	70.00	22.76	14 / 103
12	49.11	100.00	65.87	305 / 621	42.14	96.72	58.71	118 / 280
13	22.79	100.00	37.13	31 / 136	7.89	60.00	13.95	3 / 38
14	97.78	99.66	98.71	2029 / 2075	93.47	98.14	95.75	845 / 904
15	100.00	98.72	99.35	77 / 77	100.00	92.31	96.00	24 / 24
16	100.00	99.69	99.85	651 / 651	100.00	100.00	100.00	258 / 258
17	0.00	0.00	0.00	0 / 21	5.56	100.00	10.53	1 / 18
18	74.25	97.67	84.37	1889 / 2544	49.38	80.76	61.28	554 / 1122
19	17.50	100.00	29.79	7 / 40	8.00	50.00	13.79	2 / 25
20	99.89	86.87	92.93	4633 / 4638	93.75	78.48	85.44	1831 / 1953
21	96.67	91.81	94.18	4988 / 5160	86.66	85.02	85.84	1930 / 2227
22	99.05	87.97	93.18	8339 / 8419	96.47	83.65	89.61	3475 / 3602
23	51.38	100.00	67.88	56 / 109	39.47	93.75	55.56	15 / 38
24	98.62	97.64	98.13	1862 / 1888	98.30	96.66	97.47	810 / 824
25	99.67	98.84	99.25	9054 / 9084	99.20	94.22	96.65	3848 / 3879
26	73.91	100.00	85.00	51 / 69	75.00	96.00	84.21	24 / 32
27	53.55	98.69	69.43	151 / 282	67.54	98.72	80.21	77 / 114
28	52.53	100.00	68.87	208 / 396	32.05	87.72	46.95	50 / 156
29	100.00	100.00	100.00	1952 / 1952	100.00	100.00	100.00	868 / 868

Table 5.7: Generalized Linear Model detailed results

5.3.3 Gradient Boosting Model

The next fitted model is Gradient Boosting Model. For this purpose we used R package *xgboost* [4]. XGBboost is short for eXtreme Gradient Boosting. XGBoost has gained much popularity and attention as the best algorithm in a machine learning competitions.[5] The model was trained with 20 iterations and the softmax optimization.

The main averaged evaluation metrics for test and training subsets are presented in the following table:

Evaluation metric	Training subset	Test subset
Training time	1.11 hours	-
AUC	95.42	94.07
Accuracy	96.94	96.07
Recall	92.44	89.09
Precision	97.47	93.17
F-score	94.89	91.08

Table 5.8: XGBoost results

The most significant words for each class are presented in the Appendix. The basic statistics for each class separately is presented in the next table:

[4]Online; accessed 4 January 2021 https://cran.r-project.org/web/packages/xgboost/xgboost.pdf

[5]Online; accessed 4 January 2021 https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions

