Cooperation agreement no. 1.9-8/21-445-1

An early warning services for businesses

Prototype and business analysis – practical execution

Final report
30 April 2022

Team of
Experimental Statistics
Statistics Estonia

Tallinn 2022

Table of Contents

Introduction

This document presents the initial opinions and principles in the context of developing an early warning system for economic operators based on the main business analytical considerations. A prototype and conceptual solutions have been developed for launching the early warning service in Estonia. This material presents the situation concerning the prototype and the related materials as at 31 March 2022. The work on eliminating known issues and taking into consideration the feedback received will continue in April 2022. The final/supplemented report will be completed in May 2022.

The material presents the theoretical bases, the analytical considerations based on the bases, as well as the first results by those analysis components which can be used in the current stage of the work from a logical perspective (i.e. which are focussed on prototype-based moments, not the details of the final information technology solution). This material also includes comments on the situations in which the practical solutions are not aligned with previously widely shared theoretical opinion, with the reasons for the differences analysed and respective conclusions drawn.

This material consists of two chapters:

- Chapter I – theoretical bases;
- Chapter II – practical execution.

Chapter I presents the most important theoretical bases and references to the academic articles which the principles used in this work are based on. The chapter also includes first practical examples and some initial inquiries, clustering, conclusions, and the results of machine learning. These were added to the theoretical chapter in order to illustrate the connection between theoretical topics and practical work and to show more clearly how theoretical material has been used in practice.

Chapter II describes a practical development which is based on the theoretical positions provided in Chapter I and the practical interpretations of those positions. Chapter II is based on the previous chapter and discusses the technical points and solutions which the development of the prototype has reached by the time of submitting the document. Both the work process and the model developed are described. The results are highlighted and interpreted.

Some of the technical material related to the prototype is included in the main text of the document, some is provided in the annexes. The main text primarily includes the material which must be provided next to the text for comparison to facilitate understanding technical explanations; the material provided in the annexes is more indirectly illustrative and more voluminous from the technical perspective.

Different perspectives and methods were tested in the course of the practical execution, with some of them providing better, others weaker results. On the other hand, the point of developing a prototype is to analyse and achieve outcomes in different manners to decide based on the practical outcome what works and what does not, what could or could not be used as the basis for developing an information system.

Chapter I – theoretical bases

This chapter presents the theoretical bases and references to specific literature from which the theoretical bases originate. The chapter also highlights the practical first steps and initial outcomes, but primarily for the purposes of illustrating the use of theoretical positions in practical work. The development of the practical model is described more substantially in Chapter II.

1.1. General description of the nature of the issue from the perspective of business analysis

Focal to the analytical problem setting is the assessment of whether or not a specific company analysed may become insolvent. This assessment must be given with the backing of automatic computer algorithms. The assessment must take into consideration the area of activity of the company (the companies of different sectors may become insolvent with very different balance sheet and income statement statuses), the size of the company (companies of different sizes are financially vulnerable in different manners), its geographic location (the location may have a great impact on sales, as well as on the ability to obtain raw materials), and, to a certain extent, the legal context (different companies are regulated differently – for example, labour-intensive companies are considerably influence by the labour market legislation). The potential payment difficulties of a company must be assessed based on the possible combinations of all of the elements/definitions described above.

Machine-learning algorithms and the principles of AI-based work are used in developing the prototype. Therefore, the theoretical bases were worked on, the academic literature was analysed, and the solutions developed at different research institutions so far were examined. An analysis was conducted on what to take over from what has been done so far and which direction the work on the prototype should be steered (to obtain as adequate outputs as possible with an optimum input).

1.2. Use of references

A certain number of academic articles were examined to specify the theoretical perspectives and set the baseline framework. The articles selected describe the use of machine learning and artificial intelligence in predicting the economic results of companies and in identifying potential insolvency situations.

Below, brief summaries are provided of the most important inputs of the academic articles used in this work which are relied on in setting the baseline prerequisites and selecting the methods in designing this prototype.

The amount of literature examined and analysed in the course of the work exceeds those referred to here, only the sources used in this work in one way or another are referred to. The above means, however, that new sources may be added to the list of theoretical literature in the course of the work depending on the progress of the work and on the issues selected for solving in the future.

Several of the articles below were developed based on a certain specific context which does not directly overlap with the purpose of this work. In such cases, the part of the input overlapping the purpose of this work which is directly applicable to the work was picked from the articles.

Kou et al. (2021) suggest a bankruptcy prediction model for small and medium-sized companies which is based on the data of the transactions between those companies. This is an alternative solution to the models based on an analysis of the economic ratios of the accounting data of a company (balance sheet, income statement) and the analysis of so-called classical ratios is left aside completely. The authors of the methodology claim that B2B transaction information is more reliable than accounting reports. Transaction information is also dynamic information, while accounting reports express the situation at a certain moment in time and are static by nature. Transaction information shows the current situation and changing thereof, while accounting reports show the situation of the company at some point in the past. Transaction information also enable calculating the net present value, which is one of the important indicators of the 'health' of a company, but cannot be calculated based on accounting reports.

Perboli and Arabnezhad (2021) explain that it is possible to make primarily short-term predictions and mainly in the case of larger companies as a result of the analysis of the ratios of accounting reports. They suggest a machine learning method which enables making bankruptcy predictions for small and medium-sized enterprises in a longer perspective (up to 60 months) and significantly increase the accuracy of the predictions for short-term periods (up to 12 months). The authors claim that the methodology suggested by them forms a good basis for shaping economic policy, i.e. it enables interpreting the outcome of the analysis of ratios more widely.

Perboli and Arabnezhad (2021) stress that in order to obtain an adequate outcome from machine-learning, it is important to first choose the right indicators to analyse, and suggest their own points of

origin for this. They discuss in depth how to deploy machine learning in predicting and how to ensure the inputs for analytical support for the outcome. The principle of double training of data is introduced. The authors also provide a framework for interpreting the results.

Qu et al. (2019) highlight different machine learning and neural network analysis-based models which they believe to be relevant and with a high potential in predicting the situations of insolvency based on initially available accounting information describing the financial situation. They discuss the following methods of machine learning: Multivariant Discrimination Analysis (MDA), Logistic Regression (LR), Ensemble method, and Support Vector Machines (SVM). The neural network analysis methods discussed include the following: Neural Networks (NN), Deep Belief Network (DBN) and Convolutional Neural Network (CNN).

Shi and Li (2019) highlight in their extensive literature-based research the types of solvency issues which can be solved by using machine-learning methods (or methods similar to machine-learning) and which methods are used to solve the problems. The definitions indicating the issue of insolvency highlighted by them include: bankruptcy prediction, default prediction, financial failure, financial distress, insolvency, business failure. The methods/activities used to solve the problems include: neural network, support vector machine, decision tree, genetic algorithm, fuzzy, rough set, data mining, case-based reasoning, data envelopment analysis, adaboost, K-nearest neighbors, bayesian network.

Cialone (2020) presents a causality analysis in which 14,965 Italian companies were analysed based on balance sheet ratios, with 13,845 in business and 1,120 bankrupt, and created a model for assessing the development of insolvency within one year. In total, 88 input variables are used in the model. The neural network-based analyses Deep Fully Connected and Convolutional Neural Networks were used.

Cialone (2020) refers to Multiple Discriminant Analysis and Logistic Regression as the most classic (used more extensively before) models for predicting the potential insolvency of companies, while Random Forests, Boosting, and Neural Networks have been deployed more often recently. Contingent Claim analysis as a less statistical method, which is used increasingly, is also very common. The importance of Deep Neural Networks is increasing.

Chen (2020) explains that machine learning methods are becoming increasingly more popular in the analysis of financial data, but the clarity of interpreting the outcome remains the main bottleneck. He suggests the CART or (boosted) ensemble of decision trees methods. If not proceeding from these bases, interpretation of the outcome can be compared to the so-called 'black box' view.

Ucoglu (2020) provides an overview of how the methods of machine learning are used in accounting and auditing. He provides an overview based on four large audit bureaus of how they use machine learning in their work.

The companies analyses are: Ernst and Young, PwC, Deloitte, and KPMG. Ucoglu predicts that 30% of all audits will be conducted by using machine-learning and artificial intelligence-based methods by 2025.

Cao and You (2021) highlight in their approach that machine learning methods, especially those which do not have a linear structure, ensure a considerably more informative and accurate capability of predicting cash flows compared to the regular methods of predicting cash flows. Another important factor is that the models of machine learning enable finding reasonable connections from the economic perspective which may remain undetected by using regular prediction models. Machine learning may also unearth completely new important points which have not been included in the models before.

Choudhry (2018) shows how machine learning can be used in banking in analysing balance sheets on the one hand and in the management of customer service on the other hand, i.e. it includes the analytical as well as human dimensions. Another important feature of the approach is that the author specifies by which methods and in which cases to apply unsupervised versus supervised learning.

Aliaj et al. (2020) explain that similar economic data may lead some companies to the state of insolvency but not others. The decision is basically often made by the bank granting loans which may recall their loans at certain points or refuse to be flexible in changing repayment schedules. The auditors examine the data of the loans given to Italian companies in interaction with the moment of those companies becoming insolvent.

Andrés et al. (2004) highlight in the model of predicting financial results proposed by them those aspects and focuses which are reasonable to include in making predictions, what should be taken into

consideration most, and what provide predictions of the best quality. Abstract implementation of machine learning may not produce very realistic outcomes.

Amel-Zadeh et al. (2020) use the random forest model of machine learning to find the variables of accounting which are most important in predicting most adequately free cash flows of the company and other indicators which indicate an improvement of general economic operations.

The approach of León et al. (2016) is based on the assumption that the balance sheet of a company (bank) is unique for each specific company and describes the company in question. Their research with the help of the neural networks machine learning method confirms this. If a machine is capable of recognising the balance sheets of a company and describe companies based on their balance sheets, the machine can also identify the economic health of those companies based on the balance sheets.

Petropoulos et al. (2019) propose a dynamic balance sheet simulation engine which predicts the key balance sheet indicators based on the deep learning method. The authors compare the results obtained with other methods of predicting balance sheets – static balance sheet prediction methods and dynamic balance sheet prediction methods. Machine learning will significantly reduce the number of prediction errors, especially in the case of a one-year prediction period.

1.3. Two conceptual perspectives

The work includes two conceptual perspectives for predicting insolvency:

(a) the so-called classic logic based on economic ratios;

(b) the logic based on the analysis of economic transactions.

The economic ratio-based logic. The ratios of the balance sheet and income statement are used to predict potential future scenarios by using machine learning methods. Being aware of which values of the ratios indicate a declining solvency, the situations of a company potentially becoming insolvent can be predicted. The majority of the academic literature referred to in this material describes the predictions which are based on economic ratios.

Economic transaction analysis-based logic. Kou et al. (2021) suggest an insolvency prediction model for small and medium-sized companies which is based on the data of the transactions between those companies. This is an alternative solution to the models based on an analysis of the economic ratios of the accounting data of a company (balance sheet, income statement) and the analysis of so-called classical ratios is left aside completely. In this work, machine learning methods are used for examining B2B transactions as an alternative to the classic ratio-based logic.

Both the economic ratio-based logic and the economic transaction-based logic will complement one another, both analyses will be conducted, and machine learning methods will be deployed for both. The two analyses will be compared and an overlapping part found. If the outcome of one of the analyses differs significantly from the outcome of the other, there is a reason for further analysis to determine why this outcome was achieved and where a significant error may lie.

1.4. Support of Aruandlus 3.0 to this project

The early warning service will be more accurate if it is possible to use the information gathered within the framework of the Aruandlus 3.0 project as an input. Statistics Estonia is working continuously and systematically, and in wider cooperation with other state agencies (e.g. Ministry of the Environment, Ministry of Justice, Centre of Registers and Information Systems, Tax and Customs Board, Ministry of Economic Affairs and Communications, Bank of Estonia) to improve this information inquiry in order to take the concept of real-time economy more widely into use as so-called data-based reporting. Statistics Estonia is already receiving first data and there is reason to believe that increasingly more information will be received in the future.

Aruandlus 3.0 will help to increase the quality of the input information of the early warning service due to three aspects:

- improves the level of detail of the data;
- Improves the speed of receiving data;
- provides a further opportunity for modelling economic data.

The level of detail of the data. Aruandlus 3.0 is used to collect the transaction data of businesses. The transaction data are more accurate than the data from annual reports. This fact enables analysing the

company more accurately than is possible based on the consolidated balance sheet or income statement. The algorithms of early warning can be made more accurate, the computer can use more accurate parameters as the input for machine learning, and the output can be used as a source of further information.

The speed of receiving data. Aruandlus 3.0 is designed to function as a real-time solution, which means that in the case of the normal functioning of the system, it is possible to obtain the data on a monthly basis (if not faster). This provides an opportunity to assess what is happening with a company and whether or not the company remains in the same/comparable situation found based on the analysis conducted on the basis of the latest annual report. The early warning system gains an additional input for assessing the stability of the company.

Modelling of economic information. The annual reports of companies present the situation at a certain point in time (in the past) and the information may be a year or a year and a half old. If real-time transaction information is obtained via Aruandlus 3.0 (or with a shorter delay compared to the reports), this information (and the previously known relationships) can be used to model current balance sheets and income statements (with a certain level of accuracy). The latter, in turn, allows for issuing almost real-time early warning assessments (or with a temporal shift of a month or two, not based on information from a year ago).

From the temporal perspective, we may currently strongly believe that the amount (more data transmitters) and versatility (data from a higher number of different data points) of the information received within the framework of the Aruandlus3.0 project will improve, which will allow wider use of the data as an input for the early warning service. Any further data received within the framework of the Aruandlus 3.0 project will be deployed as input in the development of the early warning service as soon as they are received.

1.5. Determining the sector

Determining the sector is important, as potential insolvency is described completely differently in the case of different sectors from the perspective of ratios as well as B2B transactions. For example, in the case of determining the insolvency of the following companies, completely different parameter values must be set for the balance sheet, income statement, and transaction values for predicting insolvency (the examples are illustrative and the names informal):

- a large wholesale company;
- a small consultation company;
- a research development centre;
- an agricultural undertaking;
- an individual taxi driver;
- a metal industry;
- a dental care office;
- a labour dispute office;
- a hamburger kiosk;
- a port;
- a cleaning company involving three people.

The names provided above are examples, but it is clear by taking a look at them that they differ by their need for current assets, potential need for long-term loans, principles of stock management, need for production tools, general balance sheet volumes, profitability, turnovers, turnover speed, sensitivity to seasons, and many other parameters. Thus, it is not possible to develop a sufficiently accurate and appropriate machine learning algorithm which could find the parameters indicating insolvency for all of the companies listed above; those parameters (especially the so-called 'red values' of the parameters) are different. Solutions are, however, possible if other companies of a similar area of activity are observed.

The sectors are defined based on the Estonian Classification of Economic Activities (EMTAK) codes. However, it is not possible to take a simple approach within the framework of this work by using the EMTAK code on the first level (i.e. using the first number of the code), on the second level (by using the first two numbers), etc. This approach would be technically simpler but would paint a misleading picture. The examples below explain the logic of providing a misleading picture.

For example, if we intend to define an area of activity from the medical sector, the following areas of activity must be involved, for example.

87101 Residential nursing care activities
86211 Provision of general medical treatment

The example of the two areas of activity above makes it clear that the medical sector should include both of the aforementioned areas of activity. Thus, it is not possible to proceed to the second level of the EMTAK; we must remain at the first level of the EMTAK and include all areas of activity with EMTAK codes beginning with '8'. On the other hand, in this case, the following area of activity would be included in the medical sector:

88911 Child day-care activities

It is obvious that child day-care activities are not medical care, the medical sector. It is, however, an important area of activity which is used, is specified as their area of activity by businesses, and is not just an activity which could be
ignored as 'noise'. However, proceeding to the second level of the EMTAK and including the code '86', for example, we exclude '87', residential nursing care activities – this is still a health care institution and the medical sector cannot be discussed without involving it.

If we wish to define the financial sector, we are, among other things, faced with the

following areas of activity:

64191 Credit institutions (banks)
64301 Investment funds
64911 Financial leasing
65201 Reinsurance
65301 Pension funding
66111 Administration of financial markets
66121 Security and commodity
contracts brokerage
68101 Buying and selling of own
real estate

If we remain at the first level of the EMTAK, buying and selling of real estate is included in the financial sector, but it does not fit in the sector, of course. Thus, we cannot remain at the first level. Proceeding to the second level, the first two numbers would be used for determining the area of activity, in the context of this example, the financial sector. Which two numbers should be used, '64', '65', or
'66'? Actually, we need all three.

Next, in the case of the real estate sector, we would be faced with the following situation:

41101 Development of building projects
41201 Construction of residential and
non-residential buildings
42111 Construction of roads and
motorways
42121 Construction,
maintenance and repair of
railways and underground
railways
42131 Construction of bridges
and tunnels
68101 Buying and selling of own
real estate
68201 Rental and operating of own
or leased real estate
68311 Real estate agencies
68321 Management of buildings and
rental houses

All of the areas of activity referred to above fit under the sector of real estate activities. It is, however, especially important to include both the '4' activities based on the first number of the EMTAK code, as

well as the ‘6’ activities. By using a simplified perspective, we should use the first level of the EMTAK code, codes ‘4’ and ‘6’. This cannot be done, though, as the areas of activity listed below would be included in this case from the previous example, which certainly are not activities of the real estate sector:

64191 Credit institutions (banks)
64301 investment funds
64911 Financial leasing
65201 Reinsurance
65301 Pension funding
66111 Administration of financial markets
66121 Security and commodity contracts brokerage

If we attempt to define education, we will come across the following

areas of activity, for example:

85101 Activities of creches
85102 Activities of nurseries
85521 Music and art education
85591 Language training
85592 Computer training
85529 Other hobby education

We cannot remain at the second level of the EMTAK, i.e. code ‘85’, as this would also include the activities of creches and nurseries. This may be appropriate in certain cases, but those institutions should be left aside if education is analysed more narrowly. Therefore, we cannot remain at the second level and will proceed to the third level, to code ‘855’. This, however, still includes other hobby education (acting groups, photography groups), which may not be appropriate. Moving on to the fourth level, code ‘8559’, we manage to include (in the context of the example above) language training and computer training, but music and art education would be left out. These are part of fully regular education, though, and cannot be excluded.

The examples above are just a few of many but sufficiently figurative to show that it is not possible to use a simplified approach in determining areas of activity (by picking a certain level, some points of the EMTAK).

The solution suggested involves defining the sectors by combining the five-digit EMTAK codes; thereat, the combinations will be put together based on the respective terms of reference. Annex 1 to this material specifies the four areas of activity defined: (a) the medical sector, (b) the sector of financial services, (c) the real estate sector, and (d) the education sector.

This definition of the sectors specified in Annex 1 is conditional, as it is possible to define the same sectors in another manner, more narrowly or widely, if necessary, by using other EMKAT codes or those necessary to another extent, respectively. It is also important to define sectors/areas of activity/industries so that the conclusions drawn based on them would provide actual benefits in the issuing of early warning signals and, for this purpose, it is important to define those sectors in a targeted manner. There is no quick and easy solution.

1.6. Terminological definition of insolvency

The aim of this work is to find factors indicating that a company is about to become insolvent as early as possible. The model created is not designed to predict bankruptcies or forecast liquidation proceedings. However, based on the academic literature referred, the majority of insolvency analysis-focussed machine learning solutions are designed for predict bankruptcies.

Bankruptcy is the last level of deepening insolvency, but as machine learning methods have been developed to predict bankruptcies, the same methods can also be used to identify earlier phases of insolvency (one or two years or more before) before bankruptcy is reached. Those earlier phases (in which there is a risk of insolvency or the first signs of a mild insolvency) are the point where it may be the right time to issue an early warning sign for the company based on the specifics of the

industry/sector. Thus, while this analysis is based on academic literature on using machine learning methods to predict bankruptcies, only the parts required for and supporting the identification of the early insolvency phase are taken, not those used for identifying/predicting the (general) last phase of insolvency.

Based on subsection 1 (1) of the Bankruptcy Act, bankruptcy means the insolvency of a debtor declared by a court ruling. Subsection 1 (2) of the Bankruptcy Act explains that a debtor is insolvent if the debtor is unable to satisfy the claim of a creditor that has fallen due and such inability, due to the debtor's financial situation, is not temporary. Subsection 1 (3) of the Bankruptcy Act adds the definition of the insolvency of a debtor who is a legal person, stating that this person is insolvent if the assets of the debtor are insufficient for covering the obligations thereof and such insufficiency is not temporary.

At the time of issuing an early warning, the respective company is certainly not permanently insolvent and has probably also not yet reached the situation of temporary insolvency. However, the Bankruptcy Act also includes references to the two analysis approaches:

- what is the potential that the claims of the creditors cannot be satisfied;
- what is the situation with assets.

The capability to satisfy claims. This is an analysis approach which examines the current assets, the cash turnover rate, the general liquidity, and all related indicators of companies. The aim is to understand whether the company may end up in a situation in which its financial position becomes so non-liquid that it will not be able to cover the liabilities arising from its economic operations, even though the company is not in a difficult financial position.

The economic situation. This analysis approach examines how the financial position of a company changes in time and whether the change may indicate potential insolvency in the future. The general principles include monitoring the amount of equity and the minimum share of liabilities (moving towards difficulties, the share of liabilities usually increases and the equity decreases relatively as well as nominally).

Combining the two perspectives referred to above, for example, if there are signs that the capability to satisfy claims decreases while the financial situation deteriorates, there is certainly a reason for issuing a warning signal (if it is not too later, the signal may potentially be issued sooner).

The pre-insolvency phase based on which the warning is issued may be referred to by very different names (its technical definition is more important and this work is focussed on it). Academic literature also uses different names to refer to the insolvency analysed by machine learning. Shi (2019) mentions the following concepts:

- Bankruptcy prediction;
- Default prediction;
- Financial failure;
- Financial distress;
- Insolvency;
- Business failure.

Thus, the moment of insolvency and the pre-insolvency phase (in which the early warning signals are issued) can be referred to differently from the terminological perspective; the important thing is for it to be technically accurate and for machine learning solutions to be trained to identify it.

1.7. Using economic ratios

The economic ratios primarily used in this work (the ratios found based on the balance sheet and income statement) and principles of the logic of using thereof in this work are presented in Annex 2. This is an initial selection which may be changed and probably supplemented in the course of the work.

Cialone (2020) highlights the following balance sheet/income statement indicators and ratios based on them which he built his neural network-based analysis on (in the order used by Cialone):
Current Assets
Bank Debt to Sales
Debt / EBITDA ratio
Total Debt to Equity
Receivables Average Collection
Period Payables Average Settlement

Period EBIDTA to Interest Expenses
Working Capital to Revenues

Current Ratio
Total Fixed Tangible Assets to Equity
(Equity + Long Term Debts) /
Fixed Assets
Current Debts to Total Debts
Long Term Debts To Total Debts
Interest Expenses to
Gross Sales
Total Equity
Net Financial Position
Total Assets to Equity
Return on Equity
Return on Assets
Return on Sales
Return on Investment
Gross Sales
Gross Working Capital Turnover
Invested Capital Turnover

Total Fixed Assets
Total Liabilities to Equity
Total Current Liabilities
Equity to Total Assets
(Assets-
Inventories) / Debts
Long Term debt
Total Assets
EBITDA
EBITDA / Gross
Sales
Total Credits

Total Debt

Number of Employees

In general, the above matches the logic of the indicators provided in Annex 2 to this material. Balance sheet and income statement indicators and the ratios calculated on the basis thereof were used and the outcome achieved was used as input in deploying machine learning methods. Nominal balance sheet and income statement figures hold a relatively large share (compared to the perspective that analysis is often only based on ratios and nominal figures are not used).

The table below specifies the indicators divided into categories.

Nominal figures	Structure of capital	Efficiency	Liquidity	Profitability
Current Assets	Bank Debt to Sales	Receivables Average Collection Period	Current Ratio	Return on Equity
Total Equity	Debt/EBITDA Ratio	Payables Average Settlement Period	Gross Working Capital Turnover	Return on Assets
Net Financial Position	Total Debt to Equity		Invested Capital Turnover	Return on Sales
Gross Sales	EBITDA to Interest Expenses			Return on Investment
Total Fixed Assets	Working Capital to Revenues			
Total Current Liabilities	Total Fixed Tangible Assets to Equity			
Long Term Debt	(Equity + Long Term Debts) / Fixed Assets			
Total Assets	Current Debts to Total Debts			
EBITDA	Long Term Debts to Total Debts			

Total Credits	Interest Expenses to Gross Sales			
Total Debt	Total Assets to Equity			
Number of Employees	Total Liabilities to Equity			
	Equity to Total Assets			
	(Assets-Inventories)/ Debts			
	EBITDA/Gross Sales			

Table 1: Distribution of the indicators by categories.

The tables above indicate that Cialone (2020) primarily focussed on nominal balance sheet and income statement figures, on the one hand, and on analysing the ratios of the structure, on the other hand. From the perspective of efficiency, only the periods of the receipt of accounts receivable and the temporal framework of paying the bills are examined (those two indicators highlight how cash moves in the company and out of the company, which makes them key indicators).

As regards to liquidity, Cialone (2020) examines three indicators. First, the classic Current Ratio which shows the extent to which current assets cover current liabilities – the more current assets to cover current liabilities, the better. Next, the operating capital turnover and the invested capital turnover are examined, how many times the aforementioned capitals are turned in a year (i.e. how many times the company uses its operating capital in a year and how many times it uses its invested capital – the more times, the better).

Cialone (2020) examines profitability in four categories, i.e. profitability against total assets, equity, sales, and investments. Equity and the balance sheet volume are general indicators of the size (economic scope) of a company, while sales and investments are the remaining two key indicators – sales (turnover) show how the company is doing in the market, investments show how the owners have contributed to the sales (how much money they have invested to achieve the turnover and earn a profit from the turnover).

At first glance, the analysis of Cialone (2020) appears slanted towards nominal figures and capital structure, analysing what has been introduced from the perspective of efficiency, liquidity, and profitability; however, the inputs most frequently seen as key indicators have been selected.

The selection of Cialone (2020) is also justified by his opinion that the majority of the work (most of it) is spent on preparing data and on ensuring the right input parameters. He used the data of active, as well as bankrupt companies; thereat, the data are significantly tilted towards active companies (the number of those companies involved is significantly higher). Two methods have been used to balance the data: (A) the Synthetic Minority Oversampling Technique (SMOTE) and (b) the Adaptive Synthetic Sampling Method (ADASYN).

The preparations for the prototype based on this work are, on the one hand, based on the analysis of theoretical academic literature (from the perspective of business analysis, as well as machine learning); on the other hand, however, practical tests are being carried out to find the functioning of certain theoretical concepts on real-time data. Therefore, the approach taken did not involve first completely examining the issue from the theoretical perspective and then starting to work on the practical side. The practical side is being developed in parallel with examining the theoretical concepts.

The initial analysis uses the annual report data and the database of bankrupt companies. Linking the database of bankrupt companies with the entire database of annual reports provided a dataset comparable to the one used by Cialone (2020). As the real-time dataset is strongly tilted towards operating companies (the amount of the data from bankrupt companies is low compared to the data of actually operating companies), the Synthetic Minority Oversampling Technique (SMOTE) was also used to balance the data. The analysis is based on the data of the annual reports of the past five years from all companies.

In the case of bankrupt companies, the database is formed so that five years were counted back from the moment of bankruptcy, i.e. one year before bankruptcy was declared, two years before bankruptcy was declared, etc. The companies were declared bankrupt in different years; therefore, the database was formed so that the five pre-bankruptcy years may differ by companies, but it is important to have five years in operation before bankruptcy for each company gone bankrupt (this enables looking for the patterns preceding bankruptcy).

First, the clustering analysis was conducted (by the k-means algorithm) to see the overall situation in

business. The number of clusters was determined by using the Elbow method – a method for determining the number of clusters in the case of which adding a further cluster does not provide a significantly better additional outcome. The selected number of clusters is five; the figure below presents the curve of the Elbow method used to select the number of clusters.

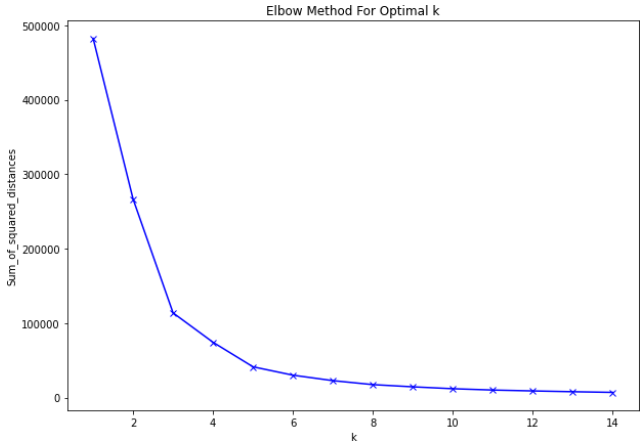


Figure 1. Determining the number of clusters.

