UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Lisete Müürsepp

Positioning Estonian Companies with Power BI and Machine Learning

Bachelor's Thesis (9 ECTS)

Supervisor:
Fredrik Milani, PhD

Tartu 2025

Positioning Estonian Companies with Power BI and Machine Learning

Abstract:

The fundamental objective of benchmarking is the constant pursuit of excellence. In order to accomplish continuous improvement, internal operations and functions are evaluated externally, and then benchmarked. The purpose of this thesis is to compare Estonian companies to one another through financial ratios. To benchmark firms against peers by industry and revenue, a Power BI dashboard was developed. Then, clustering and classification techniques were applied in Python to examine whether company groupings that are based on financials can be learned. KMeans and hierarchical clustering revealed clusters of economic significance, whereas DBSCAN did not. However, classification models could not generalize the clustering structure, and frequently overfitted the majority clusters. The results indicate that although traditional financial ratios can confirm a company's intuitive market position, machine learning models are incapable of handling imbalanced and noisy data. The work offers a scalable framework for analysts, investors, and policymakers to explore financial competitiveness among Estonian firms. Future research should focus on refining a balanced data set, external validation of the clusters, and exploration of new algorithms or robust scaling methods.

Keywords: machine learning, company positioning, Power BI, clustering, classification

CERCS: P170 Computer science, numerical analysis systems, control

Eesti ettevõtete positsioneerimine Power BI ja masinõppega

Lühikokkuvõte:

Ettevõtteid positsioneeritakse selleks, et võrrelda end parematega ja püüelda kõrgema tulemuslikkuse, efektiivsuse ja likviidsuse poole. Selleks, et hinnata ettevõtete sisemisi protsesse ja funktsioone, tehakse finantsnäitajate ja suhtarvude võrdlusanalüüs.

Selle lõputöö eesmärk on võrrelda Eesti ettevõtteid üksteisega finantssuhtarvude kaudu. Ettevõtete võrdlemiseks konkurentidega valdkonna ja tulude järgi töötatakse välja Power BI töölaud. Seejärel rakendatakse Pythonis klastrite moodustamise ja klassifitseerimise tehnikaid, et uurida, kas finantsnäitajatel põhinevat ettevõtete rühmitamist on võimalik masinõppida. KMeans ja hierarhiline klastrite moodustamine näitasid majanduslikult olulisi klastreid, DBSCAN mitte. Klassifikatsioonimudelid ei suutnud aga klastrite struktuuri üldistada ja sobitusid liiga täpselt olemasolevate klastritega, õppides pigem selgeks andmete juhuslikud eripärad, mitte üldised mustrid.

Tulemused näitavad, et kuigi traditsioonilised finantssuhtarvud suudavad kinnitada ettevõtte intuitiivset positsiooni, ei ole masinõppe mudelid võimelised käsitlema tasakaalustamata ja müraseid andmeid.

Töö pakub analüütikutele, investoritele ja poliitikakujundajatele skaleeritavat raamistikku Eesti ettevõtete finantskonkurentsivõime uurimiseks. Tulevikus tuleks keskenduda tasakaalustatud andmestiku täiustamisele, klastritevälisele valideerimisele ning uute algoritmide või robustsete skaleerimismeetodite uurimisele.

Võtmesõnad: masinõpe, ettevõtete positsioneerimine, Power BI, klaserdamine, klassifitseerimine

CERCS: P170 Arvutiteadus, arvutumeetodid, süsteemid, juhtimine (juhtimisteooria)

Table of Contents

1. Introduction

In the realm of business analytics and management, financial ratios are prevalent due to the vast amount of information in a set of financial statements and the need to compare businesses of various sizes [1]. They give stakeholders quantitative information that helps to evaluate both the present and future position of the company [2]. Because of their capacity to absorb information multi-dimensionally, ratios are the main analytical tool used to examine an organization's financial accounts [1]. The incorporation of these measures into more comprehensive business intelligence (BI) and data analytics frameworks has been made possible by the growing availability of structured financial data. Moreover, machine learning methods have grown in relevance in this situation, enabling models to predict business behaviour and identify patterns without exclusively depending on pre-established frameworks [3]. These methods are excellent for financial data since they are flexible and accurate in their predictions.