Label Nbr	Training Recall	Training Precision	Training F-score	Training Accurately predicted instances	Test Recall	Test Precision	Test F-score	Test Accurately predicted instances
0	99.24	98.42	98.83	6288 / 6336	99.35	97.89	98.61	2738 / 2756
1	97.03	99.37	98.18	1730 / 1783	95.04	99.28	97.12	690 / 726
2	100.00	99.55	99.78	222 / 222	100.00	98.96	99.48	95 / 95
3	99.35	99.35	99.35	307 / 309	98.37	69.54	81.48	121 / 123
4	96.97	100.00	98.46	32 / 33	100.00	100.00	100.00	12 / 12
5	77.88	89.62	83.33	535 / 687	71.38	87.27	78.53	192 / 269
6	97.96	100.00	98.97	48 / 49	87.50	100.00	93.33	14 / 16
7	87.04	100.00	93.07	47 / 54	67.86	100.00	80.85	19 / 28
8	75.45	98.29	85.37	461 / 611	75.27	98.10	85.19	207 / 275
9	40.96	94.44	57.14	34 / 83	21.95	52.94	31.03	9 / 41
10	91.01	98.75	94.72	395 / 434	88.24	98.21	92.96	165 / 187
11	93.75	92.51	93.13	210 / 224	84.47	82.86	83.65	87 / 103
12	92.11	99.31	95.57	572 / 621	90.71	98.83	94.60	254 / 280
13	93.38	97.69	95.49	127 / 136	78.95	93.75	85.71	30 / 38
14	96.77	98.72	97.74	2008 / 2075	94.03	97.81	95.88	850 / 904
15	100.00	100.00	100.00	77 / 77	100.00	96.00	97.96	24 / 24
16	100.00	100.00	100.00	651 / 651	100.00	100.00	100.00	258 / 258
17	90.48	100.00	95.00	19 / 21	83.33	88.24	85.71	15 / 18
18	91.04	79.67	84.98	2316 / 2544	91.36	78.36	84.36	1025 / 1122
19	80.00	94.12	86.49	32 / 40	76.00	79.17	77.55	19 / 25
20	99.85	97.15	98.48	4631 / 4638	99.33	96.09	97.68	1940 / 1953
21	99.61	95.17	97.34	5140 / 5160	99.19	94.73	96.91	2209 / 2227
22	94.17	97.26	95.69	7928 / 8419	94.09	97.05	95.55	3389 / 3602
23	94.50	97.17	95.81	103 / 109	100.00	97.44	98.70	38 / 38
24	99.42	99.95	99.68	1877 / 1888	98.91	99.88	99.39	815 / 824
25	99.55	99.64	99.59	9043 / 9084	98.22	99.58	98.90	3810 / 3879
26	97.10	100.00	98.53	67 / 69	96.88	100.00	98.41	31 / 32
27	97.16	99.64	98.38	274 / 282	98.25	98.25	98.25	112 / 114
28	91.41	98.37	94.76	362 / 396	83.97	94.93	89.12	131 / 156
29	100.000	100.000	100.000	1952 / 1952	100.000	100.000	100.000	868 / 868

Table 5.9: XGBoost detailed results

5.3.4 Support Vector Machines

The last fitted model for classification is SVM. For this purpose we used R package *e1071* [6]. SVM model was fitted with a linear kernel.

The main averaged evaluation metrics for test and training subsets are presented in the following table:

Evaluation metric	Training subset	Test subset