The k-means algorithm divides n objects in k clusters so that each object is placed in the cluster the centre of which it is closest to.

The indicators used in initial clustering include:

BI_100_1 – Balance sheet, current assets
BI_150_1 – Balance sheet, tangible assets
BI_180_1 – Balance sheet, total fixed assets
BI_190_1 – Balance sheet, total assets
BI_240_1 – Balance sheet, short-term loans
BI_250_1 – Balance sheet, payables and prepayments
BI_290_1 – Balance sheet, current liabilities
BI_310_1 – Balance sheet, non-current liabilities
BI_370_1 – Balance sheet – total liabilities
BI_400_1 – Balance sheet, share capital
BI_40_1 – Balance sheet, cash
BI_40_2 – Balance sheet, cash at the end of the previous reporting period
s1 – BI_100_1/ BI_190_1 – share of current assets in assets
s2 – BI_290_1/ BI_190_1 – share of current liabilities in assets
s3 – BI_310_1/ BI_190_1 – share of non-current liabilities in assets
s4 – BI_370_1/ BI_190_1 – total liabilities in assets
s5 – BI_400_1/ BI_190_1 – share capital is assets
s6 – BI_40_1/ BI_100_1 – share of cash in current assets

The above shows that the initial clustering used the nominal balance sheet figures and the capital structure ratios. The balance sheet volume was taken (baseline) and liabilities (insolvency is the inability to fulfil obligations) and current assets (current assets show the financial space for

manoeuvring, as well as to what extent liabilities can be covered quickly if necessary – one of the criteria of liquidity) were concentrated on.

The balance sheet figures were focussed on, as the balance sheet shows the status, while the income statement would show the dynamics within the year. Including the patters of the dynamics within the year is also important, but the system is created stage-by-stage, starting from the static foundation.

The table below presents the initial results of clustering.

clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	202,479	109,020	319,297	521,742	45,686	67,410	115,668	90,903	217,362	18,326	53,184	45,727
1	323,292	250,233	569,848	836,682	91,308	130,467	215,077	186,630	421,360	17,568	59,752	54,504
2	58,414	44,133	64,355	124,112	3,410	18,066	32,470	18,724	56,763	943	30,585	28,267
3	241,737	125,545	328,041	569,779	52,436	99,852	154,158	101,494	267,362	13,824	53,021	46,453
4	34,822	23,583	43,630	78,344	2,768	9,764	17,945	15,126	38,506	622	23,074	20,430

Table 2: Initial results of clustering:

The table below specifies the number of entries of economic operators by clusters (several entries per one company).

Cluster number	Number of companies in the cluster
0	493,985
1	463,494
2	169,205
3	505,960
4	149,264

Table 3: Number of companies in the cluster.

As explained above, the details of the companies which have become bankrupt were added to the dataset The table below presents the number of companies which have gone bankrupt by cluster

Cluster number	Number of companies gone bankrupt
0	237
1	658
2	12
3	471
4	14

Table 4: The number of companies gone bankrupt.

Annex 3 presents the number of companies which have gone bankrupt by year and cluster.

One of the main bases used in initial testing was the balance sheet volume. The volume of the assets shows the scope of the economic activity, the economic power, and stability (a higher volume should indicate more stability), etc. of the company if other criteria remain the same. The table below shows an analysis of the companies accumulated in a cluster based on size (volume of the volume of assets BI_190_1) (the algorithm based the clustering on all parameters; the analysis was conducted based on the volume of assets).

clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	202,479	109,020	319,297	521,742	45,686	67,410	115,668	90,903	217,362	18,326	53,184	45,727
1	323,292	250,233	569,848	836,682	91,308	130,467	215,077	186,630	421,360	17,568	59,752	54,504
2	58,414	44,133	64,355	124,112	3,410	18,066	32,470	18,724	56,763	943	30,585	28,267
3	241,737	125,545	328,041	569,779	52,436	99,852	154,158	101,494	267,362	13,824	53,021	46,453
4	34,822	23,583	43,630	78,344	2,768	9,764	17,945	15,126	38,506	622	23,074	20,430

Table 5: Division of the companies into clusters based on the volume of assets.

The largest companies are in cluster 1 with the average volume of assets amounting to 837,000 euros. The following companies, in clusters 0 and 3, are the so-called medium-sized companies (i.e. rather large medium-sized) with the average volumes of assets amounting to 522,000 euros and 570,000 euros. Clusters 2 and 4 include the companies which may be deemed relatively small.

Comparing table 4 and table 5, we learn that bankruptcies mostly occur among the largest companies, in cluster 1. This is followed by clusters 3 and 0, both based on the number of bankruptcies and size. The number of bankruptcies is the lowest among small companies; there are only a few bankruptcies.

This approach is also supported by practical logic, as small companies do not often reach an official bankruptcy (a lower number of creditors, tighter connections, smaller liabilities, etc.), while large companies always have a creditor who demands an official proceeding (feeling that they are unfairly left out of the distribution of assets otherwise, tax arrears are often involved, etc.).

The table below includes the ratios highlighted above and the balance sheet data on the basis of which the ratios are calculated.

clusters	BI_100_1	BI_190_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	s1	s2	s3	s4	s5	s6
0	202,479	521,742	115,668	90,903	217,362	18,326	53,184	0.39	0.22	0.17	0.42	0.04	0.26
1	323,292	836,682	215,077	186,630	421,360	17,568	59,752	0.39	0.26	0.22	0.50	0.02	0.18
2	58,414	124,112	32,470	18,724	56,763	943	30,585	0.47	0.26	0.15	0.46	0.01	0.52
3	241,737	569,779	154,158	101,494	267,362	13,824	53,021	0.42	0.27	0.18	0.47	0.02	0.22
4	34,822	78,344	17,945	15,126	38,506	622	23,074	0.44	0.23	0.19	0.49	0.01	0.66

Table 6: The ratios calculated based on the balance sheet in clusters.

The table above includes the following ratios:

s1 – BI_100_1/ BI_190_1 – share of current assets in

assets
s2 – BI_290_1/BI_190_1 – share of current liabilities in assets
s3 – BI_310_1/ BI_190_1 – share of non-current liabilities in assets
s4 – BI_370_1/ BI_190_1 – liabilities in assets in total
s5 – BI_400_1/ BI_190_1 – share capital is assets
s6 – BI_40_1/ BI_100_1 – share of cash in current

assets

The summary table below describes the clusters.

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Number of entries of the companies	493,985	463,494	169,205	505,960	149,264
Bankruptcies	237	658	12	471	14
Sizes of the companies based on the	521,742	836,682	124,112	569,779	78,344


volume of assets

Verbal size of the company	Medium-sized large companies	Large companies	Small companies	Medium-sized large companies	Small companies
s1 – Share of current assets in assets	0.39	0.39	0.47	0.42	0.44
s1 comment	Min. value	Min. value	Max. value	Average	Size 2.
s6 – Share of cash in current assets	0.26	0.18	0.52	0.22	0.66
s6 comment	Size 3.	Min. value	Size 2.	Size 4.	Max. value
s4 – total share of liabilities in the assets	0.42	0.50	0.46	0.47	0.49
s4 comment	Min. value	Max. value	Size 4.	Size 3.	Size 2.
s3 – share of long-term liabilities in the assets	0.17	0.22	0.15	0.18	0.19
s3 comment	Size 4.	Max. value	Min. value	Size 3.	Size 2.
s2 – short-term liabilities in the assets	0.22	0.26	0.26	0.27	0.23
s2 comment	Min. value	Size 2-3.	Size 2-3.	Max. value	Size 4.
s5 – share capital in the assets	0.04	0.02	0.01	0.02	0.01
s5 comment	Max. value	Size 2–3.	Min. value	Size 2–3.	Min. value
Descriptive conclusion	Little current assets, a medium amount of cash, few liabilities, a medium amount of non-current liabilities, few current liabilities	Little current assets, little cash, many liabilities, many non-current liabilities, a medium amount of current liabilities	A lot of current assets, a medium amount of cash, a medium amount of liabilities, few non-current liabilities, a medium amount of current liabilities	A medium amount of current assets, a medium amount of cash, a medium amount of liabilities, a medium amount of non-current liabilities, a lot of current liabilities	A medium amount of current assets, a lot of cash, a medium amount of liabilities, a medium amount of non-current liabilities, a medium amount of current liabilities

Table 7: A summarising table describing the result of clustering.

In the context of the table above, it is important that the assessment colours (green, yellow, red) have been used as follows: red is 'poor', yellow is 'acceptable', and green is 'good'. They are not correlated with min. and max. values. For example, the min. value of current assets is poor (thus, min. is red) and the max. value of liabilities is also poor (hence, max. is actually red).

The following can be concluded based on the table above: in the case of smaller companies, the share of current assets in the balance sheet is higher and thus, they have more relative flexibility and relative capability for solving problematic situations. In the case of smaller companies, a relatively large share of their current assets is in cash, which will in turn ensure them more room for manoeuvring from the relative perspective. In the case of the largest companies, the share of their liabilities in the balance sheet is also the highest. The companies which are third by size have the smallest share of liabilities in the balance sheet (medium large – cluster 0). In the case of the remaining companies, the share of liabilities is at the medium level.

Larger companies have more non-current liabilities that others; the balance sheet share of non-current liabilities is directly correlated with the volume of assets, with small shifts. The large share of non-current liabilities indicates dependence on funders (banks), which may create a situation in which the economic activities cannot continue in their previous form, should the economic environment change (up to the banks recalling their loans prematurely and the company going out of business). Non-current liabilities usually largely consist of bank loans and the large share of those liabilities shows, on the other hand, that the company is deemed reliable by banks (i.e. the good economic health of the company, based on which the loans were granted).

Cluster 0 stands out by the lack of current liabilities, also exhibits the minimum value with respect to the volume of all liabilities, and a relatively low volume (second from the bottom) of non-current liabilities, while being third when it comes to the volume of assets. This means that Cluster 0 describes one of the two medium-large groups which stands out clearly by conservativeness when it comes to liabilities (on the other hand, a smaller loan leverage comes with less current assets, as some of the

loan resources is usually inevitably placed in current assets). Another group of medium-large companies (Cluster 3) has a maximum value when it comes to current liabilities, as well as a higher share of current assets (i.e. accumulates debts, takes 'instant loans', and thereby has more liquid resources). Thus, conclusively, there are two groups of medium-large companies which clearly differ from one another based on economic conservatism.

Share capital tends to be an administrative indicator by nature, but being aware of the volume of assets, volume of liabilities, and the share capital, we can find the share of the equity which is not share capital (i.e. retained profit, reserves, etc.). This analysis has not been conducted in the current initial form, but must be taken into consideration upon specifying the model.

In general, table 7 shows that large companies (Cluster 1) are rigid, relatively extensively burdened with liabilities (even though they are reliable thanks to their size and are granted loans), and table 4 also shows that large companies have also experienced more bankruptcies, historically. Other market participants allow large companies to accumulate debts, trust them with payments in instalments, they are trusted by banks, but there is still an inherent rigidity which increases the likelihood of insolvency. The rigid structure is the source of a potential weakness (like a great economic scope may become an issue instead of being an advantage if the market conditions change).

Cluster 0 is a conservative medium-large; this sector has experienced 237 bankruptcies, historically. Another medium-large cluster, Cluster 3, which is an aggressive borrower; this cluster has experienced 471 bankruptcies, historically, almost exactly 50% more. It is a good example, as external resources include an inherent economic risk and the economic risk actually manifests in practice.

Different clusterings have been conducted in the course of the work by using different clustering algorithms. Some further examples of this are provided in Annex 4 to this material.

The figures in Annex 5 present the situation in which reducing twenty parameters to two parameters is presented graphically to be able to display them in a two-dimensional figure. The clusters are marked with different colours; the centroids of the clusters are marked with black x's. This is a technical virtualisation, not an interpretation designed for a (human) end user.

The analysis described above is not conclusive; on the contrary, it is a very primary view of the direction in which the development of the prototype is moving. It was included in this material not for drawing conclusion but to provide an example of how a machine is taught to recognise which company may exhibit signs of so-called economic weakness, which may in turn result in insolvency.

This means that companies are divided into clusters (groups, subgroups, links, etc.) based on economic indicators, activity indicators, etc. A machine is capable of knowing what is 'good' or 'bad' (the colours above), what is influenced by what, what are the threshold values of one cluster or another (thresholds by parameters, mutual combinations thereof) at which risks may arise, how to assess the likelihood of those risks being realised, etc.

The current situation, the baseline of the work is presented above. This is followed by further steps:

1. **additional clustering;**
2. **introduction of the income statement side, assessment of cash flows (currently only balance-sheet based);**
3. **introduction of additional ratios (Annex 2);**
4. **taking the analysis to the sectoral level (four examples in Annex 1);**
5. **defining the outliers of the clusters (in the positive and negative directions);**
6. **finding additional parameters (unsupervised learning).**

1.8. Levels of warning messages and repeat messages

Early warning messages are issued at three levels (the authors of this material believe that it would be reasonable to associate the levels with an equivalent to an insolvency proceeding):

Level 1 – the green level, the weakest level of signalling, the smallest departure from the normal situation;

Level 2 – the yellow level, the medium level of signalling, medium departure from the normal situation;
Level 3 – the red level, the strongest level of signalling, the biggest departure from the normal situation.

Level 1

A level 1 warning is issued to a company if it is clear that the company has departed from a normal situation (the meaning of 'clear' will be defined in the course of further work; it is sector-specific).
A level 1 warning may be baseless; as each company is unique, certain deviations may not actually indicate insolvency or a risk of insolvency. On the other hand, the level 1 warning should, in general, indicate that there are first signs and the solvency of the company may decline if the situation deteriorates.

Level 2

A level 2 warning is issued to a company when its solvency has already decreased and there are justified concerns (the development dynamics indicates that the solvency of the company is dropping systematically) that the solvency may drop to the point where the company will be subjected to the bankruptcy proceeding (the meaning of 'justified concerns' will be defined in the course of further work; it is sector-specific). The company has temporary payment difficulties and, if nothing is done, there is a risk of permanent insolvency. A level 2 warning indicates that measures must be taken immediately; it is not simply a false alarm or a subjective conservative assessment. One of the potential solutions for fixing the situation would be initiating a reorganisation proceeding.

Level 3

A level 3 warning is issued to a company if there is a likely risk of the company having to file an application for bankruptcy (potentially the situation of decrease of assets referred to in § 176 or § 301 of the Commercial Code).
The aim of a level 3 warning is to draw the attention of the management board of the company to the fact that the company is probably (as far as it is possible for a computer to determine) in the situation defined in subsection 1(2) or subsection 1(3) of the Bankruptcy Act and the management board is required to implement the measures specified in subsection 180(5^{1}) or subsection 306(3^{1}) of the Commercial Code.

Pursuant to 180 ($5^{1)}$ of the Commercial Code, if a private limited company is insolvent and the insolvency, due to the company's economic situation, is not temporary, the management board shall promptly but not later than within twenty days after the date on which the insolvency became evident, submit the bankruptcy petition of the private limited company to a court. (Subsection 306 ($3^{1)}$ of the Commercial Code establishes similar obligations to public limited companies).

Subsection 1 (2) of the Bankruptcy Act explains that a debtor is insolvent if the debtor is unable to satisfy the claim of a creditor that has fallen due and such inability, due to the debtor's financial situation, is not temporary. Subsection 1 (3) of the Bankruptcy Act adds the definition of the insolvency of a debtor who is a legal person, stating that this person is insolvent if the assets of the debtor are insufficient for covering the obligations thereof and such insufficiency is not temporary.

Repeat early warning message

The system is programmed to repeat the early warning message. Repetition is necessary, as the aim is to achieve a situation in which the company would implement measures, having received a warning signal.

The algorithm issues a new warning message on two occasions:
(a) the system detects that the warning level has deteriorated;
(b) six months have passed since the last warning and the situation has not changed.

Content of the message

An early warning message contains information on why the warning of a certain level was issued and why the system believes that the solvency is about to deteriorate or has already deteriorated. The content of the message must form the basis for the further action plan of the company for improving its solvency (i.e. the information on the circumstances of the decrease in the solvency must be sufficiently accurate to be useful).

1.9. Inspection of the appropriateness of the data

The early warning system must be developed based on the assumption that high-quality data are used, only the most accurate data are selected, and, in the event of linking data, the linking must be reasonable and appropriate. From the practical perspective, this means that the data used as input must be checked regularly and different input solutions must be tested (thereby assessing the appropriateness of the data). The prototype created must be based on the most rational and accurate use of input data with the parameters generating excessive data and 'data noise' excluded, if possible.

Perboli and Arabnezhad (2021) stress that in order to obtain an adequate outcome from machine-learning, it is important to first choose the right indicators to analyse, and suggest their own points of origin for this. This approach is one of the points of origin used for checking the appropriateness of the data in this work. The source cited is just one of the many which stress the necessity of the right preparation of data and the right indicators to analyse.

Perboli and Arabnezhad (2021) present in their approach the figure below which strongly focusses on the selection of indicators and on the initial cleaning/linking of data. The further machine learning tuning is also important, i.e. finding further parameters and determining their threshold values.

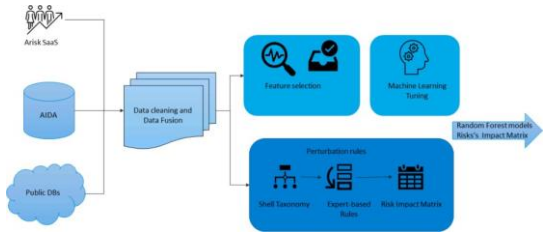


Figure 2. The importance of the selection of input indicators.

The figure above shows that the linking and prior cleaning of data have a very important role, but decisions are also made on which parameters to include and which not to include. After this stage, the phase of machine-learning may be entered.

The parameters included in the model by Perboli and Arabnezhad (2021) are presented in the table below. The table shows that on the one hand, cash flows have been focussed on (the profitable cash flow); on the other hand, however, the liabilities of the company. Production capacity has also been included, but mainly as an additional parameter.

Data set features.

Feature	Feature value	Feature type
ATT10	Absolute value	Revenue/profit
ATT11	Index/percentage (%)	Revenue/profit
ATT12	Absolute value	Revenue/profit
ATT13	Absolute value	Revenue/profit
ATT14	Index/percentage (%)	Revenue/profit
ATT15	Index/percentage (%)	Cost/debt
ATT16	Absolute value	Cost/debt
ATT17	Index/percentage (%)	Cost/debt
ATT18	Absolute value	Cost/debt
ATT19	Index/percentage (%)	Cost/debt
ATT20	Absolute value	Production
ATT21	Absolute value	Production
ATT22	Index/percentage (%)	Revenue/profit
ATT23	Absolute value	Production
ATT24	Index/percentage (%)	Cost/debt

Table 8: The parameters suggested by Perboli and Arabnezhad (2021).

The approach above has enabled the model of Perboli and Arabnezhad (2021) to examine the likelihood of bankruptcy with respect to areas of activity, turnover, location, COVID-19, support measures/loans; see the figure below.

Italian companies bankruptcy with regard to the activity.

Risk of bankruptcy	Activity	Count	Prob $< 50\%$	$50\% \leqslant$ Prob $< 70\%$	Prob $\geqslant 70\%$
Short-term	Industry	69,351	58%	18%	24%
	Commerce	46,524	56%	19%	25%
	Public	3,630	51%	23%	26%
	Service	42,097	47%	21%	32%
multirow420ptMiddle-term	Industry	69,351	53%	19%	28%
	Commerce	46,524	51%	21%	28%
	Public	3,630	48%	23%	29%
	Service	42,097	43%	22%	35%
Long-term	Industry	69,351	52%	19%	29%
	Commerce	46,524	50%	22%	29%
	Public	3,630	46%	23%	31%
	Service	42,097	42%	22%	36%

Italian companies bankruptcy with regard to company revenues (millions of euros).

Risk of bankruptcy	Revenue	Count	Prob $<$ 50%	$50\% \leqslant$ Prob $<$ 70%	Prob $\geqslant$ 70%
Short-term	< 5	127,009	51%	20%	29%
	$5 \leqslant X < 10$	17,965	64%	18%	18%
	$10 \leqslant X <$ 15	6,519	65%	16%	19%
	$\geqslant 15$	10,109	65%	16%	19%
Middle-term	< 5	127,009	47%	21%	32%
	$5 \leqslant X < 10$	17,965	60%	19%	21%
	$10 \leqslant X <$ 15	6,519	61%	17%	22%
	$\geqslant 15$	10,109	60%	18%	22%
Long-term	< 5	127,009	46%	22%	32%
	$5 \leqslant X < 10$	17,965	60%	18%	22%
	$10 \leqslant X <$ 15	6,519	61%	17%	22%
	$\geqslant 15$	10,109	59%	18%	23%

Italian companies bankruptcy with regard to company location.

Risk of bankruptcy	Location	Count	Prob $<$ 50%	$50\% \leqslant$ Prob $<$ 70%	Prob $\geqslant$ 70%
Short-term	Northeast	39,775	60%	18%	22%
	Northwest	53,045	57%	18%	25%
	Center	36,724	49%	20%	31%
	South	32,058	48%	22%	30%
Middle-term	Northeast	39,775	56%	19%	25%
	Northwest	53,045	53%	20%	27%
	Center	36,724	44%	22%	34%
	South	32,058	44%	23%	34%
Long-term	Northeast	39,775	55%	19%	26%
	Northwest	53,045	52%	20%	28%
	Center	36,724	43%	22%	35%
	South	32,058	42%	23%	35%

The bankruptcy of Piedmont companies pre- and post-COVID-19, as well as after the financial support policy (at 10%, 20%, or 30% of a company's past-year revenues).

Risk of bankruptcy	Prob. $<50\%$	$50\% \leqslant$Prob. $<$ 70%	Prob. $\geqslant 70\%$	Mean risk
Pre-COVID-19	70.7 %	28.7 %	0.6 %	29 %
Post-COVID-19	13.8 %	84.6 %	1.6%	39%
Loan 10% of revenues	15.8%	82.7%	1.5 %	40 %
Loan 20% of revenues	27.7 %	71.5%	0.7 %	33 %
Loan 30% of revenues	20.1%	79.0 %	0.9%	38%

Figure 3. The likelihood of bankruptcy by different contexts.

The outcomes of the above include, for example, which potential inputs may lead to different outcomes in the context of the appropriateness of the data or methods. In this work, different methods are used to cluster different data

and an optimum final outcome is sought to ensure an assessment on insolvency in different contexts (sectors) and by different parameters (the indicators characterising the company). The analytical experiences of other countries are very important in this context and are included in this material to show what the work is based on.

The appropriateness of the data will be checked in this work systematically based on which parameters are chosen and how they will function together.

1.10. Improving the quality of input parameters

Depending on how the system generates the initial results, further parameters must be included in the system; if possible, those parameters should be focussed on increasing the accuracy.

Andrés et al. (2004) highlight some parameters which help to define in further detail the quality of business operations, see the figure below.

Dimension	Variable	Code
Debt quality	$\frac{Current\ Liabilities}{Total\ Debt}$	V1
Indebtedness	$\frac{Equity\ Capital}{Total\ Debt}$	V2
Use of fixed capital	$\frac{Tangible\ Fixed\ Assets\ +\ Intangible\ Fixed\ Assets}{Total\ Employment}$	V3
Debt cost	$\frac{Financial\ Expenses}{Total\ Debt}$	V4
Short-term liquidity	$\frac{Current\ Assets}{Current\ Debt}$	V5
Share of labour costs	$\frac{Labour\ Cost}{Added\ Value}$	V6
Size	*Net Sales (EUR thousands)*	V7
Average sales per employee	$\frac{Net\ Sales\ (EURthousands)}{Total\ Employment}$	V8

Figure 4. The parameters of Andrés et al. (2004) which increase analytical accuracy.

From the figure above, it would be important to highlight parameter V1, for example, which shows the quality of liabilities, i.e. how much loans, including instant loans the company has taken without second thought (Current Liabilities) and the share of those liabilities in the total liabilities (other components of liabilities include the high-quality loans from banks). This parameter helps to assess how randomly loans are taken and how dangerous the loans may become in the case of one scenario or another. In this work, it was calculated based on the assumption that clustering is always based on current and non-current liabilities separately and the respective structure ratios were examined, and thus, preliminary work is being done in this direction, but further 'refining' is required.

Thereat, it is important to also stress the parameters V1 and V2 which include the dimension of human capital, show how many employees are involved (V6), and the actual productivity of the employees involved (V8).

The approach of Aliaj et al. (2020) is a good example of adding further specifying high-quality parameters. They approach this issue based on the balance sheet of a bank and thus it cannot directly be 'copied', but the principle and perspective are important (see the figure below).

ID	Description	ID	Description
L1	Granted amount of loans	**B1**	Revenues
L2	Used amount of loans	**B2**	ROE
L3	Bank's classification of firm	**B3**	ROA
L4	Average amount of loan used	**B5**	Total turnover
L5	Overdraft	**B6**	Total assets
L6	Margins	**B7**	Financial charges/operating margin
L7	Past due (loans not returned after the deadline)	**B8**	EBITDA
L8	Amount of problematic loans		
L9	Amount of non-performing loans		
L10	Amount of loans protected by a collateral		
L11	Value of the protection		
L12	Amount of forborne credit		

Table 1. Main attributes for the loan (L) and the balance-sheet (B) datasets.

Figure 5. The opportunities for quality parameters of Aliaj et al. (2020).

For example, parameters L3 (how the company is qualified), L4 (how much loan has actually been used), L6 (margins), L7 (previous debts). On the one hand, the parameters are technical, but on the other hand, they are quality parameters (for example, previous debts are not current technical indicators, but indicators on which assessments are based).

1.11. Finding further parameters

In order to refine the approach, further parameters must be introduced which would make a significant contribution. There are two possibilities for finding further parameters:

- examining the results of the work done enables concluding in which direction it would be reasonable to move and what to add;
- using the unsupervised learning machine learning method in which the machine checks without further instructions which parameters could be added.

Choudhry (2018) shows how supervised and unsupervised machine learning can be used and which techniques are suitable for this purpose (see the figure below).

Exhibit 2 Machine learning problem types classification

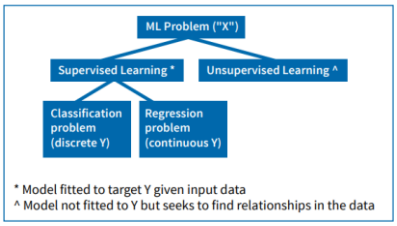


Exhibit 3 Machine learning models

Type	Techniques
Supervised Learning (Target Y given)	Decision trees Artificial neural networks Support vector machines k-nearest neighbours (non-parametric) Random forests (non-parametric) Naïve Bayes
Unsupervised Learning	k-means clustering Hierarchical clustering analysis
Recommender systems	Blend of different methodologies

Figure 6. Deployment of machine learning based on Choudhry (2018).

The left image of the figure above shows which methods are advisable to use in the case of certain types of machine learning. The k-means technique was used in this work in the context of supervised learning which is not presented in the figure. However, the figure is not designed to serve as absolute guidelines, but as one potential solution.

1.12. Plurality of parameters

In the case of adding further parameters, it is important to correctly determine the number of clusters (which was already explained above). A higher number of parameters may come with the desire to define more clusters.

However, it is important to keep in mind that the clusters must be given substance, which is difficult if the clusters are too similar. In this case, five clusters were used even in the case of a higher number of parameters

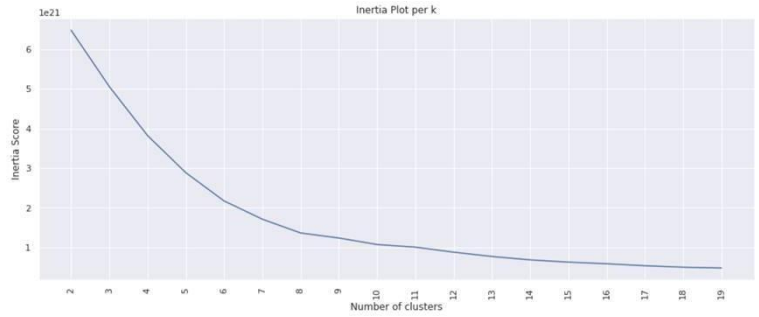


Figure 7. Determining the number of clusters.