Training time	1.56 min	-
AUC	99.61	95.52
Accuracy	99.27	96.68
Recall	99.34	93.09
Precision	99.40	93.68
F-score	99.37	93.39

Table 5.10: SVM results

The basic statistics for each class separately is presented in the next table:

[6]Online; accessed 4 January 2021 https://cran.r-project.org/web/packages/e1071/e1071.pdf

Label Nbr	Training Recall	Training Precision	Training F-score	Training Accurately predicted instances	Test Recall	Test Precision	Test F-score	Test Accurately predicted instances
0	99.98	99.92	99.95	6335 / 6336	99.27	98.63	98.95	2736 / 2756
1	99.66	99.89	99.78	1777 / 1783	95.32	97.19	96.25	692 / 726
2	100.00	100.00	100.00	222 / 222	100.00	100.00	100.00	95 / 95
3	100.00	100.00	100.00	309 / 309	99.19	99.19	99.19	122 / 123
4	100.00	100.00	100.00	33 / 33	100.00	100.00	100.00	12 / 12
5	94.32	98.93	96.57	648 / 687	71.75	79.75	75.54	193 / 269
6	100.00	100.00	100.00	49 / 49	100.00	88.89	94.12	16 / 16
7	100.00	100.00	100.00	54 / 54	92.86	96.30	94.55	26 / 28
8	97.05	98.18	97.61	593 / 611	90.55	95.77	93.08	249 / 275
9	98.80	100.00	99.39	82 / 83	46.34	59.38	52.05	19 / 41
10	100.00	99.31	99.66	434 / 434	90.37	97.13	93.63	169 / 187
11	99.11	96.94	98.01	222 / 224	80.58	83.84	82.18	83 / 103
12	100.00	99.84	99.92	621 / 621	95.71	99.63	97.63	268 / 280
13	99.26	99.26	99.26	135 / 136	89.47	97.14	93.15	34 / 38
14	99.86	99.81	99.83	2072 / 2075	95.47	98.29	96.86	863 / 904
15	100.00	100.00	100.00	77 / 77	100.00	96.00	97.96	24 / 24
16	100.00	100.00	100.00	651 / 651	100.00	100.00	100.00	258 / 258
17	100.00	100.00	100.00	21 / 21	83.33	88.24	85.71	15 / 18
18	94.38	94.79	94.58	2401 / 2544	84.49	84.72	84.61	948 / 1122
19	100.00	97.56	98.77	40 / 40	96.00	70.59	81.36	24 / 25
20	99.87	99.59	99.73	4632 / 4638	98.21	97.86	98.03	1918 / 1953
21	99.71	99.79	99.75	5145 / 5160	98.52	96.19	97.34	2194 / 2227
22	98.66	98.30	98.48	8306 / 8419	96.11	95.11	95.61	3462 / 3602
23	100.00	100.00	100.00	109 / 109	100.00	100.00	100.00	38 / 38
24	99.68	100.00	99.84	1882 / 1888	99.27	99.88	99.57	818 / 824
25	99.97	99.90	99.93	9081 / 9084	99.64	99.51	99.57	3865 / 3879
26	100.00	100.00	100.00	69 / 69	100.00	100.00	100.00	32 / 32
27	100.00	100.00	100.00	282 / 282	100.00	99.13	99.56	114 / 114
28	100.00	100.00	100.00	396 / 396	90.38	92.16	91.26	141 / 156
29	100.00	100.00	100.00	1952 / 1952	100.00	100.00	100.00	868 / 868

Table 5.11: SVM detailed results

5.4 Result comparison

There were trained four different classification algorithms on incoming transaction data. The following graph shows main evaluation metrics on the test data set for Naïve Bayes, Generalized Linear Models, Gradient Boosting and Support Vector Machines algorithms:

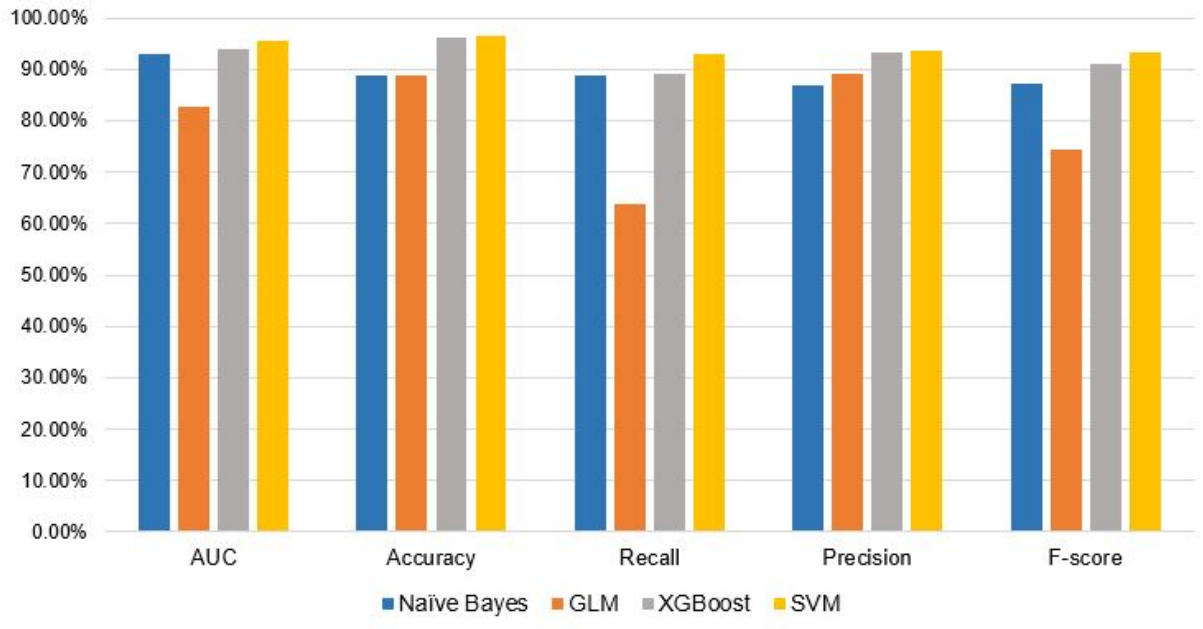

Figure 5.1: Model comparison

All the metrics are averaged over all classes. From the results it is seen that SVM model, which has $AUC = 95.52$, $Accuracy = 96.68$, $Recall = 93.09$, $Precision = 95.68$ and $F - score = 93.39$, has achieved the best results. It is not a surprise, as Support Vector Machines in the literature referred as the state of the art in the classification of the textual and sparse data. The next two models, that have achieved a little bit worse results than SVM, are Gradient Boosting Model and Naïve Bayes. The XGBoost has obtained $AUC = 94.07$, $Accuracy = 96.07$, $Recall = 89.09$, $Precision = 93.17$, $F - score = 91.08$ and the Naïve Bayes $AUC = 92.92$, $Accuracy = 88.74$, $Recall = 88.75$, $Precision = 86.79$, $F - score = 87.76$ respectively. The Generalized Linear Model achieved the poorest results : $AUC = 82.6$, $Accuracy = 88.86$, $Recall = 63.84$, $Precision = 89.09$ and $F - score = 74.38$.

Chapter 6

Conclusion

6.1 Summary

1. If there are any other payment classes than already defined by the bank?

   There were manually obtained 21 new class of incoming payments. This proves that there can be gained more information from incoming transactions. The new obtained incoming payment classes can be used in a different business areas in the bank. For instance, in an anti-money laundering area, classified customer incomes would facilitate the analysis of a customer money flow.

2. If such limited data as the incoming payments, is suitable for the classification?

   The conducted experiments prove that such type of data, as the incoming payments, is suitable for classification with a help of the machine learning algorithms. The payment text description should be preprocessed with a Natural Language Processing techniques and along with the other features of the transaction can be used as input for different classification algorithms.

3. How do different classification algorithms, such as Naïve Bayes, Support Vector Ma-

chines, Generalized Linear Models and Gradient Boosting, compare in a payment classification?

The all four trained algorithms have shown decent results. However, with the Support Vector Machines algorithm there were obtained the most accurate predictions of the payment labels. The achieved evaluation metrics of test data with SVM are : $AUC = 95.52$, $Accuracy = 96.68$, $Recall = 93.09$, $Precision = 95.68$ and $F - score = 93.39$. The weakest among four fitted models is the Generalized Linear model, which gained $AUC = 82.6$, $Accuracy = 88.86$, $Recall = 63.84$, $Precision = 89.09$ and $F - score = 74.38$.

4. If the rule-based classification approach can be substituted by the machine learning model?

   The main evaluation metrics, such as AUC and $Accuracy$ of the best algorithm are higher than 95% and $F - score$ is higher than 93%, which is quite good results for classification model. Taking that into consideration, we assume that the rule-based approach can be substituted by the SVM machine learning algorithm.

6.2 Future work

There are few possible directions for the transaction classification improvement. Since transaction description is a free-form field for user, it was noticed that quite few transaction descriptions contain grammatical mistakes and mistypes. Automatic correction of such mistakes would reduce noise in the data.

The transactions contain different languages, which also increases noise in the data. Translation of transaction description may have a positive impact reducing such noise.

Almost in the all related works based on short text classification, text were represented as Bag-of-Words. It would be interesting to test a hypothesis if representing text as *n-grams* or *syntactic parsing* would improve classification algorithm performance comparing to Bag-

of-Words.

References

[1] McEnery A. and Wilson A. *Corpus Linguistics, 2nd ed.* Edinburgh University Press, 2001, pp. 68–96.

[2] Friedman J. H. "Greedy function approximation: A gradient boosting machine". In: *Annals of Statistics, vol 29* (2001), pp. 1189–1232.

[3] Hand D. J. and Till R.J. "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning, vol. 45* (2001), pp. 71–186.

[4] Karatzoglou A. and Meyer D. "Support Vector Machines in R". In: *Journal of Statistical Software, vol. 15* (2006), pp. 1–28.

[5] Steinwart I. and Christmann A. *Support Vector Machines.* Springer, 2008, pp. 13–19.

[6] Madsen H. and Thyregod P. *Introduction to General and Generalized Linear Models.* CRC Press, 2010, pp. 6–8.