At the time of conducting the work, the balance sheet dataset includes approximately 150 different identifiers. The higher number of identifiers makes the calculations complicated. PCA (Principal Component Analysis) methodology was used to solve the problem – it is a technique of lowering dimensions which ties the variables included in the correlation to a low number of non-correlating variables (the main variables). As a result, maximum information is obtained from a low number of compressed data. The work done within the framework of this project has managed to take the number of compressed variables to 7, as is indicated in the figure below.

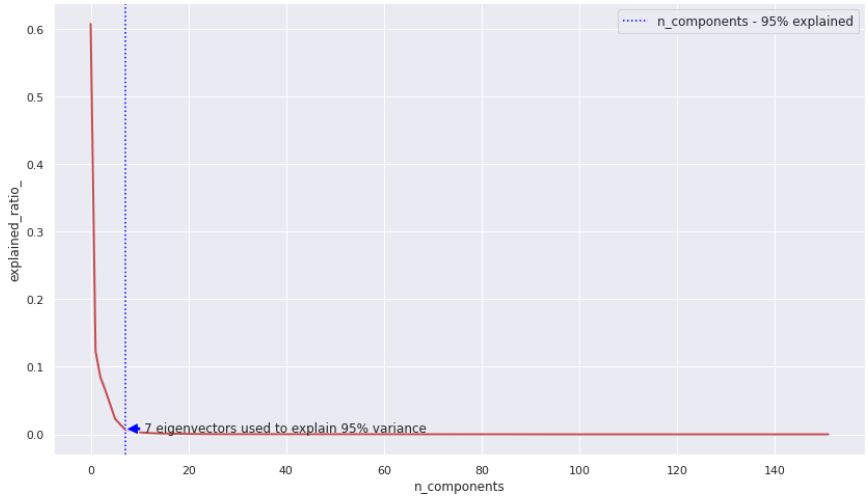


Figure 8. Reducing the number of variables to seven main parameters.

The seven parameters obtained by the PCA technique were turned into a two-dimensional figure presented in Annex 6. It had to be made two-dimensional as this enables presenting it in a figure, it is not possible to draw a seven-dimensional figure. The parameters displayed on the axes are mathematical;

the content thereof cannot be directly interpreted from the economic perspective. On the other hand, this enables determining the clusters and by analysing the content of the clusters (the sets of economic operators/entries of economic operators deemed similar in each cluster), they can be attributed economic descriptions like the previously attributed descriptions in table 7.

As explained, correlation is very important in using the PCA technique. The correlation matrix drawn up in the course of this project is provided in Annex 7. The correlation matrix only includes the most important variables, not all 150 variables which can be obtained from the balance sheet dataset. The matrix is provided for illustrative purposes in this case.

1.13. Assessment based on B2B transactions

The above describes a set of solutions which is based on a classic analysis of the balance sheet and income statement and the related data. Another approach (as mentioned before) would be to assess the solvency of an economic operator merely based on B2B transaction information (both approaches are actually combined in this project, but the initial processing differs). The work with the approach of transactions is only beginning in this project and thus, there are no technical outcomes at this point.

Kou et al. (2021) suggest a bankruptcy prediction model for small and medium-sized companies which is based on the data of the transactions between those companies. This is an alternative solution to the models based on an analysis of the economic ratios of the accounting data of a company (balance sheet, income statement) and the analysis of so-called classical ratios is left aside completely. The authors of the methodology claim that B2B transaction information is more reliable than accounting reports.

Kou et al. (2021) illustrate their approach in the following algorithmic scheme; see the figure below. This work is based on the aforementioned transaction-based model, but it is not copied in detail.

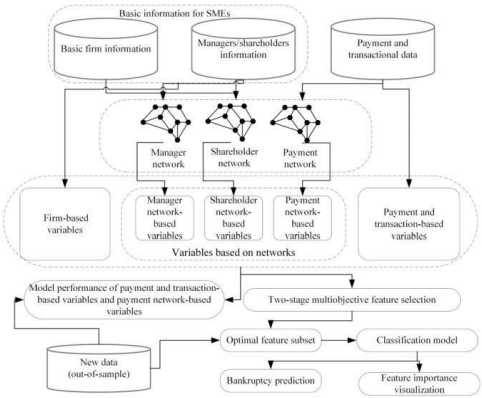


Figure 9. The transaction-based model of Kou et al. (2021).

The issue with a solvency prediction methodology which is only based on analysing transactions is how to obtain enough variables to build machine-learning based on it. The solution of Kou et al. (2021) illustrated below is to use turnovers as well as estimated profits from transactions, counting the both the number and value of the transactions, using different intervals of days, and adding volatility (risk) with an external variable.

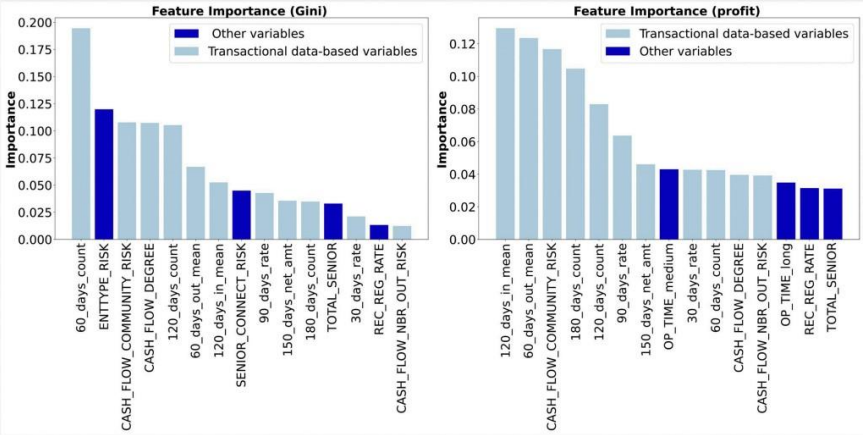


Fig. 6. Feature importance for features in an optimal feature subset.

Figure 10. The solution of Kou et al. (2021) for transaction variables.

1.14. Data

This work is mainly based on date from the annual accounts of economic operators (balance sheet and income statement analysis, in general, over five years, if possible), which are supported by the data of KMD and KMD INF value added tax returns (for transaction data). The data of the TSD income and social tax returns is also included, supplemented with the data from the employment register (to increase the quality of the data, see Figure 4 above). The data from the annual reports of the companies which have gone bankrupt are used (five years before the bankruptcy was declared). If possible (work is ongoing to make it possible), information on tax arrears is included. The list provided here may not be final; the data focus is becoming increasingly more specific in the course of the work.

The sectors are determined based on the EMTAK database, as specified in the class diagram below:

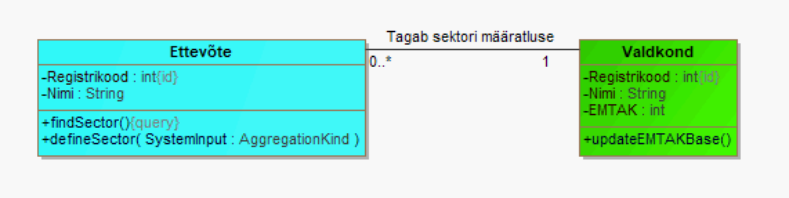


Figure 11. Determining of sectors.

The figure above shows how the sector is determined by using the EMTAK code. As explained above in this document and demonstrated in Annex 1, five-digit EMTAK codes are selected to combine industrial sectors/sectors. This is completed by the
+updateEMTAKBase() procedure.

In the case of using transaction data, the data from the annual reports of economic operators is also used in addition to the KMD INF and KMD data, as specified in the class diagram below. It is important to compare the alignment of the turnovers and to be able to connect a long-term view (annual reports) with short-term operational datasets (the KMD and KMD INF are available on a monthly basis).

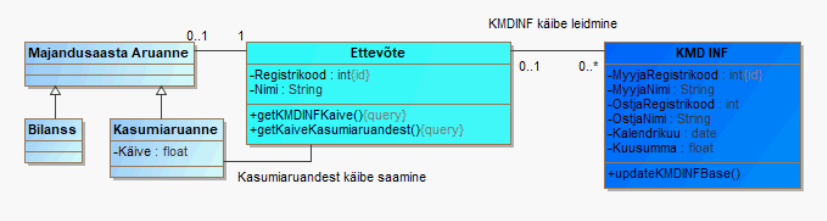


Figure 12. Connecting the KMD INF data with annual report data.

The figure above shows how the turnover from the KMD INF database is used on the one hand (inquiry: +getKMDINFKaive()) and the turnover from the income statement of the annual report of the economic operator on the other hand (+getKaiveKasumiaruandest()). It is irrelevant in the context of the figure above, but the KMD INF data are actually also supported with the data from KMD returns.

The process scheme for connecting the KMD INF data and the data from annual reports.

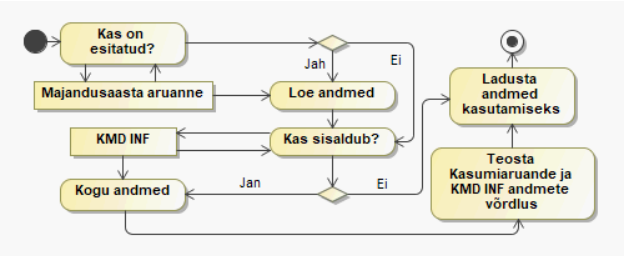


Figure 13. The process scheme for connecting the data from annual reports with the KMD INF data.

The process scheme above confirms the above, but it should be stressed that the KMD INF and income statement data are compared.

The algorithmic description of the aforementioned

cheme is provided below.

```
While Ettevõte.Registrikood □ 0 do
        For every Ettevõte.Registrikood do
                IF Ettevõte.Registrikood contains in Majandusaasta Aruanne
                        then get Kasumiaruanne.Käive
                else
                        Kasumiaruanne.Käive = Missing
                end
                IF Ettevõte.Registrikood □ KMD INF
                        then getKMDINFKaive()
                else
                        getKMDINKaive() = Missing
                end
        compare getKMDINFKaive() to Kasumiaruanne.Käive
        end
end
```

The profit of the company is also taken into consideration through the annual report.

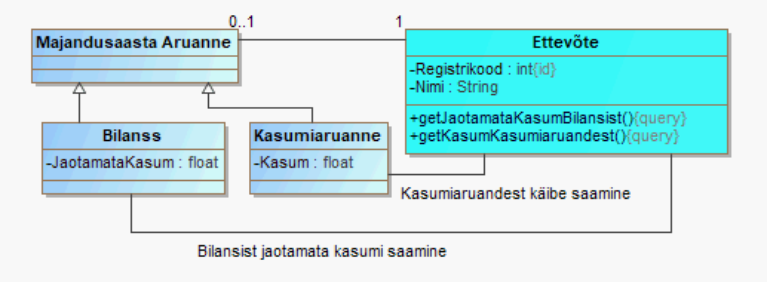


Figure 14. Profit based on the annual report.

Based on the process scheme, profit is checked routinely from annual reports, but retained profits are also highlighted – this provides a further temporal perspective for how the profit has formed and what is the dynamics of generating profit (payment of dividends is naturally also taken into consideration).

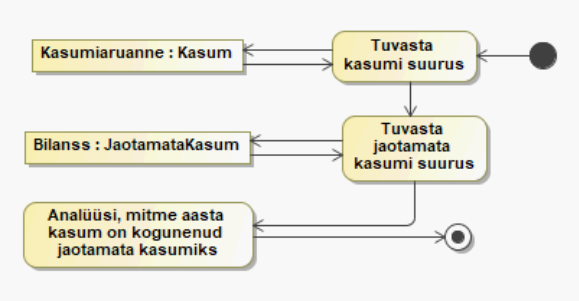


Figure 15. The process scheme for analysing

profit. The algorithmic scheme for analysing profit.

```
For every Ettevõte.Registrikood do
        get Kasumiaruanne.Kasum
        get Bilanss.JaotamataKasum
        this.KasumlikudAastad=Bilanss.JaotamataKasum/Kasumiaruanne.Kasum KasumiAnalüüs
        = {{KasumlikudAastad},{Bilanss.JaotamataKasum},
                {Kasumiaruanne.Kasum}}
end
```

The change in equity is also analysed as an important input.

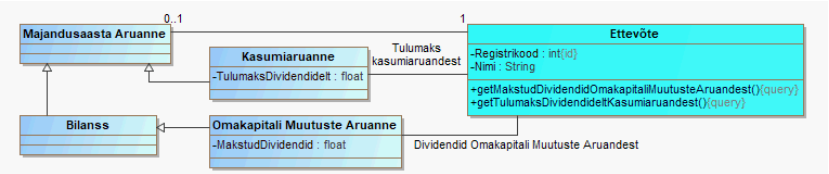


Figure 16. Analysing the change in equity.

For the purposes of an in-depth analysis, the nature of the assets of the company must also be assessed and it is described in the class diagram below. The quality of assets may significantly impact whether and how strongly the company can fight its potential insolvency. For example, some property investments may be highly liquid.

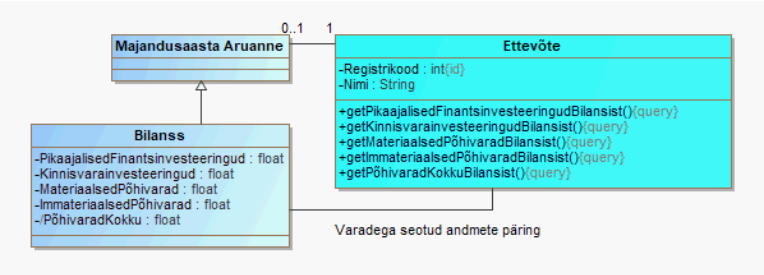


Figure 17. Analysis of the quality of assets.

Liquidity is very important in the context of solvency and payment difficulties.

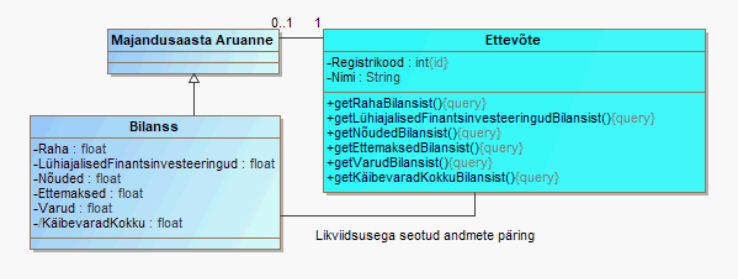


Figure 18. The data of the liquidity of a company.

From the perspective of solvency, the analysis of liabilities described in the class diagram below is important.

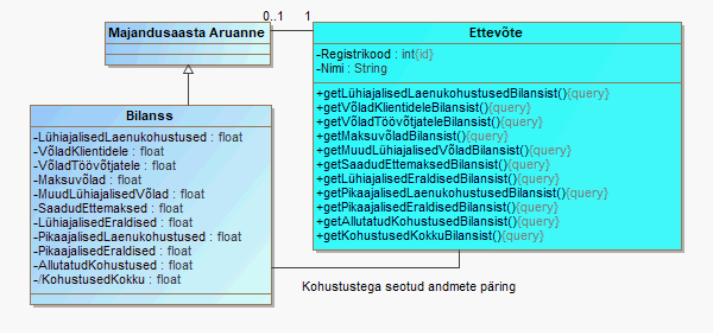


Figure 19. Analysis of the liabilities of the company.

Equity is also analysed by components (not only as changes as specified above). Equity must be analysed to get a better overview of the development dynamics and activities of the company; the so-called surplus money is accumulated under equity.

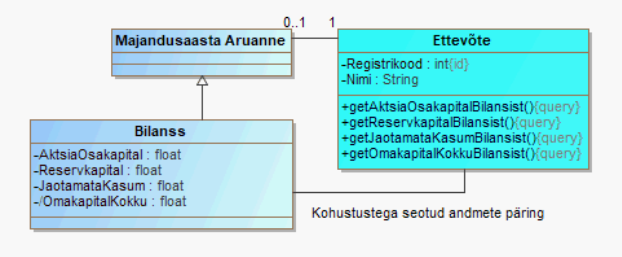


Figure 20. Analysis of the equity of the company.

As specified in the figure above, the equity analysis is viewed through share capital, legal reserve, and retained profits. The outcome of the economic activity (positive) is accumulated on the line of retained profit (payment or non-payment of dividends must be monitored) and monitoring the accumulation of this outcome provides an overview of the dynamics of the operations.

Keeping in mind that annual reports are generated at certain intervals, the generation of retained profits can also be modelled based on the monthly data from tax returns to obtain a more operative overview (being aware of the previous dynamics).

Introducing the labour data is presented in the class diagram below.

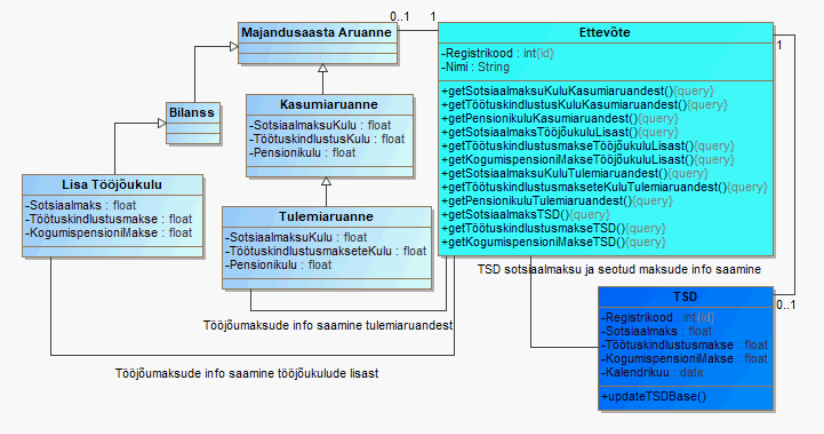


Figure 21. Input to the analysis of labour costs.

The figure above shows that the main sources of the data on labour costs include the note to the annual report on labour costs, the income statement, and the TSD tax return from the tax returns used. The TSD tax returns enable supporting the information of the annual reports which is generated at certain longer intervals in a more operative temporal interval.

The class diagram below describes introducing employment register data to add labour data for the purposes of increasing the data quality.

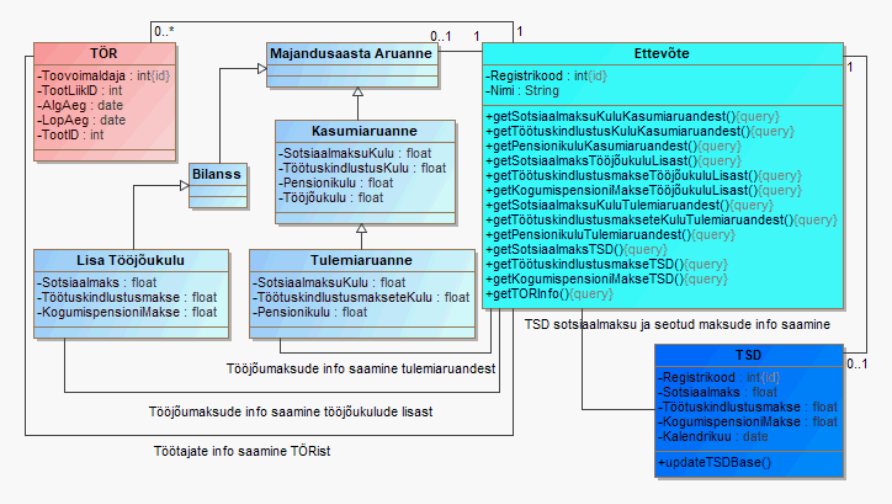


Figure 22. Introduction of the employment register data.

The class diagrams described above are those which are complete in the current phase of the work and based on which an analysis was or is being conducted are presented. New class diagrams (like process schemes) are added in the course of the work and those diagrams are more specific. The above was provided with an aim of providing the best possible overview of the current situation of the work.

Chapter II – practical execution

The chapter presents the circumstances related to the practical execution conducted within the framework of the work. The analytical logic is presented which the practical execution is based on, with examples given on the outcomes and codes/outputs. Not all tests completed, or models and clustered clusters are presented, only the examples with the most significant illustrative points. On the other hand, the examples and material provided were selected to provide a substantial overview of what is going on and of the directions of the work.

The practical execution strongly focusses on clustering, which is basically unsupervised machine-learning. The aim is to allow the machine-learning algorithms to seek patterns which may indicate potential insolvency and the associations related to the elements of insolvency in general.

Models have been created based on the outcomes of clustered data and machines have been taught to recognise different patterns and draw conclusions. Above all, balance sheet figures, the ratios of those figures, and the data on B2B transactions were used. In the course of further development of the models (not yet included in this documents), there are plans for involving labour data and the datasets defining the general economic background.

Reading this chapter, it is important to keep in mind that the product in question is still being worked on; some parts of it will developed further; in the case of others, however, the currently known baseline positions are provided.

2.1. Principles of the structure of the system

The principle process scheme of constructing the system is provided below.

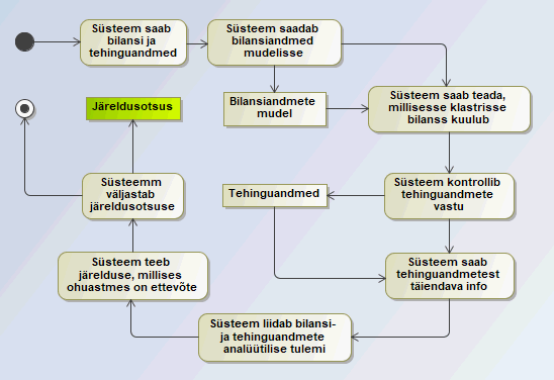


Figure 2.1.a.: The process scheme of constructing the system.

The above shows how the system must be built, based on the conceptual logic. This material also explains and illustrates the process of creating the system, as a prototype is being developed and prototypes must be as open for analysis as possible. Therefore, a schedule for creating the prototype is provided, as well as the steps taken, and the lessons learned in the course of taking the steps (setbacks, conclusions, improvements).

At the level of the prototype, the system is designed to exist as a functioning machine-learning based algorithm, i.e. it is not designed to have a user interface for the external user. On the other hand, in the course of the work, in order to enable different tests, an opinion has begun to form that a web-based user interface would facilitate testing the system and allow the stakeholders to get a better grasp of the system.

2.2. Analysis of balance sheet data

2.2.1. Clustering of balance sheet data

The figure below presents the data in the context of balance sheet volume (x) and total liabilities (y). The figure is currently limited to 1 million.

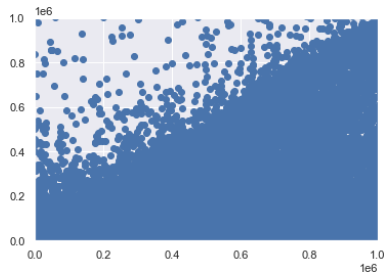


Figure 2.2.1.a.: Balance sheet data (x) vs. total liabilities (y).

The figures below present the results of the clustering of balance sheet data in the form of balance sheet vs. total liabilities and balance sheet vs. current assets. The figures are limited to 100 million and 10 million. Thus, the amounts in the figures above are higher than those in the figures below.

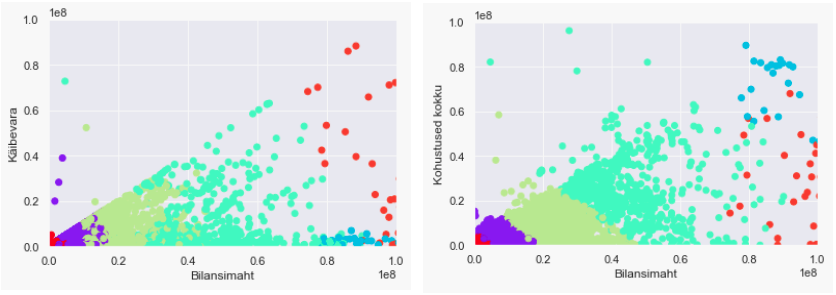


Figure 2.2.1.b.: Clustering of balance sheet data (up to the limit of 100 million)

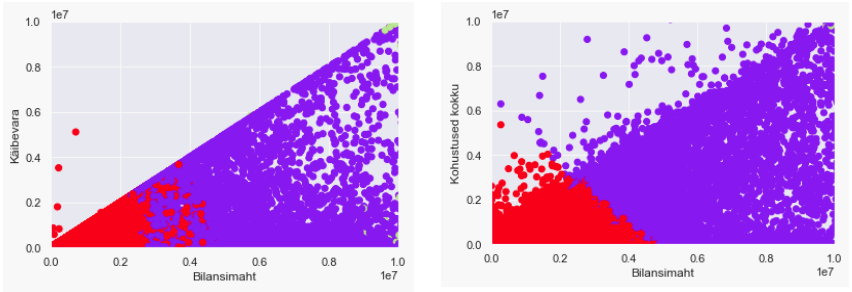


Figure 2.2.1.c.: Clustering of balance sheet data (up to the limit of 10 million)

The figure above shows that the clusters are clearly formed based on size (taking into consideration all parameters, but the size is most important).

The analysis above included the following of balance sheet entries:
- current assets;
- total short-term loans;
- total current liabilities;
- total non-current liabilities;
- total liabilities;
- total balance sheet.

This initial selection was made for the following reasons:
(a) the balance sheet volume provides an initial overview of the size of the company, which can be interpreted as a market force to a certain extent (understandably depending on the accompanying circumstances); size is correlated with a certain inertia (it is harder to shake a bigger ship, but if a big ship is going down, nothing can be done to save it);
(b) liabilities show how many loans the company has taken, to which extent the company is actually owned by its owners; loans may be of different contents and so-called 'risk levels', but their share indicates in general the extent to which the company is operating based on loans; the dynamics of the loans shows whether the operating based on loans has increased or decreased (increasing loans may indicate that the company is incapable of coping on its own – the creditors must help it survive);
(c) Separating current and non-current liabilities shows the level of risk for the short-term perspective of the company due to debts (non-current liabilities are more stable, current liabilities may become collectable soon, may affect the company in the short-term perspective); non-current liabilities (primarily long-term loans, but non-current liabilities mainly costs of long-term loans) show the trust of long-term lenders (primarily banks) for the company; this trust is the outcome of analyses; the ratio of current and non-current liabilities shows, among other things, stability in the longer perspective;
(d) short-term loans show the share of current liabilities which have been taken to fund the current activities of the company, mainly to cover for insufficient operating capital; if a company has to take short-term loans (may be compared to instant loans, in a way), it may indicate their path towards insolvency; the system monitors the share of short-term loans in current liabilities, as well as the amounts of those loans and the changes in the loans in time;
(e) current assets are liquid assets which can be used by the company for funding, if necessary (also used for current funding); the company may be big and have a strong market position, but it may still find itself in financial difficulties if there are no current assets.

The clustering performed based on the indicators above showed (the illustration provided above) that a machine can differentiate different companies and decide which cluster one or the other should be included in, but the clusters were primarily formed based on size. Large and small companies understandably have different economic dynamics and differentiating by size is therefore justified, but a more accurate perspective is required for a more adequate assessment.

Based on the above, the following indicators were added to balance sheet data:
- cash;
- advance payments;
- fixed assets.

The further indicators added help to specify the results of clustering based on the following logic:

(a) cash basically shows the current capability of the company to cover its liabilities; figuratively, it is like cash in hand which can be used to making current payments – this is a further specification of current assets;
(b) an advance payment is an asset of the company; on the one hand, the number of advance payments shows how many liabilities have been covered in advance, which allows assessing the actual amount of the liabilities (which sometimes appear smaller due to advance payments); on the other hand, they indicate the financial capability of the company (so-called 'money paid out' is part of the capability of the company);
(c) fixed assets show in the analysis how the company has invested its funds and organised its activity; under certain conditions, the lack of fixed assets indicates instability of the company (all of the cash used is spent on supplies, i.e. is directly written off and no large machinery is purchased); the dynamics of fixed assets in time are important (shrinking of the fixed assets may indicate that the company is fading away or is incapable of updating its technology – the company starts to buy semi-finished products, not being capable of manufacturing).

The illustrations below present the outcomes of the balance sheet data clustered by using further balance sheet parameters.

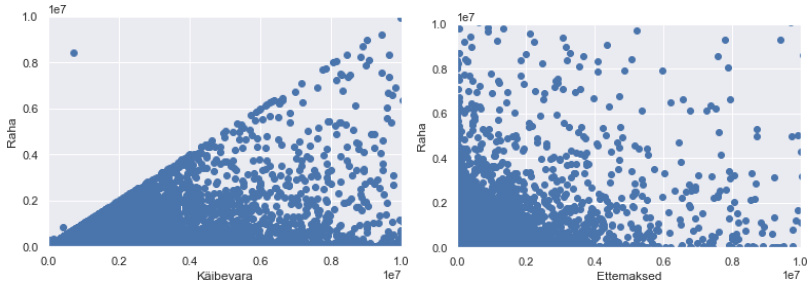


Figure 2.2.1.d.: Cash vs. current assets and cash vs. advance payments.