[7] Friedman J., Hastie T., and Tibshirani R. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software, vol. 33* (2010), pp. 1–20.

[8] Colace F. et al. "Text classification using a few labeled examples". In: *Computers in Human Behavior* (2013), pp. 689–697.

[9] Skeppe L. B. *Classify Swedish bank transactions with early and late fusion techniques.* KTH Royal Institute of Technology, 2014.

[10] Bengtsson H. and Jansson J. *Using Classification Algorithms for Smart Suggestions in Accounting Systems*. Chalmers University of Technology, 2015.

[11] Mozetič I., Grčar M., and Smailović J. "Multilingual Twitter Sentiment Classification: The Role of Human Annotators". In: *Plos One Journal* (2016).

[12] Chen T. and Guestrin C. "XGBoost: A Scalable Tree Boosting System". In: *22nd ACM SIGKDD International Conference* (2016), pp. 1–10.

[13] Folkestad O. E. E. and Vollset E. E. N. *Automatic Classification of Bank Transactions*. Norwegian University of Science and Technology, 2017.

[14] Wood S. N. *Generalized Additive Models: An Introduction with R*. Chapman and Hall / CRC, 2017, pp. 101–102.

[15] Kwartler T. *Text Mining in Practice with R*. USA: John Wiley & Sons, 2017, pp. 20–22.

[16] Mateush A. et al. "Building Payment Classification Models From Rules and Crowdsourced Labels: A Case Study". In: *Advanced Information Systems Engineering Workshops* (2018), pp. 85–97.

[17] Kaminski B., Jakubczyk M., and Szufel P. "A framework for sensitivity analysis of decision trees". In: *Central European Journal of Operations Research, vol. 26* (2018), pp. 135–159.

Appendix A

Appendix

Naïve Bayes model. The most significant variables for each class:

0 Same Contraparty Ind = 0, Contraparty Ind Org Code = 2, same surname = 1, IN ind = 1, sūnui, mamai, vyrui, Event Activity Type Code = 572

1 grąžinimas, same surname = 0, užsakymas, refund, permokos, order, OUT ind = 1, IN ind = 0

2 išvedimas, pinigų, withdrawal, Contraparty Ind Org Code = 1, same surname = 0, money, laimėjimai, pavedimas

3 pienas, Contraparty Ind Org Code = 1, pristatytą, žaliava, susietoji , OUT ind = 0, IN ind = 1

4 žalos, Ind, Event Currency Code = EUR, atlyginimas, pagal, Contraparty Ind Org Code = 1, Event Activity Type Code = 920, OUT ind = 0

5 pagal, atsiskaitymas, sąskaita, faktūra, SF, Event Currency Code = EUR, Contraparty Ind Org Code = 1, same surname = 0, OUT ind = 0

6 kareiviams, buitinės, išmokos, Event Currency Code = EUR, Contraparty Ind Org Code = 1, šauktiniams, mėn, karo, prievolininkams

7 kredito, sutartis, Event Currency Code = EUR, Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, Country Code id = LT, didinimas, sumos, vartojimo

8 prekės, same surname = 0, Contraparty Ind Org Code = 2, vinted, suknelė, batai, rubai, OUT ind =0

9 vbnr, Contraparty Ind Org Code = 1, vykd, bylos, vykdomoji, VB, Event Currency Code = EUR, Country Code id = LT

10 nuoma, butą, Event Currency Code = EUR, Same Contraparty Ind = 0, Country Code id = LT, same surname = 0, Contraparty Ind Org Code = 2, nuomai, patalpų, OUT ind =0

11 paslauga, Event Currency Code = EUR, Same Contraparty Ind = 0, Country Code id = LT, same surname = 0, OUT ind = 0, konsultacija, Contraparty Ind Org Code = 2

12 dovana, podarok, Same Contraparty Ind = 0, Event Currency Code = EUR, Contraparty Ind Org Code = 2, OUT ind =1, dovanuoju, gift

13 paskola, Event Currency Code = EUR, IN ind = 1, same surname = 0, paskolai, dienu, Contraparty Ind Org Code = 1, OUT ind =1

14 skola, Event Currency Code = EUR, Same Contraparty Ind = 0, skolinu, dolg, Country Code id = LT, Contraparty Ind Org Code = 2, same surname = 0, IN ind =1

15 payout, balance, Event Currency Code = EUR, Event Activity Type Code = 2848, Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, Country Code id = EE

16 Event Currency Code = EUR, Contraparty Ind Org Code = 1, nuolatinio, mokama, gyventojo, pajamų, lietuvos, mokestis, Same Contraparty Ind = 0, OUT ind = 0

17 kišenpinigiai, Event Currency Code = EUR, Same Contraparty Ind = 0, IN ind =1,

kišenpinigai, Contraparty Ind Org Code = 2, same surname = 0, Event Activity Type Code = 572, OUT ind =1

18 Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, same surname = 0, Event Currency Code = EUR, OUT ind = 0, IN ind = 1, Country Code id = LT, Event Activity Type Code = 920, tmp, įeinantis

19 ūkio, išlaidoms, reikmėms, Event Currency Code = EUR, Same Contraparty Ind = 0, ukinėms, avansas, IN ind = 1, Contraparty Ind Org Code = 1

20 Contraparty Ind Org Code = 2, Same Contraparty Ind = 0, OUT ind = 0, IN ind = 0, same surname = 0, mokėjimas, papildymas, pavedimas, Country Code id = LT

21 Contraparty Ind Org Code = 2, Same Contraparty Ind = 0, same surname = 0, mokėjimas, ačiū, OUT ind = 1, IN ind = 1, papildymas, pavedimas, pervedimas, pinigų, sąskaitos

22 Same Contraparty Ind = 0, same surname = 0, IN ind = 1, Contraparty Ind Org Code = 1, atlyginimas, mėn, avansas, užmokestis, alga, darbo

23 pensija, Event Currency Code = EUR, Same Contraparty Ind = 0, OUT ind = 0, Contraparty Ind Org Code = 1, IN ind = 1, pension, for, įeinantis, Country Code id = LT

24 vaikui, Event Currency Code = EUR, Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, same surname = 0, vaikam, išmoka, administracija, savivaldybė

25 Event Currency Code = EUR, Same Contraparty Ind = 0, Contraparty Ind Org Code = 1, Country Code id = LT, išmoka, ppmoks, sodros, byla, data, nurodymas, mokėjimo

26 dividendai, Contraparty Ind Org Code = 1, Country Code id = LT, same surname = 0, isin, mok, išskaičiuota, eurvnt, užsienyje, usdvnt, eur, Event Currency Code = EUR, OUT ind = 0

27 stipendija, Event Currency Code = EUR, Country Code id = LT, Contraparty Ind Org Code = 1, sąskaita, mėn, studentai, stipendijos

28 alimentai, IN ind = 1, Country Code id = LT, vaiko, išlaikymui, pinigai, elementai, vaikui, same surname = 1, Contraparty Ind Org Code = 2

29 Same Contraparty Ind = 1, Event Currency Code = EUR, Contraparty Ind Org Code = 2, OUT ind = 1, IN ind = 1, Country Code id = LT, Event Activity Type Code = 752

Generalized linear model. The most significant variables for each class:

0 same surname = 1, Country Code id = LT, mamairemontui, žmonos, sumokėtum, vaikui, Contraparty Ind Org Code = 2, Same Contraparty Ind = 0

1 užsakymas, grąžinimas, paskirtis, Same Contraparty Ind = 0, Country Code id = LT, same surname = 0, grąžinamos, OUT ind 1, IN ind 0

2 išvedimas, laimėjimai, pavedimas, withdrawal, money, pinigų, depozito, OUT ind 0, IN ind 1, Contraparty Ind Org Code = 1

3 patiekta, apželdinimą, įsipareigojimai, dekl, susietoji, kooperatinė, paraiškos, projektams, galvijų, pienas, žaliava, produktų, pristatyta

4 žalos, Ind, priemoka, ministerijos, atlyginimas, sveikatos, visuomenes, mėnesi, apskaičiuota

5 faktūros, pagal, išankstinė, sf, fakt, sąskaita, nr, bylinėjimosi

6 kareiviams, buitines, skatinimo, kaupiamoji, šauktiniams, karo, prievolininkams, sausio, išmokos, technika