The figures above show that those with more current assets also have more cash (in the case of a smaller balance sheet volume). In the case of larger companies, more current assets do not mean that the current assets are largely in cash; yet, there are also those who interpret current assets as cash.

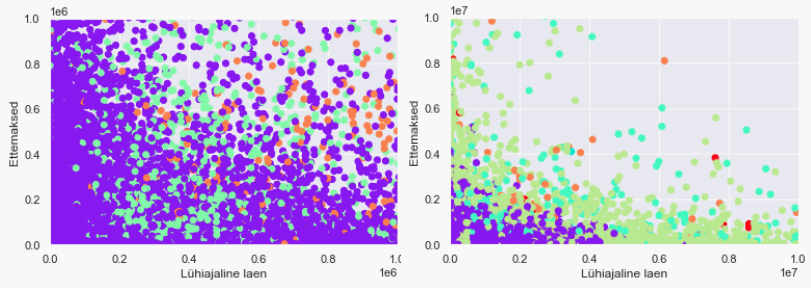


Figure 2.2.1.e.: Clustering of balance sheet data vs. short-term loans.

The figures above (the results of clustering) clearly shown that if there are more current loans, there are fewer advance payments and if you make an advance payment, you will not use a current loan for this.

The system highlights very different clusters, but this pattern is significant and telling (yet very logical) for the machine to be able to assess the impact of short-term loans and advance payments on the solvency of the company.

The two figures above are extracts from the same figure in different scales.

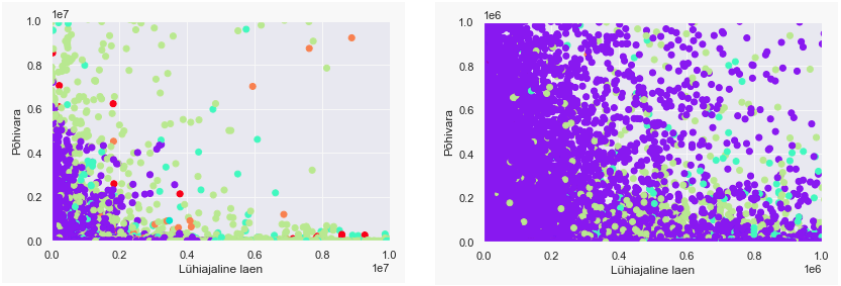


Figure 2.2.1.f.: Clustering of balance sheet fixed assets vs. short-term loans.

The figure above clearly shows the connections between current liabilities and fixed assets, it is evident that those with more current liabilities have less fixed assets:
(a) Those companies which are dependent on current liabilities (i.e. rapid load addicts) have no functional capability for acquiring fixed assets in a considerable extent;
(b) The companies which are addicted to current assets have realised their fixed assets; in the case of this option, it is especially important to monitor the dynamics of fixed assets over years.

As having fixed assets and the dynamics (liquidation) of these acquisitions show the economic foundation of a company, fixed assets have also been assessed against other indicators.

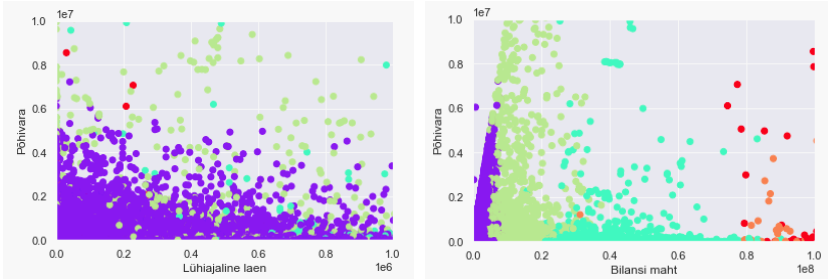


Figure 2.2.1.g.: Clustering of balance sheet fixed assets vs. short-term loans, vs. balance sheet volume. The drawings above also include the outcome of the clustering of fixed assets vs. short-term loans (i.e. the economic feasibility and sustainability vs. dependence on instant loans) in a complementary scale and the fixed assets vs. balance sheet volume on the right.

When viewing the links between the balance sheet volume and fixed assets in the context of clustering outcomes, clearly differentiating clusters can be seen, as well as correlation patterns between fixed assets and balance sheet volume. In general, the balance sheet volume indicates the size of the company, but the balance sheet structures are different in different size ranges (in the case of very large balance sheet volumes, the share of fixed assets with respect to the balance sheet volume is mostly not high).

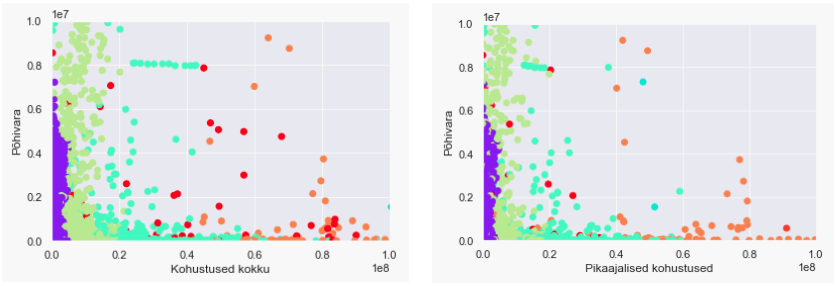


Figure 2.2.1.h.: Clustering of balance sheet data fixed assets vs. non-current liabilities vs. obligations in total.

The figure above shows that fixed assets have a specific pattern (form clear clusters), total liabilities, and total non-current liabilities. We can clearly see the dynamics of more fixed assets having a positive correlation with larger liabilities (for example, a long-term loan is taken; this is used to buy fixed assets and develop business). There is, however, the dynamic of being in trouble with coping with very extensive liabilities and, due to this or by attempting to solve the situation, the share of fixed assets is minimum (this correlation is certainly problematic from the perspective of solvency).

The figures below examine fixed assets vs. current liabilities in the context of clustering outcomes. Fixed assets vs. short-term loans were examined above. Short-term loans and current liabilities may largely overlap (however, this situation would rather indicate potential insolvency, as it is an instant loan or equivalent for keeping the company afloat), but may also not overlap (the current liabilities arising from the structure of the economic activity which do not indicate insolvency in any way). Thus, as fixed assets have turned out to be a relatively good indicator, it is important to analyse them against short-term liabilities as well.

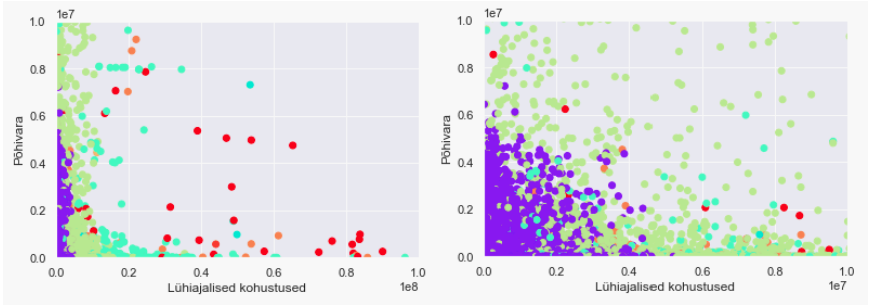


Figure 2.2.1.i.: Clustering of balance sheet data fixed assets vs. current liabilities.

The figures above show that current liabilities have specifically distinguishable cluster patterns with respect to fixed assets. There are clusters with less current liabilities, which means that there is clearly more fixed assets, and there are cluster patterns in which the company basically 'chooses' whether to keep its funds in the fixed assets or let the amount of current liabilities in the balance sheet grow (the so-called OR pattern; it would be more logical to use the AND pattern in this issue where fixed assets can be 'pumped' by letting current liabilities grow, i.e. to place short-term funds in longer-term investments, which is usually a big mistake).

In general, it may be stated that introducing additional parameters to the balance sheet was justified. Adding fixed assets, cash, and advance payments to the model improves it significantly – very clear patterns may be highlighted to draw conclusions.

2.2.2. Clustering of ratios

The financial condition of the company can be described and analysed via different balance sheet ratios. The ratios used in the work are described in Annex 2 to this material. From the perspective of machine learning, it would be important to choose the most important ratios for initial testing and add other later, if necessary. The following rations were selected for primary clustering:

s1 – share of current assets in assets;
s2 – current liabilities in assets;
s3 – non-current liabilities in assets;
s4 – liabilities in total in assets;
s5 – share capital in assets;
s6 – share of cash in assets

The figure below illustratively presents the correlations between the ratios.

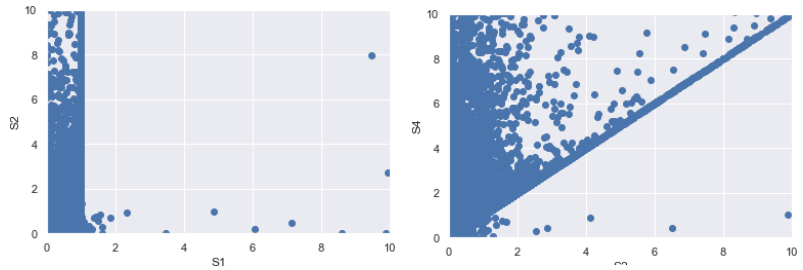


Figure 2.2.2.a.: The ratios used, S1 vs. S2, S3 vs. S4.

The figure below shows the results of the clustering of the ratios. Clustering based on KMeans(12) was currently used.

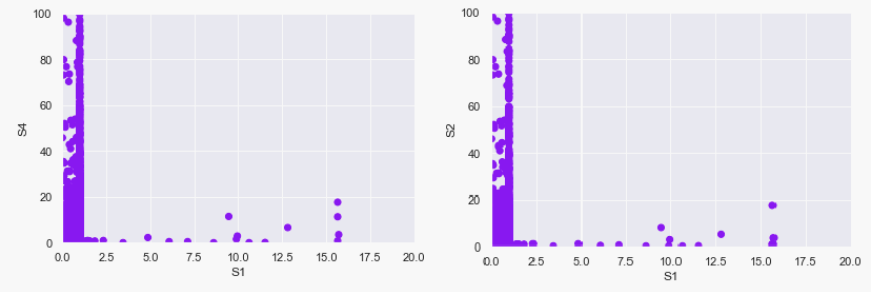


Figure 2.2.2.b.: Clustering of ratios S1 vs. S4 and S1 vs. S2.

As seen in the figure above, they form patterns, but actually only one cluster can be seen within this range, which is not a great support for further analysis.

The figures below present the result of the clustering of ratios S1 vs. S2. The figure is slightly zoomed in.

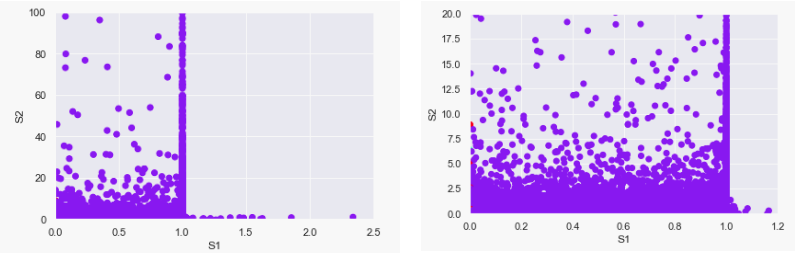


Figure 2.2.2.c.: Clustering of ratios S1 vs. S2.

As specified above, it is possible to find a certain pattern and draw conclusions based on it, but KMeans(12) only provides one cluster within this range (where the ratios should mostly be). Thus, this is of no help.

The testing conducted to remove extreme clusters and thereby being able to cluster the most interested regions from the perspective of ratios was conducted with the help of KMeans(200). The outcome is presented in the figure below, comparison with KMeans(12).

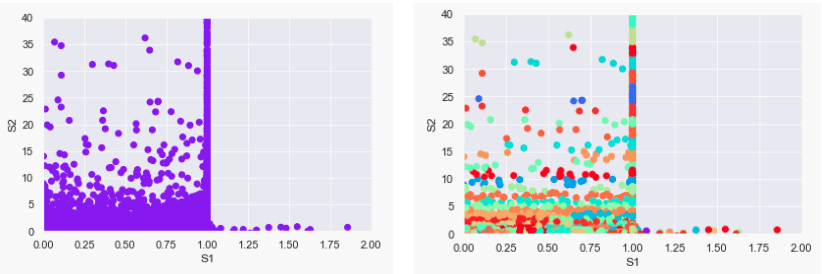


Figure 2.2.2.d.: Clustering of ratios S1 vs. S2, KMeans(12) vs. KMeans(200).

As the figure above shows (same scale, same ratios), increasing the number of clusters will help to create a situation in which clustering the ratios may actually be beneficial for the analysis.

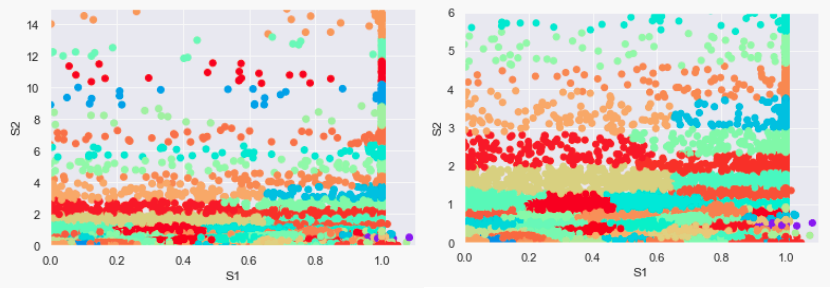


Figure 2.2.2.e.: Clustering of ratios S1 vs. S2, KMeans(200).

The figure above presents the clustering of S1 and S2 in comparison with Kmeans(200) in a somewhat more expanded manner internally. Clusters and patterns stand out clearly, but currently in a way which makes them difficult to interpret uniformly from the economic perspective.

A higher number of options and ratios were used to conduct the clustering of the ratios than described above, but the outcome is similar, in general, as illustrated above. Balance sheet ratios are very important indicators in economic analysis, but it is not easy to teach them separately to a machine. A machine cannot find clear patterns based on rations by unsupervised learning (at least by using the methods tested in this work); the patterns could be used to state directly whether or not the company may have payment difficulties.

Based on the same balance sheet indicator (described above), the machined managed to find patterns which are relevant from the economic perspective by unsupervised learning. Thus, the following step was taken in the course of the work: balance sheet data and the ratios based on the balance sheet were consolidated into one comprehensive dataset which was used as input in clustering (see below).

2.2.3. Joint clustering of balance sheet data and ratios

The outcomes of clustering balance sheet data with previously discussed ratios added to the balance sheet data provided above are presented below.

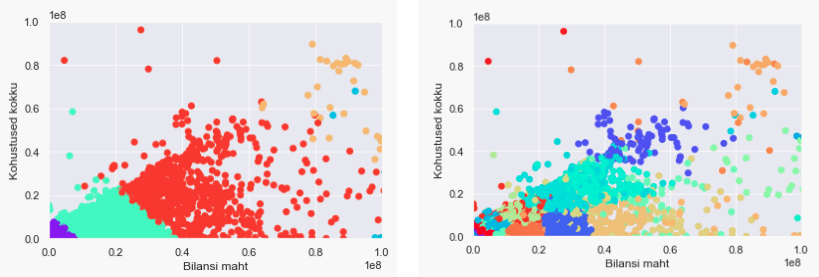


Figure 2.2.3.a.: Clustering of balance sheet data with ratios, KMeans(12).(12) and KMeans(50).

In the figure above, the figure on the left presents the joint clustering of balance sheet data and ratios in the context of twelve clusters; the figure on the right presents the same in the context of 50 clusters. The clustering conducted in the context of 50 clusters includes more clusters from the area (where the main analysis is conducted). The figure with 50 clusters also shows clearly that it is possible to differentiate more significant patters than in the case of 12 clusters. In the case of 12 clusters, it is mainly possible to differentiate the different clusters based on the sizes of the companies; in the case of 50 clusters, there are clearly distinguishable patterns (i.e. large balance sheet volume, low number of liabilities, liabilities proportional to the volume of the balance sheet, etc.).

In interpreting the clustering outcomes above (the figure on the left, KMeans(50)), it is also important to draw attention to the fact that some clusters are more compact and clearly defined, while others are diffuse. It is mostly reasonable to teach to a machine the clusters which are clearly define and have so-called compact economic indicators. There are two types of compact clusters: (a) some are defined by clearly manifesting economic indicators (a certain proportionality of the balance sheet volume and total liabilities, etc.); and (b) some are compact and relatively well-distinguished clusters from the perspective drawings, but it is difficult to give them a specific economic nature. Different clustering of economic data serves the purpose of finding the combinations and situations (patterns) which say something substantial from the economic perspective, but also enable the machine to seek them alone.

It is also important to stress from the perspective of the figures above that the schemes in the figures are two-dimensional (as the figures are two-dimensional), but several different balance sheet indicators have been clustered together and ratios have been calculated based on this. The figures can be highlighted by different indicators (taking two indicators from the entire selection and adding them to the figures). Drawing other indicators would provide different images of clusters.

In the figure below, the outcomes of the clustering of balance sheet data and ratios in the total liabilities vs. balance sheet volume context are zoomed. The figure shows that the clusters act proportionally with respect to the balance sheet and total liabilities if the volumes are small, but differently in the case of larger volumes (figure 2.2.3.a, upper right).

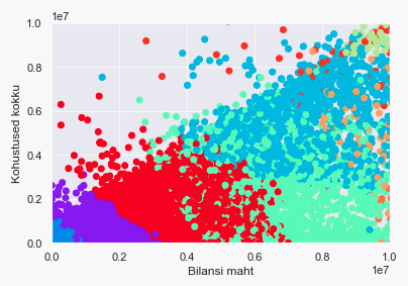
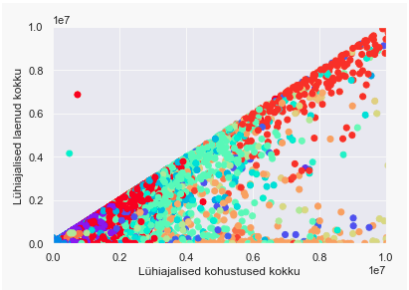


Figure 2.2.3.b.: Clustering of balance sheet data with ratios, KMeans(50).

In the figure below, the same clustering (balance sheet data and ratios together, KMeans(50)) of total short-term loans vs. current liabilities is presented in two different scales. Examining short-term loans and current liabilities together, it is possible to draw a pattern in which the so-called 'instant loan takers' stand out from among other companies (a situation in which the amount of short-term loans is almost equal to the amount of current liabilities).



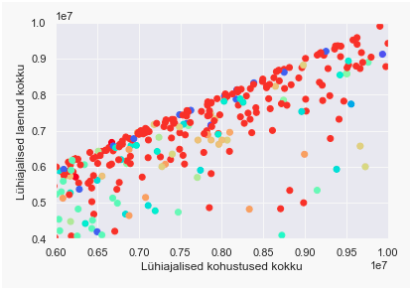


Figure 2.2.3.c.: Clustering of balance sheet data with ratios, short-term loans vs. current liabilities, KMeans(50).

The figure above shows that an axis is clearly forming with the companies whose volume of short-term loans is equal to the volume of short-term liabilities (in some cases, the amount of short-term loans exceeds the current liabilities, which indicates the likelihood of some data issues – short-term loans can be equal to or smaller than short-term obligations, which is also clearly seen in the figures).

For example, the cluster depicting the volume of current liabilities (red cluster, zoomed on the figure on the right) amounting to around six million euros clearly indicates that the companies are funding them with ‘instant loans’, which is a clear reference to potential insolvency.

On the other hand, the cluster with the volume of current liabilities of up to six million (the so-called green cluster) includes the ‘instant loan addicts’, as well as somewhat more conservative companies. Thus, the assessments cannot be uniform (at least in the context of a specific cluster).

In the figure below, the same clustering is presented for total long-term liabilities vs. total short-term liabilities (in different scales).

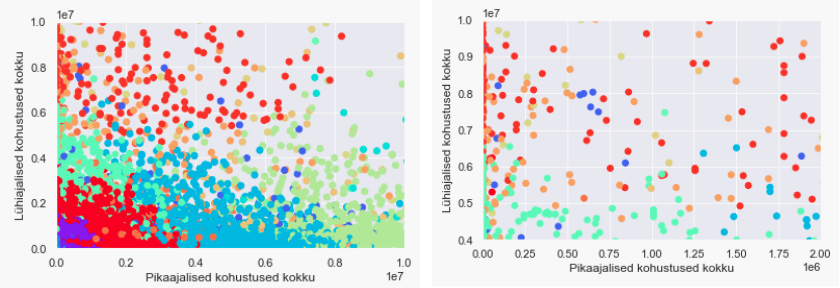


Figure 2.2.3.d.: Clustering of balance sheet data with ratios, long-term liabilities vs. current liabilities, KMeans(50).

In the figure above, we can find the red cluster of ‘instant loan takers’ discussed above, but it is clear in this respect that some of the companies there also have a high burden of non-current liabilities (which makes them riskier in this case, as there are also ‘home loans’ in addition to ‘instant loans’, which also require servicing).

The blue cluster is an interesting cluster (the volume of non-current liabilities between three to eight million euros). This cluster did not come up in the perspective of current liabilities and short-term loans, which means that the cluster is free of this risk. Instead, non-current liabilities (probably large bank loans) allow concluding that the companies are deemed reliable by banks and have positively passed the reliability test of the analysts there. The cluster was also clearly visible in the comparison of balance sheet volume and total liabilities (figure 2.2.3.b). This cluster has clear boundaries and can be interpreted from the economic perspective.

The figures and interpretations above provide an example of what is sought in developing this prototype and what grounds the work is based on. The numbers of patters involved in the work and the conclusions aimed for are actually higher. A summarising conclusion which can be highlighted in the analysis of the results of joint clustering of balance sheet data and balance sheet ratios is that as a result of this clustering, the system is capable, by using the unsupervised machine learning method, of finding patterns which can be interpreted from the economic perspective and which the machine can distinguish.

2.3. Analysis of transaction data

An additional approach introduced in the analysis for developing the prototype is clustering transaction data. Transaction data mean the amounts paid by one company to another which can be differentiated based on the amounts. The number of recipients is also known in the case of each specific payer (the specific companies which receive the amounts are known), which allows specifying the number of transaction partners. Thus, the content of transaction data consists of the amounts moving between transaction partners, as well as the number of transaction partners. The information originates from value added tax data.

The KMD_INF annex to the value added tax return specifies the details of the transactions of at least 1,000 euros. In this case, part A is used which includes sales invoices.
The information was obtained from here (https://www.emta.ee/ariklient/maksud-ja-tasumine/kaibemaks/kaibedeklaratsiooni-ja-aruannete-esitamine/kmd-inf-osa-muugiarved-taitmise-juhised):
Based on the general rule specified in subsection 11 (1) of the Value Added Tax Act, the time of supply or the time of receipt of services is deemed to be the date on which the goods are dispatched or made available to the purchaser, or the services are provided or the goods/service has been fully paid for, depending on which happens first.
Invoice data must be reported in the KMD and KMD INF forms in the month when turnover was partially or fully generated in connection with this invoice based on the time when the turnover was generated. The details of an invoice are reported in the KMD INF in the same taxation period when they are declared in the KMD.

On line 9 of the annex to the value added tax return, the taxable turnover specified in cells 1 + 2 of the KMD form in the taxation period must be specified. Thus, if the total amount of turnover is R1 + R2 + R3, the division of the turnover may be explained as follows:

- The operations and transactions taxable at the rate of 20% (R1) + the operations and transaction taxable at the rate of 9%(R2), which are reported in line 9 of the KMD_INF;
- The operations and transactions taxable at the rate of 20% (R1) + the operations and transaction taxable at the rate of 9%(R2), which are not reported in line 9 of the KMD_INF;
- The operations and transactions taxable at the rate of 0%.

This means that lines 1 and 2 of the value added tax return specify the value of the B2B transactions reported in KMD_INF, the value of the B2B transactions not reported in the KMD_INF, and sales to private individuals.
Previous analysis has shown that the value of the transactions reported in the KMD_INF has formed almost a half of the values highlighted on lines 1 and 2 of the KMD.

The main indicators were considered as the clustering input based on the KMD INF data:

- number of partners; number of companies to which a specific company has sold its services/production (from whom cash has been received);
- average payment; the average payment which the company has made within the period by the partners;
- median payment; the median payment which the company has made within the period by the partners;
- amount; the consolidated amount which the company has paid to the partners during the period, all partners in total;
- maximum; the maximum payment which the company has made to a partner during the period;
- minimum; the minimum payment which the company has made to a partner during the period.

The period of the first clustering is 30 days, i.e. one calendar month. A further 60-day or 90-day period may be chosen; they may all be examined separately or together (clustered). Separate clustering and then consolidating into one whole allows assessing the timeline, the dynamics in time (decreasing/increasing of solvency, moving from one warning category to another). One great advantage of transaction data is that they can be obtained and analysed on a monthly basis,

which means that opinions can be formed quickly on the changing of the financial situation of a company (not retrospectively: early warning vs. historical warning).

The figure below provides examples of the nature of transaction data.

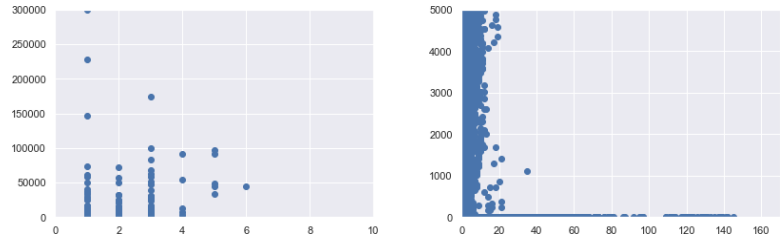


Figure 2.3.a.: Transaction data, balance sheet cluster 4, the entire real estate sector.

The left figure of the figure above presents the transaction data related to the cluster of the outcome of a specific balance sheet/ratio clustering (links are sought before the outcome clusters of different clusterings). The figure on the right shows an extract from a graph which includes all transaction data of the real estate sector. The vertical axis presents the total amount, the horizontal axis the number of partners.

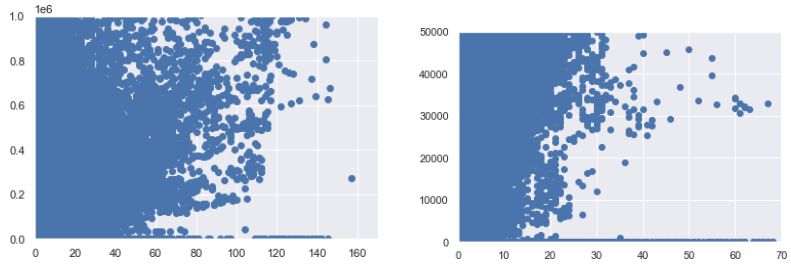


Figure 2.3.b.: Transaction data, amounts vs. number of partners, real estate.

The figures above also show transaction data in another scale to characterise the approximate nature of the data. The same is presented in the figure below.

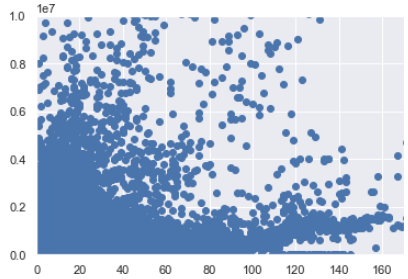


Figure 2.3.c.: Transaction data, amounts vs. number of partners, real estate.

In general, it may be claimed that transaction data differentiate from one another less than balance sheet data and the ratios thereof. On the one hand, this enables ‘easier clustering’ (i.e. fewer specific features and nuances); on the other hand, though, it is more difficult to substantiate the data from the economic perspective.

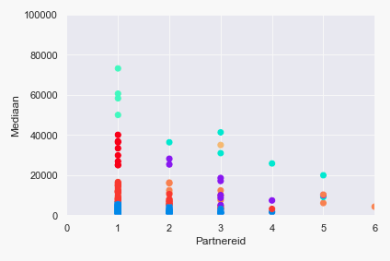


Figure 2.3.d.: Transaction data are clustered, linked with balance sheet cluster 4.

The figure above shows the outcome of the clustering of the data related to balance sheet cluster 4 by partners and median amounts. The figure below presents the same clustering by average amounts and median amounts.

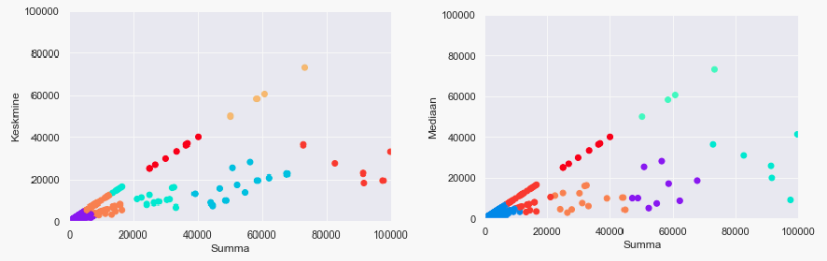


Figure 2.3.e.: Transaction data are clustered, linked with balance sheet cluster 4.