7 terminas, vartojimo, kredito, filialas, sumos, sandorį, partneriai, įsiskolinimas, sutartį, draudimo

8 ssp, mažmena, pirkinio, pirkti, pagal, prekes, batukams, treningai, kosmetinė, vinted

9 bylinėjimosi, vykd, VB, vykdomoji, byla, vykdrastai, vykdomosios, identifikatorius

10 ryšio, būsto, kompensacija, nuomininkas, zemesnuommok, pardavimas, nuoma, pradinis, turtas

11 prekiupaslaugu, teikimo, paslaugos, maketavimo, kurjerio, konsultacija, kirpėjo

12 dovanoti, dovana, myliu, dovanelė, vardadienio, gimtadienio, vakarienei, Contraparty Ind Org Code = 2

13 smspinigai, skaičius, draudimų, dienų, bustas, paskola, dydį, likučio, padengimui, bobutės, studijų, palūkanos

14 delspinigai, skola, skolinu, skolininkas, dolg, otdaju, dokumento, mano, išieskojimas

15 payout, balance, rocpid, Same Contraparty Ind = 0, Country Code id = EE, food, return, taxi

16 nuolatinio, gyventojo, pajamu, lietuvos, mokamas, mokestis, OUT ind 1, IN ind 0, dalyvio, mokymo

17 sak, nr, kišenpinigiai, kišenpinigai, išmokėjimo, mėn, same surname = 0

18 paskolos, atsiskaitimas, dalyviui, pervedimas, mokėjimas, įeinantis, Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, OUT ind = 0, IN ind = 1, tmp

19 ūkinė, atlyginimui, užsienį, ūkinėms, įmonė, išlaidoms, partneriui, expenses, namų, reikmėms, kooperatyvas

20 mokėjimas, papildymas, pavedimas, Contraparty Ind Org Code = 2, Same Contraparty Ind = 0, same surname = 0, OUT ind = 0, IN ind = 0 , papildymas, pavedimas, Country Code id = LT

21 OUT ind = 1, IN ind = 1, pavedimas, pervedimas, pinigų, Same Contraparty Ind = 0, Contraparty Ind Org Code = 2, same surname = 0, mokėjimas, Event Currency Code = EUR, Country Code id = LT

22 du, atsiskaitymai, gimnazija, atlyginimas, alga, darbo, užmokestis, avansas, av, apysk, Contraparty Ind Org Code = 1

23 valstybinių, pensija, pensijos, pension, Country Code id = LT, Event Activity Type Code = 7148, išmokėjimas

24 darzelisdu, įstatymą, vaikui, gyvenimo, savarankiško, miesto, išlaikymui, išmokos, miesto, alytaus, savivaldybė, support, kauno, mokamo, neformaliojo

25 sodros, byla, išmoka, žiniaraščio, administracijos, panevėžio, maitinimosi, parama

26 apskaičiuota, pva, pusm, eurvnt, ignitis, dividendai, išskaičiuota, usdvnt, real, estate, eso, operatorius

27 lsmu, stipendija, užsienio, svk, studentai, teikimą, stipendijų, profesinio, programa, stipendium, viešoji, doktorantų, stažuotė

28 išlaikymui, vaiko, alimentai, alimony, elementai, alementus, sūnui, dukros, vaikų, renta, menėsiui

29 papildymas, sąskaitą, mobiliąją, Contraparty Ind Org Code = 2, Same Contraparty Ind = 1 , OUT ind = 1, IN ind = 1, pinigų, pervedimas, sąskaitos

Gradient Boosting model. The most significant variables for each class:

0 same surname=1, sūnui, mamai, Contraparty Ind Org Code=2, pavedimas, mokėjimas

1 užsakymas, grąžinimas, Contraparty Ind Org Code=1, prekės, refund, OUT ind 1, IN ind 0

2 išvedimas, pinigų, Contraparty Ind Org Code = 1, withdrawal, depozitas, money, laimėjimai

3 pienas, galvijų, susietoji, žaliava, produktų, pristatytą, Contraparty Ind Org Code = 1