The figure above shows that different clusters can be differentiated clearly (as stated above, transaction data are easy to cluster).

The figure also shows that the clusters are small. The data consists of 270 entries and thus, the clusters are small. There are only 270 entries, as those were the entries which were compliant with the clustering outcome of the balance sheet data for the four clusters.

The above also shows an issue which arises in the case of linking different clustering outcomes. The sectors analysed (real estate, hotels, etc.) are of certain sizes (not very large). The clustering thereof creates clusters which are even smaller. A certain limited number of entries match those clusters in other datasets. If those are also clustered, the match is even more limited.

Construction of models is accompanied by very different datasets of limited sizes which describe accurately the economic conditions, but the amount of which is huge. The project aims to solve this situation by testing different methods for clustering and linking data to maintain the balance of the number of clusters and economic indicators. The substance of economic indicators may not be too low,

as the model would then fail to provide the required outcomes with sufficient accuracy for making predictions (warning someone).

The figure below presents clustered data of the real estate sector, amounts against the medians and average values.

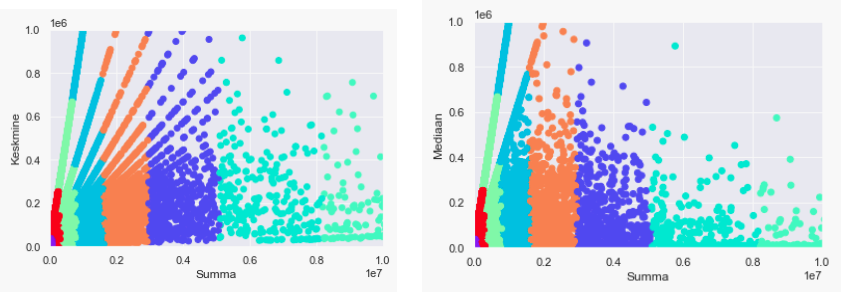


Figure 2.3.f.: Clustered transaction data, the real estate sector.

The figure above shows that it is possible to highlight different clusters when it comes to transaction data, which are clearly distinguishable, of comparable sizes, and compact. The clustering outcome is good, better than based on balance sheet data, but the issue of economic interpretation of the transaction data clusters is more complicated (as the figures show, the differences have mainly shifted based on the amounts, but this is not an indicator for assessing insolvency).

In the figure below, the same clustering of transaction data is presented in the partners vs. median scale in different scales.

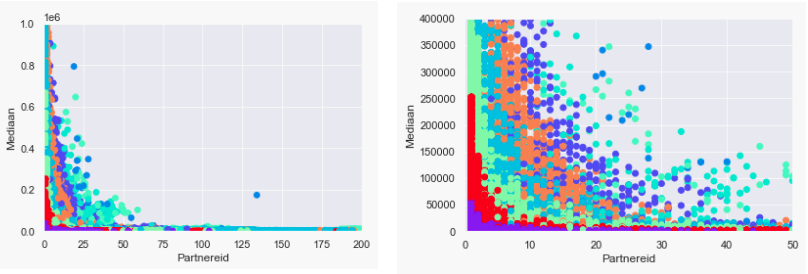


Figure 2.3.g.: Clustered transaction data, the real estate sector.

The figure above shows that transaction data clusters also draw clear patterns in the partners vs. median scales. However, those patterns are mainly divided based on size (i.e. more partners and a higher median compared to less partners and a lower median), which does not directly indicate the circumstances of potential insolvency.

The work on transaction data continues in the project, but two conclusions can be drawn in the current phase.

FIRST CONCLUSION. Transaction data form clearly distinguishable compact clusters of comparable sizes which are, however, more difficult from the financial perspective to interpret as those including the signs of insolvency. Important added value is hopefully gained from

examining the transaction clusters with respective links to balance sheet clusters (balance sheet clusters are linked with certain transactions in the transaction clusters).

SECOND CONCLUSION. Additional periods must be included for the indicators (30-day median complemented with 60-day median, 90-day median, the same when it comes to average, minimum, and maximum levels) and analyses must be conducted to determine whether the clusters provide more conclusions with signs of insolvency. The clusters of different times should also be compared, examining their change in time, i.e. the progress of the companies from one cluster to another and the economic meaning thereof.

The figure below illustrates the last conclusion. The figure presents one intermediate clustering of transaction data in the case of which the machine has included the period under the clustering input. Considering the period as an input is not wrong, but if the data are very 'brief', as transaction data are, the period may become excessively significant, which would not be correct. The figure below illustrates a case of incorrect clustering and also explains why repeat clustering is needed and how to find the best moment.

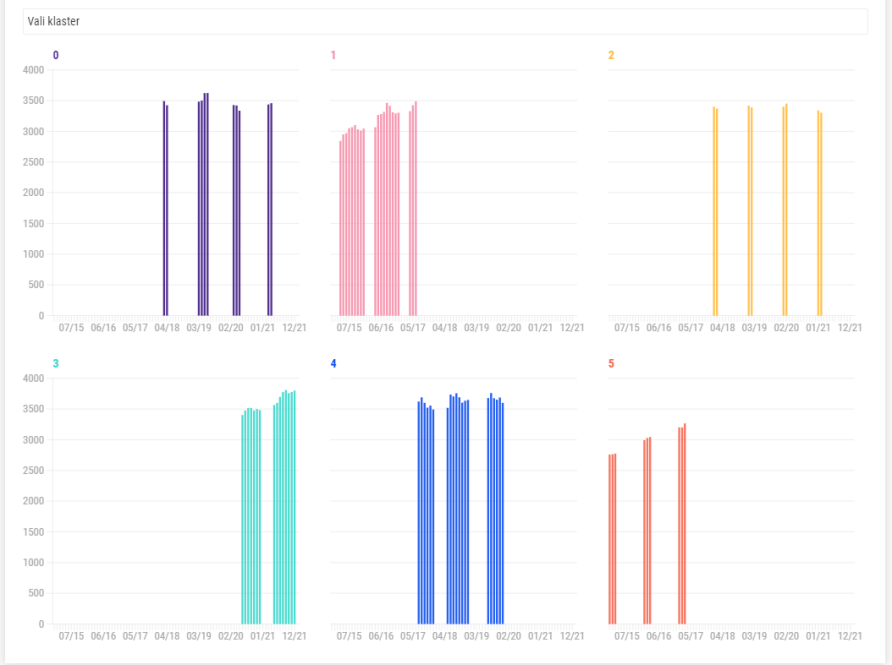


Figure 2.3.h.: Clustered transaction data, excessive importance of the period.

Annex 8 describes the division of the real estate sector into clusters by areas of

activity.

2.4. Creating the models and testing the data

2.4.1. General baseline prerequisites in creating the models

This work has involved developing models based on clustered data which are being taught to the system by using the machine learning method. Clustered data enable determining the companies which may become insolvent at a certain moment in future. We use the clustered data and the indicators of the data in training the classification model.

The work is based on the following logic:

Balance sheet data -> classification model -> prediction

Different models were tested in the course of the work: naïve bayes (gaussian, bernoulli, categorical, multinomial), svm, gradient boosting, etc. We decided in the favour of RandomForest, as even the out-of-box result is promising.

The figure below presents the output of the RandomForest base model.

RandomForest baasmudel

```
mudel = treeni(RandomForestClassifier())
```

```
The Training clf Accuracy is:  0.8687710962870535
classification_report =
              precision    recall  f1-score   support

           0       0.08      0.98      0.14       700
       0_A_1       1.00      0.84      0.91     58829
       0_A_2       0.64      0.87      0.73      1925
         0_B       0.83      0.82      0.83       827
           1       0.67      1.00      0.80         2
           2       1.00      1.00      1.00         2
           3       0.77      0.55      0.64        31
           4       0.83      0.92      0.87       195

    accuracy                           0.85     62511
   macro avg       0.73      0.87      0.74     62511
weighted avg       0.97      0.85      0.90     62511

Aega kulus: 1207.2708015441895 sekundit
```

Figure 2.4.1.a.: The outcome of the RandomForest model.

The figure above shows that a relatively high-level f1-score was occasionally achieved in teaching the model, as well as good precision, recall, and accuracy indicators.

In order to increase the efficiency of the RandomForest model used, the aspect of changing hyper parameters was added. The figure below shows the hyper parameters used in the model.

```
Parameters currently in use:

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

Figure 2.4.1.b.: The hyper parameters added to the RandomForest model.

All of the parameters will not be used in tuning the model; the parameters which will be changed to achieve a better outcome are highlighted below:

- n_estimators = number of trees in the forest;
- max_features = max number of features considered for splitting a node;
- max_depth = max number of levels in each decision tree;
- min_samples_split = min number of data points placed in a node before the node is split;
- min_samples_leaf = min number of data points allowed in a leaf node;
- bootstrap = method for sampling data points (with or without replacement).

In order to understand the potential scale of those parameters, we use RandomSearchCV.

A wide range of parameters is provided for the model and a slightly narrower selection of parameters is obtained in return.

The initial wide selection of parameters scanned (k-fold CV – 3 foldi) is specified in the figure below.

```
from sklearn.model_selection import RandomizedSearchCV
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 7, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 3, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
```

Figure 2.4.1.c.: The selection of parameters scanned.

The figure below shows the area of parameters received in which the model functions/predicts best.

```
rf_random.best_params_
```

```
{'n_estimators': 200,
 'min_samples_split': 7,
 'min_samples_leaf': 3,
 'max_features': 'auto',
 'max_depth': 100,
 'bootstrap': False}
```

Figure 2.4.1.d.: The area of parameters with the best prediction capability.

The result of RandomSearchCV is referred to GridSearchCV where the parameters are 'fine-tuned'. A scan is conducted around the results of the previous search (k-fold CV – 3 foldi).

```
from sklearn.model_selection import GridSearchCV
# Create the parameter grid based on the results of random search
param_grid = {
    'bootstrap': [False],
    'max_depth': [80, 90, 100, 110],
    'max_features': ['auto'],
    'min_samples_leaf': [2, 3, 4],
    'min_samples_split': [5, 7, 9],
    'n_estimators': [100, 200, 300, 400]
}
```

Figure 2.4.1.e.: Using GridSearchCV. The figure below

shows the result of GridSearchCV.

```
grid_search.best_params_
```

```
{'bootstrap': False,
 'max_depth': 110,
 'max_features': 'auto',
 'min_samples_leaf': 3,
 'min_samples_split': 7,
 'n_estimators': 300}
```

Figure 2.4.1.f.: The result of GridSearchCV.

The results of GridSearchCV are used in the further training of the model.

2.4.2. Using prediction models

The material below describes the prediction of data in the course of this work, plus illustrative examples to provide an overview; in practice, a higher number of different methods and several different solutions are used.

The figure below presents reading the data.

```
def data_preparation(x):
    '''
    Algandmete töötlus, et viia andmed mudelile sobivale kujule
    '''
    df_x = pd.read_csv(x)
    df_x.replace([np.inf, -np.inf], np.nan, inplace=True)
    df_x.fillna(0, inplace=True)

    return df_x
```

```
# Loeme sisse testandmed
to_predict = data_preparation('../../data/raw/forecasting.csv')

to_predict
```

Figure 2.4.2.a.: Reading the data.

The figure below presents a sample extract from the data read.

PERIOD_NM	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
2019	18201.0	2012.0	2012.0	20213.0	0.0	1158.0	1158.0	0.0	1158.0	2556.0	8577.0	8327.0
2017	787467.0	331317.0	342454.0	1129921.0	7121.0	556432.0	563553.0	0.0	563553.0	0.0	93436.0	10271.0
2012	1316520.0	186972.0	186972.0	1503492.0	2206.0	675259.0	692465.0	9368.0	701833.0	51200.0	783465.0	720148.0
2018	1080009.0	2600.0	2600.0	1082609.0	7500.0	160584.0	168084.0	541719.0	709803.0	23966.0	10904.0	27491.0
2016	84700.0	98060.0	98060.0	182760.0	56699.0	0.0	56699.0	20000.0	150049.0	2550.0	38474.0	29164.0
2019	20000.0	56000.0	120000.0	1200000.0	23000.0	3000.0	4000.0	40000.0	44000.0	3000.0	12000.0	11000.0
2019	10000.0	30000.0	40000.0	500000.0	120000.0	2000.0	120000.0	240000.0	360000.0	5000.0	3000.0	3500.0
2018	7000.0	20000.0	20000.0	100000.0	25000.0	4000.0	30000.0	40000.0	70000.0	10000.0	5000.0	4000.0

Figure 2.4.2.b.: A data sample.

The figure below presents the reading of models which have been read before.

```
# Loeme varasemalt treenitud RF mudelid
rf_base = joblib.load('../../models/rf_base.sav')
rf_tuned = joblib.load('../../models/rf_tuned.sav')
```

Figure 2.4.2.c.: The reading of models which have been read

before. The figure below presents predicting.

```
def predict_data(model, model2, data):
    '''
    Mudelite testimine
    '''
    X_predict = data.drop(columns=['kood'])
    predictions = model.predict(X_predict)
    predictions2 = model2.predict(X_predict)

    data['pred'] = predictions
    data['pred_2'] = predictions2

    return data
```

```
# Prognoosimine
predictions = predict_data(rf_base, rf_tuned, to_predict)
predictions
```

Figure 2.4.2.d.: Using the models for predicting.

The figure below presents a result of predicting with columns pred and pred 2 specifying the clusters to which the machine believed certain data rows tested to belong.

PERIOD_NM	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2	pred	pred_2
2019	18201.0	2012.0	2012.0	20213.0	0.0	1158.0	1158.0	0.0	1158.0	2556.0	8577.0	8327.0	0_A_1	0
2017	787467.0	331317.0	342454.0	1129921.0	7121.0	556432.0	563553.0	0.0	563553.0	0.0	93436.0	10271.0	0_A_1	0_A_1
2012	1316520.0	186972.0	186972.0	1503492.0	2206.0	675259.0	692465.0	9368.0	701833.0	51200.0	783465.0	720148.0	0_A_2	0_A_2
2018	1080009.0	2600.0	2600.0	1082609.0	7500.0	160584.0	168084.0	541719.0	709803.0	23966.0	10904.0	27491.0	0_A_1	0_A_1
2016	84700.0	98060.0	98060.0	182760.0	56699.0	0.0	56699.0	20000.0	150049.0	2550.0	38474.0	29164.0	0_A_1	0_A_1
2019	20000.0	56000.0	120000.0	1200000.0	23000.0	3000.0	4000.0	40000.0	44000.0	3000.0	12000.0	11000.0	0_A_1	0_A_1
2019	10000.0	30000.0	40000.0	500000.0	120000.0	2000.0	120000.0	240000.0	360000.0	5000.0	3000.0	3500.0	0_A_1	0_A_1
2018	7000.0	20000.0	20000.0	100000.0	25000.0	4000.0	30000.0	40000.0	70000.0	10000.0	5000.0	4000.0	0_A_1	0_A_1
2020	10.0	6000.0	12000.0	250000.0	150000.0	10.0	180000.0	30000.0	210000.0	2000.0	10.0	300.0	0_A_1	0_A_1
2021	200.0	2000.0	5000.0	120000.0	90000.0	200.0	100000.0	5000.0	105000.0	10.0	200.0	400.0	0_A_1	0_A_1
2020	193097.0	43095.0	43095.0	236192.0	6133.0	150722.0	156855.0	6245.0	163100.0	2556.0	5946.0	37448.0	0_A_1	0_A_1
2021	3341507.0	1552252.0	62212413.0	65553920.0	696896.0	767586.0	1464482.0	28243839.0	29708321.0	10000.0	3262134.0	731815.0	4	4
2021	43694.0	9115.0	9893.0	53587.0	4213.0	531.0	4744.0	860.0	5604.0	2500.0	24050.0	36550.0	0_A_1	0_A_1

Figure 2.4.2.e.: A result of predicting.

There are two predictions: pred is the base model and bred_2 included tuned hyper parameters. They provide different results to a certain extent.

2.4.3. Real-life verifying of the prediction model

The prediction models undergo real-life verifying within the framework of this work. A company is picked which the model has placed in a certain cluster; the economic behaviour of the company is analysed based on information from third sources; and the conclusion is drawn on whether or not the machine has drawn correct conclusions.

For example, the system deemed risky the behaviour of a company with the share capital of 10,000 euros which increased its investments in buying real estate by 60 percent to 12.3 million euros in 2019 (i.e. this amount was invested within the last year), the volume of the real estate portfolio reached 35 million euros, the loan burden increased from 9 million euros to 15 million euros within the space of the year (with 6 million euros borrowed within the last year).

The system found a balance sheet with a strong loan leverage which grew at a very different (higher) pace compared to the normal and concluded that this loan-based rapid growth could be risky. The conclusion is correct, as the risk of payment difficulties may be a likely scenario if real estate prices drop (further bank guarantees and requirements for instalments) and the real estate may also not be capable of covering the loan expenses.

2.4.4. Euclidean distance analysis of data

The Euclidean distance is used for determining the relative distance between the data (tested data vs. model data; teaching data vs. tested data):

$$d = \sqrt{[\,(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

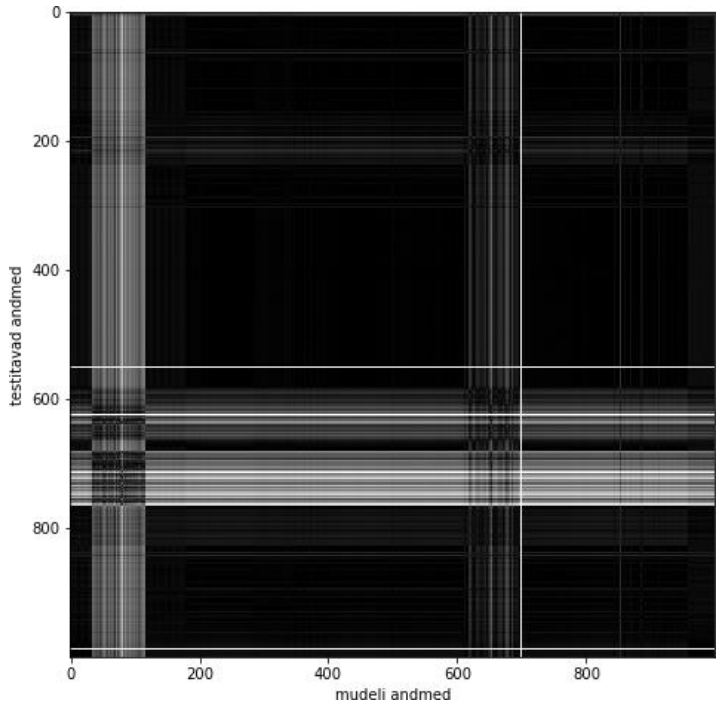


Figure 2.4.4.a.: The outcome distance matrix.

The figure above presents the distance matrix which shows the difference between the model data and tested data (i.e. the input data used for the prediction). This is a tool which is used widely in this work. It can be used to examine the similarity of training data and testing data, the differences between model data and prediction data, to check the prediction data against training data, etc.

2.4.5. Using the loss function in optimising the model

The loss function is a useful tool in optimising a model. The purpose of specifying (optimising) the model is minimising the cost function. It is very important to optimise the model (models) which assess businesses in this work.

The figure below presents the 'cost history' and also the functions of improving the accuracy of the model, in parallel (Classification accuracy).

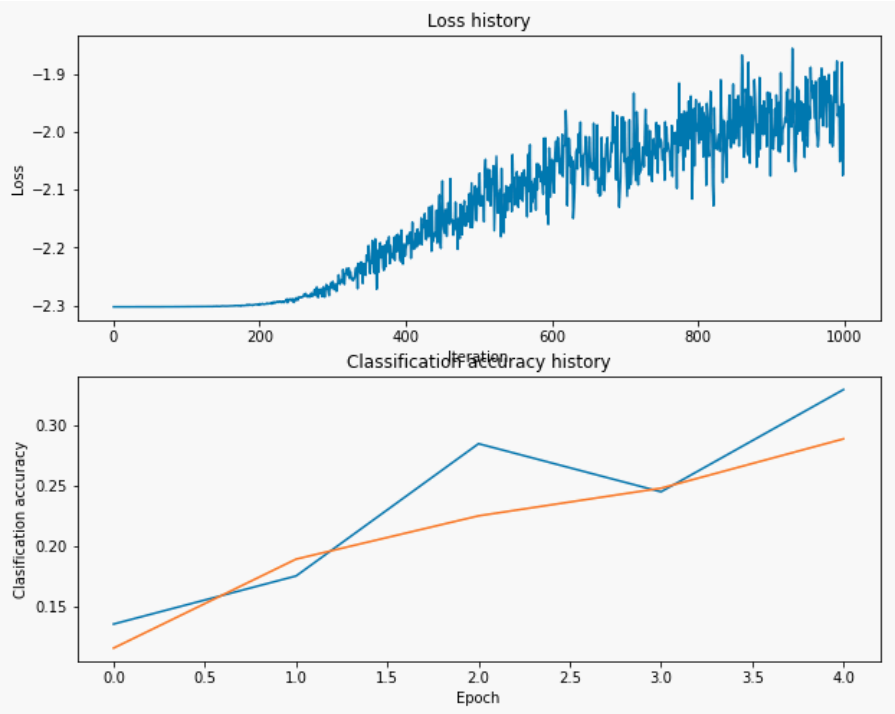


Figure 2.4.5.a.: Loss history and Classification accuracy.

As is apparent from the figure above, the expenses (loss) decrease and the accuracy of the model improves. The indicators provided are not the final indicators, as better parameters are expected from the model. They are, however, an illustration of how models are worked with and better solutions are sought within the framework of this work.

The figure below presents the training loss history in the second training case.

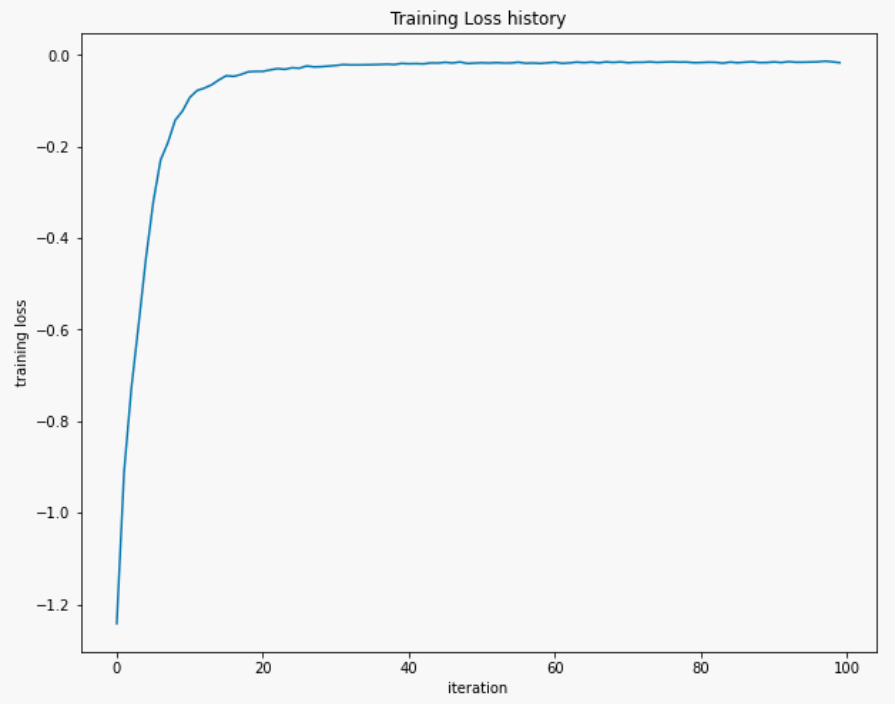


Figure 2.4.5.b.: Loss history.

The loss function in the case of a higher number of iterations is provided below. The figure shows that the more iterations, the lower the loss function.

Figure 2.4.5.c.: Loss history.

The figure below shows how accuracy has been taken to a considerably better level (compared to figure 2.4.5.a., for example – even though Classification accuracy is highlighted there). Both training and validation histories have been generated.

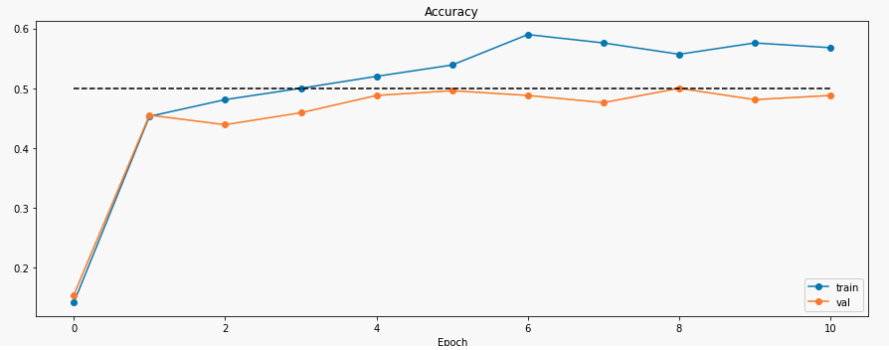


Figure 2.4.5.d.: Training and validation history. An example of a code for plotting training loss history is provided below.

```
plt.subplot(2, 1, 1)
plt.title('Training loss')
plt.plot(solver.loss_history, 'o')
plt.xlabel('Iteration')

plt.subplot(2, 1, 2)
plt.title('Accuracy')
plt.plot(solver.train_acc_history, '-o', label='train')
plt.plot(solver.val_acc_history, '-o', label='val')
plt.plot([0.5] * len(solver.val_acc_history), 'k--')
plt.xlabel('Epoch')
plt.legend(loc='lower right')
plt.gcf().set_size_inches(15, 12)
plt.show()
```

Figure 2.4.5.e.: Training loss plotting.

Below, a code example is provided for calculating and plotting the final training loss.

```
net = init_toy_model()
stats = net.train(X, y, X, y, learning_rate=1e-1, reg=5e-6, num_iters=100, verbose=False)

print('Final training loss: ', stats['loss_history'][-1])

# plot the loss history
plt.plot(stats['loss_history'])
plt.xlabel('iteration')
plt.ylabel('training loss')
plt.title('Training Loss history')
plt.show()
```

```
Final training loss:  -0.01713275892330469
```

Figure 2.4.5.f.: Final training loss assessment and plot.

The example below is concerned with minimising training loss and improving accuracy.

(Iteration 1 / 4900) loss: 2.300403
(Epoch 0 / 10) train acc: 0.141000; val_acc: 0.154000
(Epoch 1 / 10) train acc: 0.453000; val_acc: 0.455000
(Iteration 501 / 4900) loss: 1.442970
(Epoch 2 / 10) train acc: 0.481000; val_acc: 0.439000
(Iteration 1001 / 4900) loss: 1.455718
(Epoch 3 / 10) train acc: 0.500000; val_acc: 0.459000
(Iteration 1501 / 4900) loss: 1.386509
(Epoch 4 / 10) train acc: 0.520000; val_acc: 0.488000
(Iteration 2001 / 4900) loss: 1.374266
(Epoch 5 / 10) train acc: 0.539000; val_acc: 0.496000
(Iteration 2501 / 4900) loss: 1.179609
(Epoch 6 / 10) train acc: 0.590000; val_acc: 0.488000
(Iteration 3001 / 4900) loss: 1.315973
(Epoch 7 / 10) train acc: 0.576000; val_acc: 0.476000
(Iteration 3501 / 4900) loss: 1.294193
(Epoch 8 / 10) train acc: 0.557000; val_acc: 0.500000
(Iteration 4001 / 4900) loss: 1.371872
(Epoch 9 / 10) train acc: 0.576000; val_acc: 0.481000
(Iteration 4501 / 4900) loss: 1.143015
(Epoch 10 / 10) train acc: 0.568000; val_acc: 0.488000

An example of seeking the best accuracy.

Training hidden_size: 400
Training learning_rate: 0.003
Training reg: 0.025
Training batch_size: 500
iteration 0 / 1200: loss -2.302402
iteration 100 / 1200: loss -1.649690
iteration 200 / 1200: loss -1.509460
iteration 300 / 1200: loss -1.383044
iteration 400 / 1200: loss -1.513232
iteration 500 / 1200: loss -1.415503
iteration 600 / 1200: loss -1.431067
iteration 700 / 1200: loss -1.428305
iteration 800 / 1200: loss -1.259119
iteration 900 / 1200: loss -1.285956
iteration 1000 / 1200: loss -1.279291
iteration 1100 / 1200: loss -1.136774
Current val_acc: 0.537
Best Accuracy: 0.537
Best Hidden Size: 400
Best Learning Rate:
0.003 Best reg: 0.025
Best batch_size: 500
Training hidden_size: 400
Training learning_rate: 0.003
Training reg: 0.03
Training batch_size: 500
iteration 0 / 1200: loss -2.302366
iteration 100 / 1200: loss -1.741899
iteration 200 / 1200: loss -1.589219
iteration 300 / 1200: loss -1.432846
iteration 400 / 1200: loss -1.482398
iteration 500 / 1200: loss -1.392184
iteration 600 / 1200: loss -1.401652

iteration 700 / 1200: loss -1.345791
iteration 800 / 1200: loss -1.334304
iteration 900 / 1200: loss -1.374942
iteration 1000 / 1200: loss -1.112893
iteration 1100 / 1200: loss -1.142935
Current val_acc: 0.508
Training hidden_size: 400
Training learning_rate: 0.003
Training reg: 0.035
Training batch_size: 500
iteration 0 / 1200: loss -2.302311
iteration 100 / 1200: loss -1.713333
iteration 200 / 1200: loss -1.517223
iteration 300 / 1200: loss -1.536061
iteration 400 / 1200: loss -1.389710
iteration 500 / 1200: loss -1.371919
iteration 600 / 1200: loss -1.359302
iteration 700 / 1200: loss -1.442901
iteration 800 / 1200: loss -1.327040
iteration 900 / 1200: loss -1.130709
iteration 1000 / 1200: loss -1.229156
iteration 1100 / 1200: loss -1.193381
Current val_acc: 0.508
Training hidden_size: 400
Training learning_rate: 0.003
Training reg: 0.05
Training batch_size: 500
iteration 0 / 1200: loss -2.302266
iteration 100 / 1200: loss -1.594231
iteration 200 / 1200: loss -1.592442
iteration 300 / 1200: loss -1.582597
iteration 400 / 1200: loss -1.403937
iteration 500 / 1200: loss -1.298283
iteration 600 / 1200: loss -1.398085
iteration 700 / 1200: loss -1.219892
iteration 800 / 1200: loss -1.228358
iteration 900 / 1200: loss -1.197194
iteration 1000 / 1200: loss -1.280821
iteration 1100 / 1200: loss -1.017234
Current val_acc: 0.486
Final: Best Accuracy:
0.537
Final: Best Hidden Size: 400
Final: Best Learning Rate:
0.003 Final: Best reg: 0.025
Final: Best batch_size: 500
And finally: BE CAREFUL WITH THOSE NEURAL NETWORKS!!!

2.4.6. Experimental statistical perspective of artificial neural networks

This work also uses the experimental statistical neural network-based perspective. In this phase of the work, it is difficult to tell to what extent the use of artificial neural networks will pay off in this work, but examples are provided below on what is used within the framework of the work and how.

The figure below demonstrates an example of the Affine Forward code.

```
# Test the affine_forward function

num_inputs = 2
input_shape = (4, 5, 6)
output_dim = 3

input_size = num_inputs * np.prod(input_shape)
weight_size = output_dim * np.prod(input_shape)

x = np.linspace(-0.1, 0.5, num=input_size).reshape(num_inputs, *input_shape)
w = np.linspace(-0.2, 0.3, num=weight_size).reshape(np.prod(input_shape), output_dim)
b = np.linspace(-0.3, 0.1, num=output_dim)

out, _ = affine_forward(x, w, b)
correct_out = np.array([[ 1.49834967,  1.70660132,  1.91485297],
                        [ 3.25553199,  3.5141327,   3.77273342]])

# Compare your output with ours. The error should be around 1e-9.
print('Testing affine_forward function:')
print('difference: ', rel_error(out, correct_out))
```

```
Testing affine_forward function:
difference:  9.7698500479884e-10
```

Figure 2.4.6.a.: Affine Forward.

The figure below demonstrates an example of the Affine Backward code.

```
# Test the affine_backward function
np.random.seed(231)
x = np.random.randn(10, 2, 3)
w = np.random.randn(6, 5)
b = np.random.randn(5)
dout = np.random.randn(10, 5)

dx_num = eval_numerical_gradient_array(lambda x: affine_forward(x, w, b)[0], x, dout)
dw_num = eval_numerical_gradient_array(lambda w: affine_forward(x, w, b)[0], w, dout)
db_num = eval_numerical_gradient_array(lambda b: affine_forward(x, w, b)[0], b, dout)

_, cache = affine_forward(x, w, b)
dx, dw, db = affine_backward(dout, cache)

# The error should be around 1e-10
print('Testing affine_backward function:')
print('dx error: ', rel_error(dx_num, dx))
print('dw error: ', rel_error(dw_num, dw))
print('db error: ', rel_error(db_num, db))
```

```
Testing affine_backward function:
dx error:  1.0908210113205496e-10
dw error:  2.273805557790167e-10
db error:  7.736978834487815e-12
```

Figure 2.4.6.b.: Affine Backward.

The figure below demonstrates an example of the Relu Forward Function code.

```
# Test the relu_forward function

x = np.linspace(-0.5, 0.5, num=12).reshape(3, 4)

out, _ = relu_forward(x)
correct_out = np.array([[ 0.,          0.,          0.,          0.,        ],
                        [ 0.,          0.,          0.04545455,  0.13636364,],
                        [ 0.22727273,  0.31818182,  0.40909091,  0.5,       ]])

# Compare your output with ours. The error should be around 5e-8
print('Testing relu_forward function:')
print('difference: ', rel_error(out, correct_out))
```

```
Testing relu_forward function:
difference:  4.999999798022158e-08
```

Figure 2.4.6.c.: Relu Forward Function.

The figure below provides an example of the Relu Backward Function code.

```
np.random.seed(231)
x = np.random.randn(10, 10)
dout = np.random.randn(*x.shape)

dx_num = eval_numerical_gradient_array(lambda x: relu_forward(x)[0], x, dout)

_, cache = relu_forward(x)
dx = relu_backward(dout, cache)

# The error should be around 3e-12
print('Testing relu_backward function:')
print('dx error: ', rel_error(dx_num, dx))
```

```
Testing relu_backward function:
dx error:  3.2756349136310288e-12
```

Figure 2.4.6.d.: Relu Backward Function.

```
from layer_utils import affine_relu_forward, affine_relu_backward
np.random.seed(231)
x = np.random.randn(2, 3, 4)
w = np.random.randn(12, 10)
b = np.random.randn(10)
dout = np.random.randn(2, 10)

out, cache = affine_relu_forward(x, w, b)
dx, dw, db = affine_relu_backward(dout, cache)

dx_num = eval_numerical_gradient_array(lambda x: affine_relu_forward(x, w, b)[0], x, dout)
dw_num = eval_numerical_gradient_array(lambda w: affine_relu_forward(x, w, b)[0], w, dout)
db_num = eval_numerical_gradient_array(lambda b: affine_relu_forward(x, w, b)[0], b, dout)

print('Testing affine_relu_forward:')
print('dx error: ', rel_error(dx_num, dx))
print('dw error: ', rel_error(dw_num, dw))
print('db error: ', rel_error(db_num, db))
```

```
Testing affine_relu_forward:
dx error:  6.750573928879482e-11
dw error:  8.162015570444288e-11
db error:  7.826724021458994e-12
```

Figure 2.4.6.e.: Affine Relu Forward Function.

```
np.random.seed(231)
num_classes, num_inputs = 10, 50
x = 0.001 * np.random.randn(num_inputs, num_classes)
y = np.random.randint(num_classes, size=num_inputs)

dx_num = eval_numerical_gradient(lambda x: softmax_loss(x, y)[0], x, verbose=False)
loss, dx = softmax_loss(x, y)

# Test softmax_loss function. Loss should be 2.3 and dx error should be 1e-8
print('Testing softmax_loss:')
print('loss: ', loss)
print('dx error: ', rel_error(dx_num, dx))
```

```
Testing softmax_loss:
loss:  2.302545844500738
dx error:  9.384673161989355e-09
```

Figure 2.4.6.f.: Softmax Loss Function testing.

The figures below demonstrate an example of the TwoLayerNet testing code.

```
np.random.seed(231)
N, D, H, C = 3, 5, 50, 7
X = np.random.randn(N, D)
y = np.random.randint(C, size=N)

std = 1e-3
model = TwoLayerNet(input_dim=D, hidden_dim=H, num_classes=C, weight_scale=std)

print('Testing initialization ... ')
W1_std = abs(model.params['W1'].std() - std)
b1 = model.params['b1']
W2_std = abs(model.params['W2'].std() - std)
b2 = model.params['b2']
assert W1_std < std / 10, 'First layer weights do not seem right'
assert np.all(b1 == 0), 'First layer biases do not seem right'
assert W2_std < std / 10, 'Second layer weights do not seem right'
assert np.all(b2 == 0), 'Second layer biases do not seem right'

print('Testing test-time forward pass ... ')
model.params['W1'] = np.linspace(-0.7, 0.3, num=D*H).reshape(D, H)
model.params['b1'] = np.linspace(-0.1, 0.9, num=H)
model.params['W2'] = np.linspace(-0.3, 0.4, num=H*C).reshape(H, C)
model.params['b2'] = np.linspace(-0.9, 0.1, num=C)
X = np.linspace(-5.5, 4.5, num=N*D).reshape(D, N).T
scores = model.loss(X)
correct_scores = np.asarray(
  [[11.53165108,  12.2917344,   13.05181771,  13.81190102,  14.57198434, 15.33206765,  16.09215096],
   [12.05769098,  12.74614105,  13.43459113,  14.1230412,   14.81149128, 15.49994135,  16.18839143],
   [12.58373087,  13.20054771,  13.81736455,  14.43418138,  15.05099822, 15.66781506,  16.2846319 ]])
scores_diff = np.abs(scores - correct_scores).sum()
assert scores_diff < 1e-6, 'Problem with test-time forward pass'

print('Testing training loss (no regularization)')
y = np.asarray([0, 5, 1])
loss, grads = model.loss(X, y)
correct_loss = 3.4702243556
assert abs(loss - correct_loss) < 1e-10, 'Problem with training-time loss'
```

Figure 2.4.6.g.: TwoLayerNet testing 1.

```
model.reg = 1.0
loss, grads = model.loss(X, y)
correct_loss = 26.5948426952
assert abs(loss - correct_loss) < 1e-10, 'Problem with regularization loss'

for reg in [0.0, 0.7]:
    print('Running numeric gradient check with reg = ', reg)
    model.reg = reg
    loss, grads = model.loss(X, y)
    for name in sorted(grads):
        f = lambda _: model.loss(X, y)[0]
        grad_num = eval_numerical_gradient(f, model.params[name], verbose=False)
        print('%s relative error: %.2e' % (name, rel_error(grad_num, grads[name])))
        assert rel_error(grad_num, grads[name]) < 0.6, "Error with gradient for " + name
```