4 žalos, Ind, atlyginimas, priemoka, ministerijos, apskaičiuota, Contraparty Ind Org Code = 1, OUT ind = 0

5 sąskaita, faktūra, sf, pagal, nr, bylinėjimosi, Contraparty Ind Org Code = 1

6 kareiviams, buitines, prievolininkams, skatinimo, kaupiamoji, šauktiniams, karo, išmoka, Contraparty Ind Org Code = 1, mėn

7 kreditas, sumos, vartojimo, skolinimosi, Contraparty Ind Org Code = 1, Country Code id = LT, didinimas, numeris

8 prekės, Contraparty Ind Org Code = 2, OUT ind = 0, suknelė, batai, džinsai, vinted

9 vbnr, vykdomoji, bylinėjimosi, vykd, VB, byla, vb, Contraparty Ind Org Code = 1, Event Currency Code = EUR

10 būtas, nuoma, Contraparty Ind Org Code = 2, patalpų, autonuoma, mėn

11 paslaugas, konsultacija, už, nr, kurjerio, OUT ind = 0, Contraparty Ind Org Code = 2, suteiktas

12 dovana, dovanoju, gimtadienio, Contraparty Ind Org Code = 2, gift, podarok, Event

Currency Code = EUR, OUT ind =1

13 paskolos, būsto, smspinigai, padengimui, bobutės , IN ind = 1, Contraparty Ind Org Code = 1, OUT ind =1

14 skola, skolinu, dolg, Contraparty Ind Org Code = 2, same surname = 0, IN ind =1, delspinigai, išieskojimas

15 payout , balance, Country Code id = EE, food, return, Contraparty Ind Org Code = 1, taxi, Event Activity Type Code = 2848

16 gyventojų, pajamu, lietuvos, nuolatinio, mokamas, mokestis, IN ind 0, Contraparty Ind Org Code = 1, OUT ind = 0

17 kišenpinigiai, kišenpinigai, išmokėjimo, mėn, Same Contraparty Ind = 0, IN ind =1, Contraparty Ind Org Code = 2, Event Activity Type Code = 572

18 įeinantis, mokėjimas, Contraparty Ind Org Code = 1, Same Contraparty Ind = 0, IN ind = 1, OUT ind = 0, tmp

19 ūkinėms, išlaidoms, partneriui, reikmėms, expenses, namų, IN ind = 1, Contraparty Ind Org Code = 1

20 Contraparty Ind Org Code = 2, OUT ind = 0, IN ind = 0, mokėjimas, papildymas, pavedimas, Same Contraparty Ind = 0, Country Code id = LT

21 Contraparty Ind Org Code = 2, OUT ind = 1, IN ind = 1, Same Contraparty Ind = 0, same surname = 0, mokėjimas, ačiū, pinigų, papildymas, pavedimas

22 DU, atlyginimas, darbo, užmokestis, Contraparty Ind Org Code = 1, avansas, mėn, alga, dienpinigiai

23 pensija, išmokėjimas, valstybinių, Country Code id = LT, Event Activity Type Code = 7148, Event Currency Code = EUR, OUT ind = 0, IN ind = 1, pension

24 išmoka, vaikams, savivaldybė, išlaikymui, support, Event Currency Code = EUR, Contraparty Ind Org Code = 1

25 sodros, byla, išmoka, žiniaraščio, administracijos, parama, Event Currency Code = EUR, Contraparty Ind Org Code = 1, Country Code id = LT, nurodymas

26 dividendai, eurvnt, išskaičiuota, real, eur, Contraparty Ind Org Code = 1, Event Currency Code = EUR, OUT ind = 0

27 stipendija, užsienio, studentai, teikimą, profesinio, programa, stipendium, Event Currency Code = EUR, OUT ind = 0, Country Code id = LT, Contraparty Ind Org Code = 1, mėn

28 išlaikymui, vaiko, alimentai, elementai, sūnui, dukros, menėsiui, alimony, same surname = 1, Contraparty Ind Org Code = 2

29 Same Contraparty Ind = 1 , OUT ind = 1, IN ind = 1, Contraparty Ind Org Code = 2, pervedimas, mobiliąją, sąskaitų, savo