```
Testing initialization ...
Testing test-time forward pass ...
Testing training loss (no regularization)
Running numeric gradient check with reg =  0.0
W1 relative error: 1.52e-08
W2 relative error: 3.30e-10
b1 relative error: 8.37e-09
b2 relative error: 1.34e-10
Running numeric gradient check with reg =  0.7
W1 relative error: 2.53e-07
W2 relative error: 7.98e-08
b1 relative error: 1.56e-08
b2 relative error: 7.76e-10
```

Figure 2.4.6.h.: TwoLayerNet testing 2.

A further code is provided in Annexes

9-11.

Chapter III – the activities performed in the course of the work

The first version of the final report of developing the business analysis and prototype of the early warning service was submitted on 31 March 2022. A respective roundtable was organised on 12 April 2022 whether the situation of the prototype and potential further developments were discussed. 22 April was set as the deadline for providing feedback.

Chapter III of this material describes:

- what has been done within the framework of the work after submitting the report on 31 March 2022, i.e. how the prototype has been supplemented, taking into consideration what was known by the end of March 2022, which bottlenecks had been detected, and in which directions further work was going to be done;
- how have the opinions discussed on 12 April been taken into consideration in the work and to which extent are they going to be taken into consideration;
- how has the feedback received by 22 April been taken into consideration in the work and to which extent is it going to be taken into consideration;
- other important activities and opinions carried out which were taken into consideration in April 2022.

This material also includes a summary chapter. Some of opinions and conclusions drawn as a result of the feedback received in April 2022 are not discussed in this Chapter III, but in the summary chapter. The material is distributed between this Chapter III and the summary chapter based on primarily including in the summary the ideas and forward-looking perspectives which provide a compact overview. From the perspective of providing comprehensive information, Chapter III and the summary chapter complement one another.

3.1. Drawing up additional models for sectors

Annex 1 to this material specifies the four areas of activity defined: (a) the medical sector, (b) the sector of financial services, (c) the real estate sector, and (d) the education sector. The work done so far has mainly been concerned with analysing the field of real estate. This selection was made somewhat randomly based on the principle of 'having to start somewhere'.
On the other hand, the real estate sector was selected first, as it is inevitably (depending on how widely to define it, see Annex 1) a sector which has a significant impact on the Estonian economy. For the proper functioning of the early warning system, the prototype must be tested in different sectors.

The prototype may function well in some sectors, but not so well in others. It is as important to obtain proper answers in small niche sectors as it is in the sectors of a wider scope, because the warnings issued are useless otherwise and fail to serve their purpose. The development of the system began from large sectors, but that does not mean that smaller sectors will be ignored or considered less important. The sectors are currently divided as indicated below, but it may become necessary to define the sectors differently in the course of testing.

In April 2022, the models (i.e. different algorithms) were tested with the data of the sectors listed in Annex 1:

- the medical sector;
- the sector of financial services;
- the education sector.

Furthermore, the following were defined in April and attempts were made to use the model for testing the data from the following sectors:

- accommodation services;
- catering;
- entertainment;
- agriculture;
- information technology;
- electricity generation;
- wholesale;
- retail;
- logistics.

Defining some of the aforementioned areas of activity and the data of those areas provided clearer results, others less clear results. The sectors differ greatly and have their own different issues. For example, the medical sector is a clear public sector area of activity. In the case of financial services, the central bank, commercial banks, insurance undertakings, financial advisers, and currency exchangers (which are all very different companies by nature and the central bank is not a company at all) are involved in the field. Eesti Energia is involved in electricity generation in Estonia, as well as many small businesses (again very different companies). Agricultural undertakings area also different due to their areas of activity (one set of rules applies to dairy producers, another to producers of cereals).

The examples above are just a few examples of the technical circumstances which have been identified (which is inevitable in this work) as a result of defining different areas of activity. It is important to pay attention to these issues in the course of test and developing the prototype and to solve them gradually (by specifying the sectors in further detail or adjusting models appropriately or both).

Conducting real-life tests can be of a significant assistance in making the model more specific (see the Chapter 'Preparation of real-life tests').

3.2. Preparation of real-life tests

In April 2022, the work involved active preparations for real-life tests. Conducting real-life tests means that actual data of businesses is taken and sent to the prototype (the data are entered into the model). The prototype draws a conclusion about the business and issues it. The prototype says whether the company should be issued a certain warning (the warnings have three different levels) and the prototype also issues further information on why the respective warning was issued to the company.

The discussions held on organising real-life tests have led to a conclusion that it would be reasonable and optimal to conduct the tests with the Harju Economic Development Center. The Harju Economic Development Center has been very helpful and thought along efficiently on how to make the prototype viable and how it could be most beneficial for actual companies. The centre can use its practical competence and network of connections to help put together the right test group on the one hand and to obtain full feedback on the opinions of the companies tested and the ideas of those companies on how this warning could serve them in the best way possible on the other hand.

In this material, it is concluded that the Harju Economic Development Center is a good, appropriate, and optimal (the best) partner in every way for testing the prototype on actual economic operators (companies) with the required feedback from those economic operators.

3.3. Gradual improvement of the prototype

At the roundtable on 12 April 2022, one of the issues discussed, among others, was that an early warning system which is based on machine learning models will be learning and becoming smarter gradually depending on how much it is used (how much input is entered into the model and how much of it is saved by the model or taken into consideration in future decision-making).

This fact was primarily kept in mind in the course of complementing and improving the model in April 2022. The model has been trained with increasingly more relevant data, with different sectors also taught to the model. Solutions have been implemented which attempt to exclude so-called false positive outcomes (i.e. such outcomes in which the machine issues a warning, but there is actually no reason for the warning).

– A company may be different from a certain pattern and thus find itself in the focus of attention, but this differentiation may not be an indicator of potential insolvency.

The model is supplemented when the work on conducting the (aforementioned) real-life tests begins. The model receives feedback on whether it is behaving appropriately and can be taught further, if necessary. Thus, as was discussed at the meeting of April 12, the model may be less accurate at first, but its level of accuracy can be increased in the course of further testing. In the course of the pre-testing preparation of the model, it is intended to take the prototype to a level at which testing would immediately provide positive outcomes; however, in the case of the positive scenario, the model will continue to be adjusted in the course of testing and based on the testing results when the tests have been completed.

3.4. Technical solutions for the use of the prototype

Technical solutions were worked on in April 2022 for the prototype to be usable, i.e. above all testable, and if the test indicate an appropriate result, it can be used as a basis for creating an information system later.

The cooperation agreement does not call for creating a user interface (and this position is held). Thus, the model is not visible to or usable by third parties (even those who monitor the work and are directly involved in testing) via a network address. Building a user interface, especially in a secure form, on data which are subject to confidentiality requirements is work-intensive and creating a user interface has not been deemed important, as it has been considered necessary to use the man-hours for developing the model itself.

On the other hand, there has been the need for the prototype model to be usable and testable (which has also been worked on). The initial or baseline solution suggested for testing the prototype is a solution in which the data tested are sent to Statistics Estonia and the people in tasked with the development of the prototype let the prototype model to assess what the model thinks of the data. In this case, the format of receiving the input is not important, the people working on the prototype can make the format of the data suitable for the prototype.

Another solution (in addition to the basic solution described above) still considered on the side of the development of the prototype is to create a solution in which authorised users (the people monitoring the development of the prototype and the people related to the testing) can enter the data tested through an internet browser via a user interface and receive a response from the model (i.e. are the data used to issue a certain warning and if so, what kind of a warning with appropriate explanations). A user interface is not deemed part of the basic version of the system due to the reasons described above, but taking into consideration the fact that a user interface would significantly increase the efficiency of testing (the system can learn as a result of testing, the more testing, the better the system), the user interface is actively worked on (the user interface was actively worked on in April 2022 and the work will continue in May 2022).

3.5. Use of the data

Development of the prototype began based on balance sheet and income statement data. The balance sheets and income statements which could indicate issues (or potential issues) in the context of decreasing of the solvency were identified. The structures of balance sheets and income statements were analysed by using the machine learning method, as well as verified based on classic economic analysis ratios (the ratios used are provided in Annex 2 to the material).

The data of B2B transactions was introduced to the prototype creation and development as another approach. The analysis models were based on who and how much, how frequently, in which amounts, and with which patterns was making payments and to whom, how many transaction partners to the parties have and how does this number of transaction partners change in time or by examining certain subcategories of the data (e.g. only large payments, etc.).

While the first phase of creating the prototype was mainly focussed on the balance sheet and income statement data, the development of transaction data models was focussed on in April 2022, which would enable using relatively brief transaction data to draw substantive conclusions on changes in the solvency of a company (in a way, transaction data provide less information than specific balance sheet data, for example – for example, a high volume of liabilities or an increasing volume of liabilities may indicate very specifically that solvency is decreasing; on the other hand, a decrease in the number of transaction partners may not mean that the market is disappearing, the company is simply concentrated on a certain customer segment, the volumes of the transactions with the customers must also be specified). It is much more difficult to obtain reasonable information from transaction data compared to balance sheet data. On the other hand, transaction data are basically available on a monthly basis, while the balance sheet and income statement data from annual reports often become available with a delay of a year or a year and a half today (in this context, it is hard to discuss ‘early warning’ – the data are simply too old for this).

If the real-time economy solutions that are being developed in Estonia start providing solutions which will bring annual reports (or components thereof) into databases in real time (or with a delay on one calendar month), it will be possible to issue early warning signals based on the data. As long as this has not happened, it is important to do in-depth work with transaction data. In the context of this prototype, it is important to stress that on the one hand the prototype is designed to contribute to the development of real-time economy (by using transaction data, for example); on the other hand, though, the developments of real-time economy solutions are important for the project (this would provide more adequate input for the prototype and the future information system for issuing warnings).

As of April 2022, export data are also tested in the model, in addition to the annual report data. The share of the export turnovers of some sectors is considerably high and thus, it is not possible to draw adequate conclusions about any changes in the solvency of those companies merely based on the turnover generated in Estonia, yet, the prototype (and the future information system) must also be capable of issuing early warnings to the companies which are largely involved in exporting their production/services.

Labour market data is an important data input which is now being included in the prototype and were first tested in April 2022. An increase or decrease in the number of employees:, the capability of the company to pay compatible wages, and the qualifications of the employees hired (previous experience, education) are the indicators which include statically and dynamically (if those indicators change) information on the solvency of companies.

In April 2022, adding the data of tax arrears as an input to the prototype model was also worked on. Yet, no access to the data on tax arrears has yet been gained within the framework of the work and the data have thus not yet been added to the model. On the other hand, should this opportunity arise when the development of the model is inevitably still ongoing, the data on tax arrears will also be included. Tax arrears generally indicate reduced solvency (if there are no issues with solvency, tax arrears do not normally rise, companies fail to pay their taxes because they do not have sufficient funds, not because they choose to delay the payments) and the dynamics of the changes in tax arrears allows assessing changes in the solvency of the company.

The work on creating the prerequisites for including third data also began in April 2022. Third data mean the following:

- address data: the possibility to teach the prototype location-specific circumstances (such as the tourism in Tallinn is different from the tourism in Põlva, Pärnu, or Kuressaare, which are specific tourism cities);
- weather data: the possibility to teach the prototype the circumstances arising from the specifics of the weather; for example, from the perspective of agricultural production, it is important to know whether the production occurs in Järva County, on the islands, or in South-East Estonia;
- population data: the possibility to teach the prototype where people live, where there are more births, where there are more deaths – this enables showing how the market forms and include new indicators based on this;
- electricity data: the electricity consumption data by all metering points which are available with the accuracy of one hour enable assessing economic activity (living activity) in real time, which allows taking a qualitative step forward towards obtaining real-time inputs in the context of real time economy; those inputs can be added to the prototype as important parameters.

3.6. Elimination of errors

Elimination of errors was also worked on in April 2022. In this context, errors primarily mean incorrect interpretations by the model or inadequate solutions to the model. Both types of errors are inaccuracies that occur inevitably in the course of work, but those errors must be

eliminated to obtain a functioning prototype which can be used to create an information system (the occurrence of such errors must be monitored constantly and the errors must be eliminated if detected).

One certain type of a problematic issue is a situation in which the model takes a certain input from the information offered as input in a disproportionately large extent and the results will be slanted towards this input (without any objective economic reason for this). For example, in the case of transaction data, a situation arose where the year was assigned too much significance, resulting on the conclusion (which is clearly incorrect) that 'if you had a turnover in 2017 and at the beginning of 2018, you have a potential risk of becoming insolvent in 2022.'

Another type of a typical issue is a situation in which defining a sector and then separating more and less problematic sectors based on balance sheet analysis and transaction data results in the amount of input data left to teach the respective model to the machine being too small.

SUMMARY

This work involved developing a prototype and conceptual solutions for launching an early warning system in Estonia. The material presents the initial opinions and principles by which the early warning service could and should be guided. The work done, the most commonly found issues in the course of the work, and the manner of solving the issues are described. Explanations are also provided on why exactly this path was chosen to develop the prototype.

The summary below highlights several key opinions which must be taken into consideration in the case of the early warning service developed and which the further work on introducing the early warning system in Estonia will be based on.

Testing the prototype

The prototype of the early warning service has been conceptually developed and has passed initial tests. In the current phase, it is now possible to start conducting real-life tests. Real-life tests mean that entrepreneurship consultants (Harju Economic Development Center) send to the prototype the economic data of actually operating companies and the prototype provides responses on whether or not to issue an early warning, if so, then which warning to issue and, in the case of issuing a warning, explanation of why the warning was issued. Entrepreneurship consultants conduct an interview with the company and thereby check whether the warning issued (if it was issued) is appropriate for the company. In the course of the interview, feedback is received on what to use for further improvement of the prototype.

The method of testing described here was worked out based on discussions on the prototype and the authors of this work believe that it is an efficient and appropriate method of testing the prototype. They also believe that the Harju Economic Development Center is a competent and appropriate organisation for conducting the testing. On the other hand, if testing the prototype reveals another method or if it is feasible to use another, additional method in addition to the one described, the authors are prepared to consider alternative testing options.

Developing the information view for consultants

It has been suggested in the discussions on the prototype that if the early warning system issues a warning, it should also issue an explanation of why the warning was issued and what to do with the warning. It has been claimed that a business may not be competent (businesses are of very different backgrounds) to interpret the content and context of the message and it would be important to involve a consultant.

On the other hand, if a situation arises in which a consultant should be involved, it would be reasonable to add a further information window for the consultant (the consultant view, a commented summary for the consultant, etc.), which would help the consultant to explain to the business what their issue consists of and how it could be solved, what to do in order to solve the issue, etc. At this point, it is believed that further work should be done to ensure access to further information if a company requires the assistance of a consultant in interpreting a warning. In this case, the consultant should be able to see the entire information sent to the company and perhaps also receive a further explanatory input about additional elements (from the legal perspective, must the company grant permission for this?).

Thereat, it is important to stress that early warnings are only sent to the authorised persons at companies. Involving consultants can be possible if the authorised person of the company has expressed their interest in this and grants the respective permission. Involving consultants is an option, not an automatic accompanying process. Involving consultants is above all thinkable and necessary in more complicated situations – in certain cases, the nature of the early warning is very clear and unambiguous; in other cases, however, it is harder to understand the content of the warning and it is also harder to understand what should be done to solve the root problem of the early warning.

Development of the prototype

The prototype was developed with an aim of serving as the basis for the development of an early warning information system. This was kept in mind in developing the prototype and the work has been based on this. On the other hand, several ideas and perspectives have come up in the course of developing the prototype (and during discussions), which could also be included in the further development of the prototype and could become important components of the functioning information system. There are currently two main ideas.

First, the prototype and the later information system should be developed so that if the system issues an early warning to a company, the system can also issue scenarios to the company, describing how the insolvency would deepen and what situation the company would find itself in if the company behaved in a certain manner, for example, as well as how the solvency would improve if the company acted in another manner. The purpose of early warning is to make businesses think and draw attention to issues, thereby ensuring a situation in which the general solvency level of the business sector improves. The scenarios for further action help the company to think about and understand what exactly they should do to solve the problems once they have received a warning (the warning may come with an explanation of why it was issued, but not all companies can predict further development scenarios based on such explanation yet). Early warning models include a lot of important information and more information is added in the course of the work, making them a good foundation for suggesting future scenarios.

Secondly, the prototype can be developed further towards developing the view of the consultant, i.e. what to add to the view of the consultant and which additional tools to give to the consultant for the consultant to be able to support the company in the best possible manner and for the early warning thereby ensuring maximum practical benefits.

In the light of the two ideas described above, the further work should be organised so, if possible, that further development of the prototype in the aforementioned directions occurs in parallel with the development of the information system. These two works (creating the information system and further development of the prototype, in parallel) would not disturb one another and could be done simultaneously. First, the information system will be launched in the so-called basic version, with additional components added later.

Use of the early warning solution in other countries

The issue of introducing an early warning system is not only relevant in Estonia, but in other countries as well. The solutions to be developed within the framework of this work is as much as possible based on the latest developments of information technology (machine learning, artificial neural networks, etc.). This approach to the issue of an early warning service is innovative and therefore other countries may not be on the brink of launching similarly solved services in their respective work. Even if a certain country has used artificial neural networks or machine learning for creating similar models, their work may result in different models (models are trained based on specific data, inputs are defined, etc.; very different development paths can be chosen in this respect).

Based on the above, the authors believe that the early warning prototype to be developed and the information system based on it may be products which arise interest in other countries and Estonia will be able to provide this information as its contribution into international cooperation.

The models trained and the information technology framework required for the work thereof are basically computer software which can be scaled or distributed (adaptable, if necessary). There are slight differences between different countries, but the issues which require solving are generally similar.

From the perspective of Statistics Estonia, it may be stated here that there is experience with the statistical data of different countries (primarily through cooperation with Eurostat) and it can be said based on this experience/knowledge that the model can be fed with data from very different countries.

Based on the above, it is very important to state here that other countries also need a similar service like Estonia and, keeping in mind the work done within the framework of this project, we find ourselves in a situation in which Estonia can offer its solution to the partner countries.

If the idea of introducing the work done within the framework of the project to other countries is continued, there are plans to develop a website for this purpose where the work done within the framework of the project and the solutions ensured by the software created as a result of the project (including how the software could be adjusted to the peculiarities of a specific country) could be explained in an international language.

Development of real-time economy

In the course of conducting the project, several paths led to the conclusion that real-time economy and the related development directions were very important for the implementation of the early warning concept and for its systematic deployment. The early warning solution will support the operations of real-time economy, but real-time economy solutions are also a very important input in the introduction and development of the early warning service.

The individuals in charge for the real-time economy approach who have monitored the development of this project and given advice have provided a lot of help and support to make the early warning system viable. In the further development of the early waning service, it is important to retain the competent support from the individuals in charge for the real-time economy approach, which has been enjoyed so far.

Development of the information system

The prototype will be followed by creating an information system. At this point, the authors believe that the following source logic could be considered for building the information system. Statistics Estonia could be the developer of the information system (the development of the prototype shows that the data must be regularly examined in depth, regular reality tests must be conducted to understand if the machine is drawing correct conclusions – the people of Statistics Estonia can monitor the data). The Ministry of Justice could be the owner of the system, as the early warning levels will be aligned with legally defined levels of insolvency (in which cases there is simply a risk, when to initiate reorganisation, when to file a bankruptcy application).

A representative of the Ministry of Justice has provided very competent and relevant comments/instructions in the course of this project which have greatly helped to shape the prototype so that it will not be necessary to develop ‘two legal environments’ in Estonia in parallel, but the system created will be aligned with other legal practice as much as possible. Based on the above, it is suggested here to consider a solution in the case of which the Ministry of Justice would be the owner of the early warning system to be developed.

The source logic presented here is meant to serve as a basis for discussions, not a final opinion.

Further schedule

Some timelines for moving on with the projects are provided below: Testing the prototype: May–June

2022

Analysis of the results of testing the prototype: June–July 2022

Making changes in the prototype as a result of testing the prototype: June–July 2022

Proposing a plan for developing the information system based on the prototype: July–

August 2022

Beginning of the development of the information system: October 2022

Completion of the information system: by the end of 2023

Proposing a plan for a solution for making the system scalable: July–August 2022

Proposing a plan for further development of the prototype: July–August 2022

Beginning of the further development of the

prototype: September 2022

Presentation of the developed prototype: February

2023

It is planned to develop the prototype further in parallel with developing the information system and the process should be completed so that the outcomes of the further development of the prototype could be involved in the development of the information system.

The deadline in the schedule above are indicative, designed for discussing and changing, if necessary.

Annexes

The following Annexes are enclosed to this material:

Annex 1. Definition of sectors with EMTAK

Annex 2. The economic ratios used
Annex 3. The companies gone bankrupt based on clusters and years
Annex 4. Further clustering
Annex 5. Reducing of the clusters to two parameters
Annex 6. Clustering into a two-dimensional system with the PCA technique Annex 7. Correlation between the variables
Annex 8. Division into clusters by areas of activity
Annex 9. The TwoLayerNet code
Annex 10. The
Layers code Annex
Annex 11. The Optim
code

References

Aliaj, T., Anagnostopoulos, A., Piersanti, S. (2020). Firms Default Prediction with MachineLearning. Bank of Italy (2020).

Amel-Zadeh, A., Calliess, J.-P., Kaiser, D., Roberts, S. (2020). Machine Learning-Based Financial Statement Analysis. Oxford-Man Institute of Quantitative Finance, University of Oxford (2020), pp. 1-55.

Andrés, J., Lorca, P., Bahamonde, A., Coz, J., J. (2004). The Use of Machine Learning Algorithms for the Study of Business Profitability: A New Approach Based on Preferences. *The International Journal of Digital Accounting Research.* Vol 4, N 8 (2004), pp. 99-124.

Cao, K., You, H., (2021). Fundamental Analysis via Machine Learning. Hong Kong University of Science and Technology. (2021), pp. 1-62.

Chen, J., M. (2020). An Introduction to Machine Learning for Panel Data. Michigan State University (2020), pp. 1-67.

Choudhry, M. (2018). Machine Learning in Banking: How To Transform both Balance Sheet Management and Customer Service Provision. *The European Financial Review.* (April-May 2018), pp. 2-8.

Cialone, G., (2020). Bankruptcy Prediction by Deep Learning. Stanford University CS230 Winter 2020.

Kou, G., Xu, Y., Peng, Y., Shen, F., Chen, Y., Chang, K., Kou, S. (2021). Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. *Decision Support Systems.* No 140 (2021), pp. 1-14.

León, C., Moreno, J., F., Cely, J. (2016). Whose Balance Sheet is this? Neural Networks for Banks' Pattern Recognition. *Borradores de Economía.* Núm 959 (2016), pp. 1-34.

Perboli, G., Arabnezhad, E. (2021). A Machine Learning based DSS for mid and long-term company crisis prediction. *Expert Systems With Applications.* No 174 (2021), pp. 1-12.

Petropoulos, A., Siakoulis, V., Vlachogiannakis, N., Stavroulakis, E. (2019). Deep-Stress: A deep learning approach for dynamic balance sheet stress testing. Bank of Greece. (2019), pp. 1-22.

Qu, Y., Quan, P., Lei, M., Shi, Y. (2019). Review of bankruptcy prediction using machine learning and deep learning techniques. *Procedia Computer Science.* No 162 (2019), pp. 895-899.

Shi, Y., Li, X. (2019). A bibliometric study on intelligent techniques of bankruptcy prediction for corporate firms. *Heliyon.* No 5 (2019), pp. 1-12.

Ucoglu, D. (2020). Current machine learning applications in accounting and auditing. [redacted] Vol 12 (2020), pp. 1-7.

Annex 1: Definition of sectors with EMTAK

The medical sector:

Name	EMTAK
Hospitalisation services	EMTAK 86101
Provision of general medical treatment	EMTAK 86211
Provision of specialised medical treatment	EMTAK 86221
Provision of dental treatment	EMTAK 86231
Activities of emergency medical staff and paramedics	EMTAK 86901
Activities of medical laboratories	EMTAK 86902
Provision of nursing care	EMTAK 86903
Activities of midwives	EMTAK 86904
Activities of sanatoriums	EMTAK 86905
Residential nursing care activities	EMTAK 87101
Activities of diagnostic cabinets and centres	EMTAK 86906
Residential care activities for mental retardation and mental health	EMTAK 87201
Residential care activities for substance abuse	EMTAK 87202
Residential care activities for the elderly and disabled	EMTAK 87301
Activity of institutions providing alternative care service	EMTAK 87901
Other healthcare activities not classified elsewhere	EMTAK 86909
Retail sale of medical and orthopaedic goods in specialised stores	EMTAK 47741
Manufacture of other medical and dental instruments and supplies	EMTAK 32509

The sector of financial services:

Name	EMTAK
Central banking	EMTAK 64111
Credit institutions (banks)	EMTAK 64191
Activities of holding companies	EMTAK 64201
Trusts, funds and similar financial entities	EMTAK 64301
Financial leasing	EMTAK 64911
Pawn shops	EMTAK 64921
Other credit granting, except pawn shops	EMTAK 64929
Other financial service activities, except insurance and pension funding	EMTAK 64991
Life insurance	EMTAK 65111
Non-life insurance	EMTAK 65121
Reinsurance	EMTAK 65201
Pension funds	EMTAK 65301
Administration of financial markets	EMTAK 66111
Security and commodity contracts brokerage	EMTAK 66121
Currency exchange	EMTAK 66129
Financial counselling	EMTAK 66191
Other activities auxiliary to financial services that are not classified elsewhere	EMTAK 66199
Risk and damage evaluation	EMTAK 66211
Activities of insurance agents and brokers	EMTAK 66221
Other activities auxiliary to insurance and pension funding	EMTAK 66291
Fund management activities	EMTAK 66301

The real estate sector:

Name	EMTAK
Development of building projects	EMTAK 41101
Construction of residential and nonresidential buildings	EMTAK 41201
Construction of roads and motorways	EMTAK 42111
Construction, maintenance and repair of railways and underground railways	EMTAK 42121
Construction of bridges and tunnels	EMTAK 42131
Construction of utility projects for fluids	EMTAK 42211
Water well drilling and liquidation	EMTAK 42212
Construction of utility projects for electricity and telecommunications	EMTAK 42221

Construction of water projects	EMTAK 42911
Construction of other civil engineering projects	EMTAK 42991
Demolition	EMTAK 43111
Site formation and clearance work	EMTAK 43121
Drainage work and amelioration, including drainage of agricultural or forestry land	EMTAK 43122
Other earth and soil works	EMTAK 43129
Test drilling and boring	EMTAK 43131
Installation of electrical wiring and fittings	EMTAK 43211
Installation of fire and burglar alarm systems	EMTAK 43212
Installation of telecommunication wirings and antennas	EMTAK 43213
Installation of heating, ventilation and air conditioning equipment	EMTAK 43221
Installation of plumbing and sanitary equipment	EMTAK 43222
Insulation work activities	EMTAK 43291
Other construction installation	EMTAK 43299
Plastering	EMTAK 43311
Installation of doors, windows and staircases of wood or other materials	EMTAK 43321
Other joinery installation	EMTAK 43329
Floor and wall covering	EMTAK 43331
Painting and glazing	EMTAK 43341
Other building completion and finishing	EMTAK 43391
Roofing activities	EMTAK 43911
Ground works, concrete works and other bricklaying works	EMTAK 43991
Pottery works, construction work of chimneys and fire places	EMTAK 43992
Erecting and dismantling of scaffolds and work platforms	EMTAK 43993
Other specialised construction activities	EMTAK 43999
Buying and selling of own real estate	EMTAK 68101
Rental and operating of own or leased real estate	EMTAK 68201
Real estate agencies	EMTAK 68311
Management of buildings and rental houses	EMTAK 68321
Management of gardening and cottage associations, etc.	EMTAK 68322
Other real estate management or related activities	EMTAK 68329

The education sector:

Name	**EMTAK**
Activities of creches	EMTAK 85101
Activities of nurseries	EMTAK 85102
Activities of nursery-elementary schools	EMTAK 85201
Activities of elementary schools	EMTAK 85202
Activities of nursery-basic schools	EMTAK 85311
Activities of basic schools	EMTAK 85312
Activities of general upper secondary schools	EMTAK 85313
Activities of vocational educational institutions	EMTAK 85321
Activities of professional higher education institutions	EMTAK 85411
Activities of academic higher education institutions	EMTAK 85421
Sports schools	EMTAK 85511
Other sports and recreational education	EMTAK 85519
Music and art education	EMTAK 85521
Activities of dance schools	EMTAK 85522
Other hobby education	EMTAK 85529
Driving school activities	EMTAK 85531
Language training	EMTAK 85591
Computer training	EMTAK 85592
Other education not classified elsewhere	EMTAK 85599
Educational support activities	EMTAK 85601
Research and experimental development on biotechnology	EMTAK 72111
Other research and experimental development on natural sciences and engineering	EMTAK 72191
Research and experimental development on social sciences and humanities	EMTAK 72201

Annex 2: The economic ratios used

Ratio/formula used to find the ratio	Explanation of the function of the ratio	Interpretation of the ratio / what does it show
RATIOS DESCRIBING LIQUIDITY		
The liquidity ratios show the freedom of the position of the company from the financial perspective; this provides the basis for an initial assessment on the likelihood of the company experiencing financial difficulties. Financial difficulties primarily materialise through how the company is able to fulfil its obligations to external partners and thus, it is important to examine how the liquidity of the company helps in this context. Based on the above, liquidity ratios are examined first.		
Working Capital *Working capital = current assets – short-term liabilities*	Shows the actual amount of money which the company can use for its daily economic operations.	The more working capital a company has, the better. Whether or not the amount of working capital is sufficient can be assessed together with other economic indicators.
Working Capital Turnover *Working capital turnover = sales revenue / working capital*	Shows how many times the company uses its working capital over the year.	The figure should remain between 2–10, preferably between 5–8. The figure shows how many times working capital is turned over a year. The higher the number, the more efficiently the company uses its working capital, but also the more vulnerable the company if the working capital should be lost or if its amount is reduced (however, a relatively lower amount of money is sufficient to help the company out).
Current Ratio *Current ration = current assets / short-term liabilities*	Shows the level of solvency in terms of the extent by which amount current assets exceed the amount of current liabilities.	A figure in the range of 1.0–1.5 should be deemed rather weak (an even lower figure is very weak). The range of 1.5–2.0 is deemed strong with no issues with paying debts generally observed. A higher figure is even better, but may in turn indicate overcapitalisation (inefficient use of capital).
Quick Ratio *quick ratio = (current assets – stocks – advance payments) / current liabilities*	Shows the short-term liquidity of the company, i.e. how the company is capable of satisfying its current liabilities by using its most liquid assets. This figure is important if it becomes necessary for the company to pay all its current liabilities.	If the quick ratio is in the range of 0.9–1, the company is deemed to be able to service its current liabilities without any issues. A figure over 1 is very good. On the other hand, if the figure is too high, the financial assets are not being used efficiently.
Cash Ratio *cash ratio = (cash + short-term financial investments) / current liabilities*	Shows how many short-term liabilities the company can cover almost immediately. A good indicator for assessing the risk of insolvency of a company if something unexpected happens and the liabilities which were due further in the future must be paid immediately.	A cash ratio between 0.5–1 is deemed normal. A higher figure is very good. If the figure exceeds 2–3, the company is overcapitalised and the financial assets are being used inefficiently.
RATIOS DESCRIBING THE CAPITAL STRUCTURE		
The ratios describing the capital structure show the structure of the placement of the capital of the company. The ratios allow looking into the company, understanding the actual market capability thereof, and assess in the context of the specific industry whether the company may develop financial difficulties. The capital structure ratios provide a so-called framework for the liquidity indicators described above.		
Debt Ratio *debt ratio = total liabilities / total assets*	Shows the extent to which external capital has been used to obtain the existing assets of the company. A good indicator for assessing the so-called general creditor risk and the risk of potential payment difficulties arising therefrom.	The assessment to the indicator largely depends on which sector the company is operating in, how intensively borrowing is used in the sector in general, the stability of the sector, and the type of the liability. In some cases, even 30% is a high figure; in other cases, this figure may rise to up to 80% without substantial issues.
Debt to Debt plus Equity *Debt to debt plus equity = non-current liabilities / (non-current liabilities + equity)*	Shows how aggressively the company has taken loans and the level of risks arising from this borrowing. Differs from the debt ratio by being directly focussed on the assessment of loan liabilities (the debt ratio examines liabilities/external capital in a wider perspective). Creditors can often most directly influence what is going on at the company.	The higher this indicator, the higher the financial risk. Creditors may decide to recall loans prematurely, which may occur automatically due to business circumstances related to third parties.

Interest Coverage Ratio *interest coverage ratio = (operating profit + financial revenues) / financial expenditure*	Shows the capability of the company to cover its interest expenses from operation profit, i.e. how many times the company can pay its interest expenses from the current cash flows earned. The size and dynamics (the direction of its development over different periods) of the indicator show whether the company is about to be faced with liabilities which it cannot cover, thereby becoming insolvent.	The higher this indicator, the better. If the indicator is too low, servicing the financial expenses of the company takes up most of the operating profit, creating a situation in which the company is 'drowning in debt'.

Debt to Equity Ratio *debt to equity ratio = total liabilities / equity*	Shows the share of using external capital against equity, which expresses the extent of the loan risk against equity. The indicator enables checking how the risk from external creditors is expressed with respect to equity and how it may indicate potential insolvency.	Based on a general opinion, 2.0 is a poor indicator and 1.0 is a good indicator. Thus, the higher the value of the indicator, the worse it is; and the lower, the better it is. The context of the sector is important; some sectors use debt for their operations more extensively than others.
Equity to Assets Ratio *equity to asset ratio = equity / total assets*	Shows the share of the assets belonging to the owners of the company in all assets of the company. The lower this indicator, the less control the owners have over the company from the financial perspective and the more the company can be influenced by external factors. Enables assessing the extent to which the risk of insolvency is in the hands of people.	The higher the indicator, the better. However, if the value of the indicator is 100% or close to this level, this means, in principle, that no external resources have been used at all, which usually indicates unreasonable management.
RATIOS DESCRIBING PROFITABILITY		
The profitability ratios show the extent to which the company is productive to develop, grown, and operate sustainably. The higher the profitability of the company, the likelier that the company is able to solve its financial problems independently if finding itself on the threshold when it comes to solvency.		
Net Profit Margin *net profit margin = net profit / sales revenue*	Shows the share of net profit in the sales revenue, i.e. the amount of net profit for the company from each euro of the turnover of the company, the share of the company in this turnover. Enables analysing how changes (growth) in the turnover lower the risk of insolvency; and which curve it is based on.	The higher the indicator, the better. There is no maximum limit (i.e. it can be 100%, theoretically). The 'normal' value of the indicator depends on the sector, but the range of 10–15% is generally assessed as positive.
Return on Assets *return on assets = EBIT / total assets*	Shows the productivity of the assets, i.e. the productivity of the means for which the assets were acquired. Enables assessing the risk of insolvency against assets.	The higher the return on assets, the better. Depending on the sector and the nature of the assets, excessive burdening of the assets may result in the so-called 'risk of breaking', which would in turn party suspend operations.
Return on Equity *return on equity = (net profit – dividends from preference shares) / average equity*	Shows the efficiency of the use of equity, i.e. the efficiency of the functioning of the equity. Enables assessing the role of the productivity assigned to equity in preventing potential insolvency, in addition to the productivity of assets.	The higher the return on equity, the better. On the other hand, if the share of equity in the balance sheet is low, more assets (if other conditions remain the same) (i.e. loan leverage) may increase the return on equity, but the negative side of this rise is excessive loan leverage.
RATIOS WHICH DESCRIBE PRIMARY EFFICIENCY		
The ratios describing primary efficiency enable taking a closer look into the company, to check if the daily operations of the company have been launched efficiently, if there are any significant deficiencies in the daily activity, the causes for other (described above) negative (unfavourable) ratios. The primary efficiency ratios are examined last from the perspective of potential insolvency, as the general approach is to move from general to an individual, to first obtain a general vision and understanding of the company and then move on to the level of details.		
Accounts Receivables Turnover *accounts receivables turnover = sales revenue / average volume of receivables*	Shows how many times the sales revenue exceeds the volume of the receivables (primarily trade receivables in connection with unpaid invoices). Shows the efficiency of the activity of the company in connection with receiving money for the sales operations completed (which could be influenced by disputes over the quality of a product/service). The indicator may also characterise the market capability of the company from the substantive perspective.	The higher the indicator, the better, as this indicates that cash is received efficiently for services/products. Depends significantly on the payment routine of the sector.
Receivables Average Collection Period *receivables average collection period = 360 / turnover rate of receivables*	This is a derivative of the turnover rate of receivables, showing how much time it takes for the company to receive money from the customers, on average. If the customers fail to pay, this may cause payment difficulties.	The indicator is expressed in days. The indicator should not be much higher than the terms of payment granted (there are always some delays; the indicator will never be exactly as long as the terms of payment).
Inventory Turnover *inventory turnover = expenses on production sold / average stocks*	Shows how many times the stocks of the company are sold, i.e. the efficiency of using the stocks. The stocks are the production to be left in the warehouse and requiring realisation. If the company produces to the warehouse and the client fails to buy the production, the company may encounter financial difficulties.	The higher the indicator, the better, the lower the amount of products in the warehouse, and the amount of money under those products.
Inventory Collection Period *inventory collection period = 360 / inventory turnover*	This is a derivative of the inventory turnover which shows how many days it takes for the stocks to be marketed. If marketing the stocks of a company takes a long time (i.e. selling the production produced to the warehouse takes a long time), the risk of insolvency may arise.	The lower this indicator, the better, as the lower the amount of stocks in the warehouse and the faster money is received for the production (if other conditions remain the same).

Operating cycle *operating cycle = inventory collection period + receivables average collection period*	The indicator shows the period in which (the duration of the period) the company is able to market its inventories and receive money from the clients. The shorter the period, the less likely the financial difficulties of the company.	The indicator is expressed in days, it is very good if this indicator is slightly higher than the normal payment term granted in the sector (the indicator may fall under the payment term; and can be approached as the bottom threshold).
Accounts Payable Turnover Ratio *accounts payable turnover ratio = cost of the goods sold / average current liabilities*	The indicator shows the efficiency of servicing current liabilities from the perspective of the cost of the goods sold. That is how many times are the current liabilities paid within the reporting period. The more capable the company is of paying its current liabilities, the lower the risk of insolvency.	The higher this indicator, the better. The optimum (average) level strongly depends on the peculiarities of the sector.
Payables Average Settlement Period *payables average settlement period = 360 / accounts payable turnover rate*	This is a derivative of the accounts payable turnover rate, which shows how many days it takes for the company to pay its current liabilities. The faster the company can pay its current liabilities, the less likely the company is to find itself in financial difficulties.	The lower the indicator, the better; the quicker the company is able to pay its current liabilities, if necessary. A too low figure, however, indicates that the company has very few current liabilities or is operating in the conditions of excess cash (excess cash means that the same amount of money could be used to operate much extensively – the competitors could use a lower amount of money to operate as extensively).
Funding cycle *funding cycle = operating cycle – payables average settlement period*	The indicator shows the number of days in which the company must find further financial resources for operating. The lower the number, the lower the risk of insolvency.	The optimum level depends on the sector, but the indicator may differ significantly in the case of one specific company with no need for concern. However, a negative cycle would indicate excess cash, which means that the same amount of money could be used to do more business.
Assets Turnover Ratio *asset turnover ratio = sales revenue / average total assets*	The indicator shows how efficiently the assets of the company are being used. The indicator is relatively general from the perspective of primary efficiency (total turnover vs. total volume of assets), and provides the most summarised overview of the efficient use of the assets; there are no nuances in the way.	The more efficient the use, and provided that the assets are used for proper purposes, the better the higher efficiency – i.e. the more cash is earned by using the assets, the better. The higher this indicator, the better the financial situation, the faster the situation may improve, and the lower the risk of insolvency.

Annex 3: Number of bankrupt companies by year

	Cluster	Companies
2009	1	192
	2	4
2010	1	129
	2	1
2011	1	131
	2	2
2012	1	107
	2	1
	3	1
2013	1	99
	2	2
	3	2
2014	2	1
	3	157
2015	2	1
	3	89
2016	3	102
	4	3
2017	0	1
	3	120
	4	2
2018	0	114
	4	3
2019	0	71
	4	2
2020	0	48
	4	4
2021	0	3

Annex 4: Further clustering

Table L1: Average, K-means: code, EMA_BI_ID-d not taken into consideration in the clustering. PERIOD_NM by logarithms before clustering (the dates should be made comparable for the machine). BI_100 – BI_40 indicators normalised with MinMaxScaler before clustering. The ratios were calculated later.

Table L2: Medians, K-means: code; EMA_BI_ID was not taken into consideration in clustering. PERIOD_NM by logarithms before clustering (the dates should be made comparable for the machine). BI_100 – BI_40 indicators normalised with MinMaxScaler before clustering. The ratios were calculated later.

Table L3: Average, all the same as in L1, L2, but the 'outliers' were removed which the clustering algorithm did not like, in general. Z-score = 3, i.e. 99.7% of the data remains, 0.3% of the outliers are left out.

Table L4: Medians, all the same as in L1, L2, but the 'outliers' were removed which the clustering algorithm did not like, in general. Z-score = 3, i.e. 99.7% of the data remains, 0.3% of the outliers are left out.

Table L1:

clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	174,950	213,481	405,298	436,558	116,311	82,703	126,130	312,141	226,825	14,372	48,832	44,403
1	174,383	266,334	447,299	447,716	129,206	101,252	119,751	360,983	216,132	17,151	71,777	61,763
2	325,517	486,705	914,517	834,326	235,214	168,224	261,025	739,489	504,326	20,947	64,214	60,113
3	247,690	236,418	492,256	570,159	129,936	140,719	197,929	388,452	328,450	14,812	50,645	45,265
4	184,973	257,111	439,333	463,400	127,089	104,896	128,688	366,798	233,655	17,984	63,250	56,165

Table L2:

clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	10181.00	6666.00	8683.00	17696.0	4230.0	2916.00	4272.0	14190.00	6267.00	2556.00	3973.0	3196.00
1	9968.00	8986.00	6679.50	15836.0	4708.0	3778.00	3146.0	16547.00	4240.00	2500.00	5571.0	4457.00
2	9750.55	6353.17	10147.51	17954.0	5603.0	3604.36	6328.0	17707.43	9250.44	2556.47	3025.0	2684.29
3	10062.00	6205.50	9250.00	17938.0	4200.0	2906.00	5039.0	14274.50	7321.00	2556.00	3535.0	2909.00
4	10927.00	9291.00	8333.00	18282.0	5145.0	4274.00	4100.0	17837.00	5819.00	2508.00	5317.0	4103.00

Table L3:

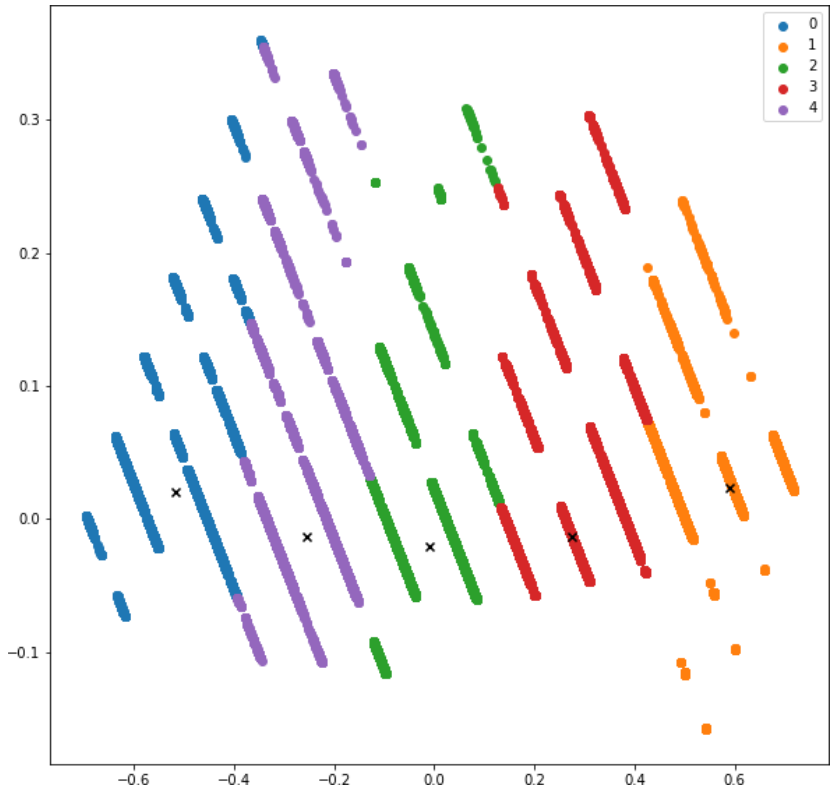
clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	129,446	130,841	275,062	305,567	79,277	63,094	98,388	228,106	176,836	8,644	27,760	26,318
1	132,930	157,411	281,662	311,089	78,576	72,622	85,604	261,983	159,131	7,847	44,924	39,380
2	126,057	129,782	272,419	301,666	70,715	59,797	84,892	226,328	156,247	8,047	34,827	30,835
3	123,261	159,042	270,417	288,120	75,935	67,832	76,424	254,436	143,102	7,005	50,171	43,088
4	130,621	125,326	275,445	310,881	71,027	59,857	91,851	215,041	163,163	8,564	30,264	27,053

Table L4:

clusters	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2
0	9,671	6,283	10,000	17,785	5,520	3,564	6,262	17,379	9,128	2,556	3,010	2,684
1	10,855	9,170	8,198	18,144	5,094	4,226	4,060	17,591	5,757	2,508	5,271	4,068
2	10,118	6,600	8,571	17,584	4,200	2,890	4,234	14,043	6,204	2,556	3,951	3,179
3	9,902	8,862	6,567	15,718	4,638	3,733	3,116	16,292	4,194	2,500	5,524	4,417
4	9,998	6,148	9,135	17,813	4,150	2,877	5,000	14,065	7,239	2,556	3,516	2,895

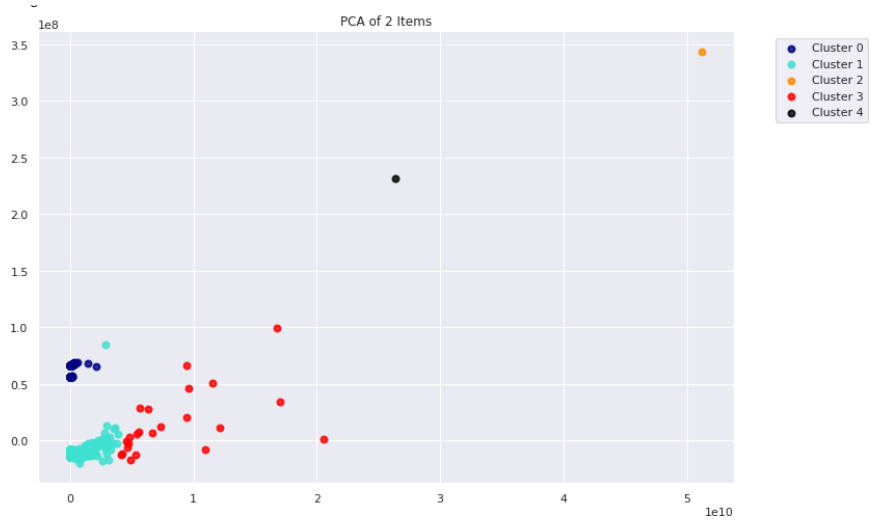
Annex 5: Reducing of the clusters to two parameters

The figure presents a situation in which reducing twenty parameters to two parameters is presented graphically to be able to display them in a two-dimensional figure. The clusters are marked with different colours; the centroids of the clusters are marked with black x's.

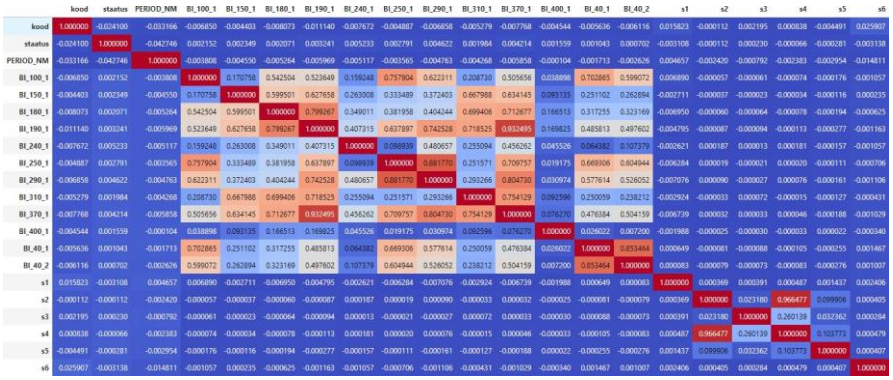


Annex 6: Clustering into a two-dimensional system with the PCA technique

In the figure, the seven indicators achieved with the PCA methodology are clustered into two-dimensional figures to achieve five clusters.



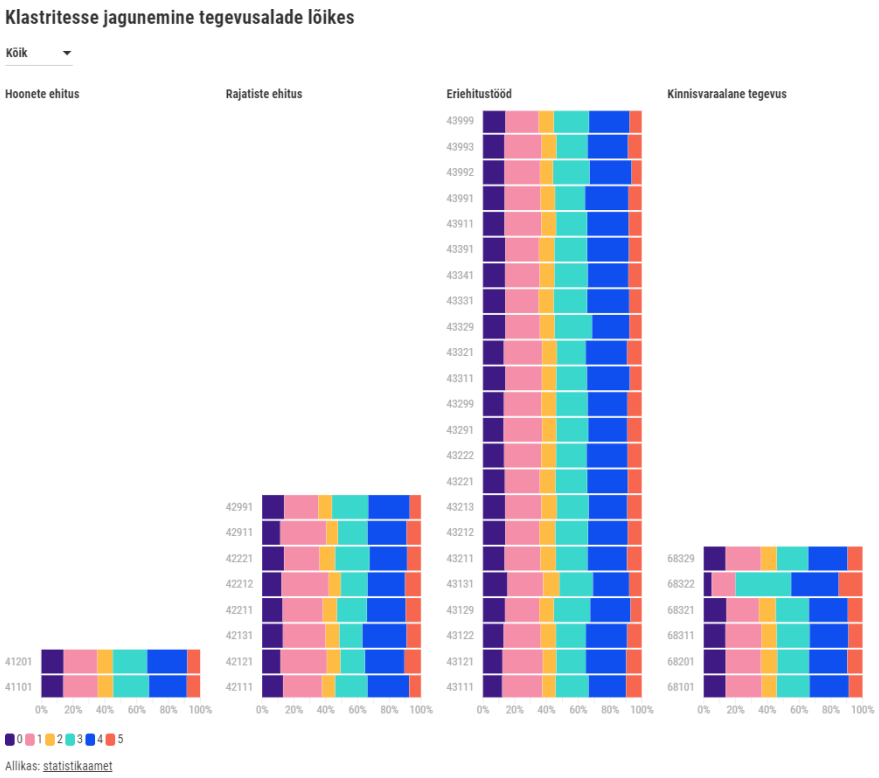
Annex 7: Correlation between the variables



	kood	staatus	PERIOD_NM	BI_100_1	BI_150_1	BI_180_1	BI_190_1	BI_240_1	BI_250_1	BI_290_1	BI_310_1	BI_370_1	BI_400_1	BI_40_1	BI_40_2	s1	s2	s3	s4	s5	s6
kood	1.000000	-0.024100	-0.033166	-0.006850	-0.004403	-0.008073	-0.011140	-0.007672	-0.004887	-0.006858	-0.005279	-0.007768	-0.004544	-0.005636	-0.006116	0.015823	-0.000112	0.002195	0.000838	-0.004491	0.025907
staatus	-0.024100	1.000000	-0.042746	0.002152	0.002349	0.002071	0.003241	0.005233	0.002791	0.004622	0.001984	0.004214	0.001559	0.001043	0.000702	-0.003108	-0.000112	0.000230	-0.000066	-0.000281	-0.003138
PERIOD_NM	-0.033166	-0.042746	1.000000	-0.003808	-0.004550	-0.005264	-0.005969	-0.005117	-0.003565	-0.004763	-0.004268	-0.005858	-0.000104	-0.001713	-0.002626	0.094657	-0.002420	-0.000792	-0.002383	-0.002954	-0.014811
BI_100_1	-0.006850	0.002152	-0.003808	1.000000	0.170758	0.542504	0.523649	0.159248	-0.757904	0.622311	0.208730	0.505656	0.038898	0.702865	0.599072	0.006890	-0.000057	-0.000061	-0.000074	-0.000176	-0.001057
BI_150_1	-0.004403	0.002349	-0.004550	0.170758	1.000000	0.599501	0.627658	0.263008	0.333489	0.372403	0.667988	0.634145	0.093135	0.251102	0.262894	-0.002711	-0.000037	-0.000023	-0.000034	-0.000116	0.000235
BI_180_1	-0.008073	0.002071	-0.005264	0.542504	0.599501	1.000000	0.799267	0.349011	0.381958	0.404244	0.699406	0.712677	0.166513	0.317255	0.323169	-0.006950	-0.000060	-0.000064	-0.000078	-0.000194	-0.000625
BI_190_1	-0.011140	0.003241	-0.005969	0.523649	0.627658	0.799267	1.000000	0.407315	0.637897	0.742528	0.718525	0.932495	0.169825	0.485813	0.497602	-0.004795	-0.000087	-0.000094	-0.000113	-0.000277	-0.001163
BI_240_1	-0.007672	0.005233	-0.005117	0.159248	0.263008	0.349011	0.407315	1.000000	0.098939	0.480657	0.255094	0.456262	0.045526	0.064382	0.107379	-0.002621	0.000187	0.000313	0.000181	-0.000157	-0.001057
BI_250_1	-0.004887	0.002791	-0.003565	0.757904	0.333489	0.381958	0.637897	0.098939	1.000000	0.881770	0.251571	0.709757	0.019175	0.669306	0.604944	-0.006284	0.000019	-0.000021	0.000020	-0.000111	-0.000706
BI_290_1	-0.006858	0.004622	-0.004763	0.622311	0.372403	0.404244	0.742528	0.480657	0.881770	1.000000	0.293266	0.804730	0.030974	0.577614	0.526052	-0.007076	0.000090	-0.000027	0.000076	-0.000161	-0.001106
BI_310_1	-0.005279	0.001984	-0.004268	0.208730	0.667988	0.699406	0.718525	0.255094	0.251571	0.293266	1.000000	0.754129	0.092596	0.250059	0.238212	-0.002924	-0.000033	0.000072	-0.000015	-0.000127	-0.000431
BI_370_1	-0.007768	0.004214	-0.005858	0.505656	0.634145	0.712677	0.932495	0.456262	0.709757	0.804730	0.754129	1.000000	0.076270	0.476384	0.504159	-0.006739	0.000032	0.000033	0.000046	-0.000188	-0.001029
BI_400_1	-0.004544	0.001559	-0.000104	0.038898	0.093135	0.166513	0.169825	0.045526	0.019175	0.030974	0.092596	0.076270	1.000000	0.026022	0.007200	-0.001988	-0.000025	-0.000030	-0.000033	0.000022	-0.000340
BI_40_1	-0.005636	0.001043	-0.001713	0.702865	0.251102	0.317255	0.485813	0.064382	0.669306	0.577614	0.250059	0.476384	0.026022	1.000000	0.853464	0.000649	-0.000081	-0.000088	-0.000105	-0.000255	0.001467
BI_40_2	-0.006116	0.000702	-0.002626	0.599072	0.262894	0.323169	0.497602	0.107379	0.604944	0.526052	0.238212	0.504159	0.007200	0.853464	1.000000	0.000083	-0.000079	-0.000073	-0.000083	-0.000276	0.001007
s1	0.015823	-0.003108	0.004657	0.006890	-0.002711	-0.006950	-0.004795	-0.002621	-0.006284	-0.007076	-0.002924	-0.006739	-0.001988	0.000649	0.000083	1.000000	0.000369	0.000391	0.000487	0.001437	0.002406
s2	-0.000112	-0.000112	-0.002420	-0.000057	-0.000037	-0.000060	-0.000087	0.000187	0.000019	0.000090	-0.000033	0.000032	-0.000025	-0.000081	-0.000079	0.000369	1.000000	0.023180	0.966477	0.099906	0.000405
s3	0.002195	0.000230	-0.000792	-0.000061	-0.000023	-0.000064	-0.000094	0.000313	-0.000021	-0.000027	0.000072	0.000033	-0.000030	-0.000088	-0.000073	0.000391	0.023180	1.000000	0.260139	0.032362	0.000284
s4	0.000838	-0.000066	-0.002383	-0.000074	-0.000034	-0.000078	-0.000113	0.000181	0.000020	0.000076	-0.000015	0.000046	-0.000033	-0.000105	-0.000083	0.000487	0.966477	0.260139	1.000000	0.103773	0.000479
s5	-0.004491	-0.000281	-0.002954	-0.000176	-0.000116	-0.000194	-0.000277	-0.000157	-0.000111	-0.000161	-0.000127	-0.000188	0.000022	-0.000255	-0.000276	0.001437	0.099906	0.032362	0.103773	1.000000	0.000407
s6	0.025907	-0.003138	-0.014811	-0.001057	0.000235	-0.000625	-0.001163	-0.001057	-0.000706	-0.001106	-0.000431	-0.001029	-0.000340	0.001467	0.001007	0.002406	0.000405	0.000284	0.000479	0.000407	1.000000

Annex 8: Division into clusters by areas of activity

Kommenteerinud [P1]: **Pildi tõlge vaja lisada.**



Annex 9: The TwoLayerNet code

```
from builtins import range
from builtins import object
import numpy as np

from layers import *
from layer_utils import *


class TwoLayerNet(object):

    def init (self, input_dim=3*32*32, hidden_dim=100, num_classes=10,
                 weight_scale=1e-3, reg=0.0):
        """
        Initialize a new network.

        Inputs:
        - input_dim: An integer giving the size of the input
        - hidden_dim: An integer giving the size of the hidden layer
        - num_classes: An integer giving the number of classes to classify
        - weight_scale: Scalar giving the standard deviation for random
          initialization of the weights.
        - reg: Scalar giving L2 regularization strength.
        """
        self.params ={}
        self.reg = reg
        self.cache = {}


        self.params['W1'] = weight_scale * np.random.randn(input_dim, hidden_dim)
        self.params['b1'] = np.zeros(hidden_dim)
        self.params['W2'] = weight_scale * np.random.randn(hidden_dim, num_classes)
        self.params['b2'] = np.zeros(num_classes)


    def loss(self, X, y=None):
        """
        Compute loss and gradient for a minibatch of data.

        Inputs:
        - X: Array of input data of shape (N, d_1, ..., d_k)
        - y: Array of labels, of shape (N,). y[i] gives the label for X[i].

        Returns:
        If y is None, then run a test-time forward pass of the model and return:
        - scores: Array of shape (N, C) giving classification scores, where
          scores[i, c] is the classification score for X[i] and class c.

        If y is not None, then run a training-time forward and backward pass and
        return a tuple of:
        - loss: Scalar value giving the loss
```

```
    - grads: Dictionary with the same keys as self.params, mapping parameter
      names to gradients of the loss with respect to those parameters.
    """
    scores = None

    W1, b1 = self.params['W1'],
    self.params['b1'] W2, b2 =
    self.params['W2'], self.params['b2']
    hidden, self.cache['hidden'] = affine_relu_forward(X, W1,
    b1) scores, self.cache['out'] = affine_forward(hidden, W2,
    b2)


    # If y is None then we are in test mode so just return scores
    if y is None:
       return scores

    loss, grads = 0, {}

    loss, delta3 = softmax_loss(scores, y)
    loss = loss + 0.5*self.reg*np.sum(W1**2) + 0.5*self.reg*np.sum(W2**2)
    delta2, grads['W2'], grads['b2'] = affine_backward(delta3, self.cache['out'])
    _, grads['W1'], grads['b1'] = affine_relu_backward(delta2,
    self.cache['hidden']) grads['W2'] += self.reg * W2
    grads['W1'] += self.reg * W1

    return loss, grads


class FullyConnectedNet(object):
    """
    A fully-connected neural network with an arbitrary number of hidden layers,
    ReLU nonlinearities, and a softmax loss function. This will also implement
    dropout and batch normalization as options. For a network with L layers, the
    architecture will be

    {affine – [batch norm] – relu – [dropout]} × (L – 1) – affine – softmax

    where batch normalization and dropout are optional, and the {...} block is
    repeated L – 1 times.

    Similar to the TwoLayerNet above, learnable parameters are stored in the
    self.params dictionary and will be learned using the Solver class.
    """

    def init (self, hidden_dims, input_dim=3*32*32, num_classes=10,
             dropout=0, use_batchnorm=False, reg=0.0,
             weight_scale=1e-2, dtype=np.float32,
       seed=None): self.use_batchnorm = use_batchnorm
       self.use_dropout = dropout > 0
       self.reg = reg
       self.num_layers = 1 + len(hidden_dims)
       self.dtype = dtype
       self.params = {}
```

```
    for i in range(self.num_layers):
        if i == 0 :
            self.params['W' + str(i+1)] = weight_scale * np.random.randn(input_dim, hidden_dims[i])
            self.params['b' + str(i+1)] = np.zeros(hidden_dims[i])
            if self.use_batchnorm:
                self.params['gamma' + str(i+1)] = np.ones(hidden_dims[i])
                self.params['beta' + str(i+1)] = np.zeros(hidden_dims[i])
        elif i < self.num_layers – 1:
            self.params['W' + str(i+1)] = weight_scale * np.random.randn(hidden_dims[i-1],
            hidden_dims[i]) self.params['b' + str(i+1)] = np.zeros(hidden_dims[i])
            if self.use_batchnorm:
                self.params['gamma' + str(i+1)] = np.ones(hidden_dims[i])
                self.params['beta' + str(i+1)] = np.zeros(hidden_dims[i])
        else:
            self.params['W' + str(i+1)] = weight_scale * np.random.randn(hidden_dims[i-1], num_classes)
            self.params['b' + str(i+1)] = np.zeros(num_classes)


    self.dropout_param = {}
    if self.use_dropout:
        self.dropout_param = {'mode': 'train', 'p': dropout}
        if seed is not None:
            self.dropout_param['seed'] = seed

    self.bn_params = []
    if self.use_batchnorm:
        self.bn_params = [{'mode': 'train'} for i in range(self.num_layers – 1)]

    # Cast all parameters to the correct datatype
    for k, v in self.params.items():
        self.params[k] = v.astype(dtype)


def loss(self, X, y=None):
    """
    Compute loss and gradient for the fully-connected net.

    Input / output: Same as TwoLayerNet above.
    """
    X = X.astype(self.dtype)
    mode = 'test' if y is None else 'train'

    if self.use_dropout:
        self.dropout_param['mode'] = mode
    if self.use_batchnorm:
        for bn_param in self.bn_params:
            bn_param['mode'] = mode

    scores = None

    a = {'layer0' :
    X} self.cache =
    {}
```

```
    for i in range(self.num_layers):
        W, b = self.params['W'+str(i+1)], self.params['b'+str(i+1)]
        l, l_prev = 'layer'+str(i+1), 'layer'+str(i)
        if mode == 'train':
            bn_params = {'mode':
        'train'} else:
            bn_params = {'mode':
        'test'} if i < self.num_layers –
        1:
            if self.use_batchnorm and self.use_dropout:
                gamma, beta = self.params['gamma'+str(i+1)], self.params['beta'+str(i+1)]
                a[l], self.cache[l] = affine_batchnorm_relu_dropout_forward(a[l_prev], W, b, gamma, beta,
bn_params, self.dropout_param)
            elif self.use_dropout:
                a[l], self.cache[l] = affine_relu_dropout_forward(a[l_prev], W, b,
            self.dropout_param) elif self.use_batchnorm :
                gamma, beta = self.params['gamma'+str(i+1)], self.params['beta'+str(i+1)]
                a[l], self.cache[l] = affine_batchnorm_relu_forward(a[l_prev], W, b, gamma, beta,
            bn_params) else:
                a[l], self.cache[l] = affine_relu_forward(a[l_prev], W,
        b) else:
            a[l], self.cache[l] = affine_forward(a[l_prev], W, b)

    scores = a['layer'+str(self.num_layers)]


    if mode == 'test':
        return scores loss,

    grads = 0.0, {}

    last = self.num_layers
    d = {}
    loss, dout = softmax_loss(scores, y)
    grads = {}
    w = 'W' + str(last)
    b = 'b' + str(last)
    c = 'layer' + str(last)
    dh, grads[w], grads[b] = affine_backward(dout,
    self.cache[c]) loss += 0.5 * self.reg *
    np.sum(self.params[w]**2)
    grads[w] += self.reg * self.params[w]
    for i in reversed(range(last -1)):
        w = 'W' +
        str(i+1) b = 'b' +
        str(i+1)
        gamma = 'gamma' + str(i+1)
        beta = 'beta' + str(i+1)
        c = 'layer' + str(i+1)
        if self.use_batchnorm and self.use_dropout:
            dh,          grads[w],          grads[b],          grads[gamma],          grads[beta]          =
affine_batchnorm_relu_dropout_backward(dh, self.cache[c])
        elif self.use_dropout:
            dh, grads[w], grads[b] = affine_relu_dropout_backward(dh,
        self.cache[c]) elif self.use_batchnorm:
```

```
            dh, grads[w], grads[b], grads[gamma], grads[beta] = affine_batchnorm_relu_backward(dh,
self.cache[c])
        else:
            dh, grads[w], grads[b] = affine_relu_backward(dh, self.cache[c])

        loss += 0.5 * self.reg * np.sum(self.params[w]**2)
        grads[w] += self.reg * self.params[w]



    return loss, grads
```

Annex 10: The Layers code

```
from builtins import range
import numpy as np

def affine_forward(x, w,
    b): """
    Computes the forward pass for an affine (fully-connected) layer.

    The input x has shape (N, d_1, ..., d_k) and contains a minibatch of N
    examples, where each example x[i] has shape (d_1, ..., d_k). For
    example, batch of 500 RGB CIFAR-10 images would have shape (500,
    32, 32, 3). We will reshape each input into a vector of dimension D = d_1 *
    ... * d_k,
    and then transform it to an output vector of dimension M.

    Inputs:
    - x: A numpy array containing input data, of shape (N, d_1, ..., d_k)
    - w: A numpy array of weights, of shape (D, M)
    - b: A numpy array of biases, of shape (M,)

    Returns a tuple of:
    - out: output, of shape (N, M)
    - cache: (x, w,
    b) """
    out = None

    out = x.reshape(x.shape[0], -1).dot(w) +

    b cache = (x, w, b)
    return out, cache


def affine_backward(dout,
    cache): """
    Computes the backward pass for an affine layer.

    Inputs:
    - dout: Upstream derivative, of shape (N, M)
    - cache: Tuple of:
      - x: Input data, of shape (N, d_1, ... d_k)
      - w: Weights, of shape (D, M)

    Returns a tuple of:
    - dx: Gradient with respect to x, of shape (N, d1, ..., d_k)
    - dw: Gradient with respect to w, of shape (D, M)
    - db: Gradient with respect to b, of shape (M,)
    """
    x, w, b = cache
    dx, dw, db = None, None, None

    dx = dout.dot(w.T).reshape(x.shape)
    dw = x.reshape(x.shape[0], -1).T.dot(dout)
```

```
    db = np.sum(dout, axis=0)

    assert dx.shape == x.shape, "dx.shape != x.shape: " + str(dx.shape) + " != " +str(x.shape)
    assert dw.shape == w.shape, "dw.shape != w.shape: " + str(dw.shape) + " != " +
    str(w.shape) assert db.shape == b.shape, "db.shape != b.shape: " + str(db.shape) + " != " +
    str(b.shape)

    return dx, dw, db


def relu_forward(x):
    """
    Computes the forward pass for a layer of rectified linear units (ReLUs).

    Input:
    - x: Inputs, of any shape

    Returns a tuple of:
    - out: Output, of the same shape as x
    - cache:x
    """
    out = None

    out = np.maximum(x, 0)

    cache = x
    return out, cache


def relu_backward(dout,
    cache): """
    Computes the backward pass for a layer of rectified linear units (ReLUs).

    Input:
    - dout: Upstream derivatives, of any shape
    - cache: Input x, of same shape as dout

    Returns:
    - dx: Gradient with respect to x
    """
    dx, x = None, cache

    mask = x > 0
    dx = dout * mask

    return dx


def dropout_forward(x, dropout_param):
    """
    Performs the forward pass for (inverted) dropout.

    Inputs:
    - x: Input data, of any shape
```

```
    - dropout_param: A dictionary with the following keys:
      - p: Dropout parameter. We drop each neuron output with probability p.
      - mode: 'test' or 'train'. If the mode is train, then perform dropout;
        if the mode is test, then just return the input.
      - seed: Seed for the random number generator. Passing seed makes this
        function deterministic, which is needed for gradient checking but not in
        real networks.

    Outputs:
    - out: Array of the same shape as x.
    - cache: tuple (dropout_param, mask). In training mode, mask is the dropout
      mask that was used to multiply the input; in test mode, mask is None.
    """
    p, mode = dropout_param['p'],
    dropout_param['mode'] if 'seed' in dropout_param:
        np.random.seed(dropout_param['seed'])

    mask =
    None out =
    None

    if mode == 'train':

    elif mode == 'test':


    cache = (dropout_param, mask)
    out = out.astype(x.dtype, copy=False)

    return out, cache


def dropout_backward(dout,
    cache): """
    Perform the backward pass for (inverted) dropout.

    Inputs:
    - dout: Upstream derivatives, of any shape
    - cache: (dropout_param, mask) from dropout_forward.
    """
    dropout_param, mask = cache
    p, mode = dropout_param['p'], dropout_param['mode']

    dx = None
    if mode == 'train':

    elif mode == 'test':
        dx = dout
    return dx


def softmax_loss(x, y):
    """
    Computes the loss and gradient for softmax classification.
```

Inputs:

- x: Input data, of shape (N, C) where x[i, j] is the score for the jth class for the ith input.
- y: Vector of labels, of shape (N,) where y[i] is the label for x[i] and 0 <= y[i] < C

Returns a tuple of:

- loss: Scalar giving the loss
- dx: Gradient of the loss with respect to x

"""

```
shifted_logits = x – np.max(x, axis=1, keepdims=True)
Z = np.sum(np.exp(shifted_logits), axis=1,
keepdims=True) log_probs = shifted_logits – np.log(Z)
probs = np.exp(log_probs)
N = x.shape[0]
loss = -np.sum(log_probs[np.arange(N), y]) / N
dx = probs.copy()
dx[np.arange(N), y] -=
1 dx /= N
return loss, dx
```

Annex 11: The Optim code

```
import numpy as np

"""
This file implements various first-order update rules that are commonlyused
for training neural networks. Each update rule accepts current weights and
the gradient of the loss with respect to those weights and produces the next
set of weights. Each update rule has the same interface:

def update(w, dw, config=None):

Inputs:
  - w: A numpy array giving the current weights.
  - dw: A numpy array of the same shape as w giving the gradient of the
    loss with respect to w.
  - config: A dictionary containing hyperparameter values such as learning
    rate, momentum, etc. If the update rule requires caching values over many
    iterations, then config will also hold these cached values.

Returns:
  - next_w: The next point after the update.
  - config: The config dictionary to be passed to the next iteration of the
    update rule.

NOTE: For most update rules, the default learning rate will probably not
perform well; however, the default values of the other hyperparameters should
work well for a variety of different problems.

For efficiency, update rules may perform in-place updates, mutating w and
setting next_w equal to w.
"""


def sgd(w, dw, config=None):
    """
    Performs vanilla stochastic gradient descent.

    config format:
    - learning_rate: Scalar learning
    rate. """
    if config is None: config = {}
    config.setdefault('learning_rate', 1e-2)

    w -= config['learning_rate'] * dw
    return w, config


def sgd_momentum(w, dw, config=None):
    """
    Performs stochastic gradient descent with momentum.
```

```
    config format:
    - learning_rate: Scalar learning rate.
    - momentum: Scalar between 0 and 1 giving the momentum value.
      Setting momentum = 0 reduces to sgd.
    - velocity: A numpy array of the same shape as w and dw used to store a
      moving average of the gradients.
    """
    if config is None: config = {}
    config.setdefault('learning_rate', 1e-2)
    config.setdefault('momentum', 0.9)
    v = config.get('velocity', np.zeros_like(w))

    next_w = None

    v = config['momentum']*v – config['learning_rate']*dw
    next_w = w + v


    config['velocity']= v

    return next_w, config



def rmsprop(x, dx, config=None):
    """
    Uses the RMSProp update rule, which uses a moving average of squared
    gradient values to set adaptive per-parameter learning rates.

    config format:
    - learning_rate: Scalar learning rate.
    - decay_rate: Scalar between 0 and 1 giving the decay rate for the squared
      gradient cache.
    - epsilon: Small scalar used for smoothing to avoid dividing by zero.
    - cache: Moving average of second moments of gradients.
    """
    if config is None: config = {}
    config.setdefault('learning_rate', 1e-2)
    config.setdefault('decay_rate', 0.99)
    config.setdefault('epsilon', 1e-8)
    config.setdefault('cache', np.zeros_like(x))

    next_x = None

    config['cache'] = config['decay_rate'] * config['cache'] + (1 – config['decay_rate']) * dx**2
    next_x = x – config['learning_rate'] * dx / (np.sqrt(config['cache']) + config['epsilon'])



    return next_x, config


def adam(x, dx, config=None):
```

"""
Uses the Adam update rule, which incorporates moving averages of both the
gradient and its square and a bias correction term.

config format:
- learning_rate: Scalar learning rate.
- beta1: Decay rate for moving average of first moment of gradient.
- beta2: Decay rate for moving average of second moment of gradient.
- epsilon: Small scalar used for smoothing to avoid dividing by zero.
- m: Moving average of gradient.
- v: Moving average of squared gradient.
- t: Iteration number.
"""
if config is None: config = {}
config.setdefault('learning_rate', 1e-3)
config.setdefault('beta1', 0.9)
config.setdefault('beta2', 0.999)
config.setdefault('epsilon', 1e-8)
config.setdefault('m', np.zeros_like(x))
config.setdefault('v', np.zeros_like(x))
config.setdefault('t', 1)

next_x =

None

config['t'] += 1
config['m'] = config['beta1']*config['m'] + (1-config['beta1'])*dx
config['v'] = config['beta2']*config['v'] + (1-config['beta2'])*(dx**2)
mt = config['m'] / (1-config['beta1']**config['t'])
vt = config['v'] / (1-config['beta2']**config['t'])
next_x = x – config['learning_rate'] * mt / (np.sqrt(vt) + config['epsilon'])

return next_x, config