Since 2022, the Estonian e-Business Register has published comparable data on all legal businesses registered in Estonia. Despite the richness of the datasets, they have not been extensively studied in benchmarking or machine learning contexts, which creates an opportunity to explore the extent to which this data can be analysed. More specifically, this paper seeks to understand how to benchmark one company against its industry using different financial metrics, like working capital turnover, debt and cash ratio, and net profit margin. Given this, the aim of the thesis was to develop a BI model for assessing company competitiveness, and machine learning methods to help analyse which technique is the best to answer to why the company is in one position or another.

The thesis is divided into two parts. The first part compares a company against its industry peers and against companies that are in the same annual revenue range by using 15 financial indicators. Through this comparison, the company can assess where it is positioned compared to others in terms of liquidity, efficiency and profitability. In the second part, different models are tested to help answer why the company is at a certain position. To determine that, two clustering techniques are implemented and later tested with five different classification algorithms. The purpose of this is to determine if a company's financial profile can be used to reliably predict its cluster membership.

To achieve the goal, data from the Estonian e-business Register is processed and modelled in Microsoft Power BI to build a dashboard that positions companies. Then, the prepared data is exported into Python, where clustering methods, including KMeans, hierarchical and DBSCAN, along with supervised classification, including Naive Bayes, SVM, Gradient Boosting, and Random Forest, are applied. The reusable BI benchmarking tool for Estonian companies and an evaluation of machine learning methods for financial positioning can contribute to company executives, investors, analysts, and policy makers, who may benefit from this framework to observe trends and assess financial health at the sectoral level.

The rest of the thesis is structured as follows. Chapter 2 provides background on BI in financial analysis, the role of financial ratios, and machine learning in company positioning. Chapter 3 outlines the methodology used in Power BI and Python. Chapter 4 presents the results of clustering and classification. Chapter 5 discusses the findings, their implications and limitations. Chapter 6 concludes the thesis and proposes ideas for future research. To improve readability, grammar, and consistency of the text, ChatGPT GPT-3.5 and GPT-4o models were occasionally used.

2. Background

This chapter provides background information about business intelligence in the context of financial analysis, discusses financial ratios and their importance in analytics, and introduces the integration of financial positioning and machine learning.

2.1 Business Intelligence in Financial Analysis

The literature on business intelligence (BI) suggests that using business intelligence systems can have a multitude of advantages, including increased customer satisfaction, quicker and simpler access to information, lower information technology costs, and increased enterprise competitiveness [4]. Therefore, there is a strong need for a system that combines knowledge management with decision support procedures. Given that decision support and knowledge management entail complementary actions, like retrieval, storing and sharing knowledge, and model development and preservation, they can be integrated to generate a synergy [5]. The knowledge management functionality's retrieval, storing and sharing of knowledge improves the dynamic development and preservation of decision support models, which in turn enhances the decision support process [6].

Microsoft Power BI is an ecosystem of BI tools, which is all about utilising data to make better decisions across industries. It is designed to transform raw data into informative insights and supports the entire data pipeline which can be divided into five areas: domain, data, model, analysis, visualisation. Therefore, the tool integrates data processing, modelling and analysis. A remarkable service that is included is Power Query, which transforms data and keeps it linked to its original source. Also, in Power BI, one can define relationships in data models, perform calculations with the usage of Data Analysis Expressions (DAX) language, and share dashboards and reports with stakeholders [7].

2.2 Financial Ratios and their Analytical Importance

Some of the best indicators of an organization's potential for long-term growth and successful operation in a competitive market setting are its financial ratios, which are commonly used by professionals for financial assessment. Ratios are the primary analytical instrument for examining an organization's financial accounts because of their multi-dimensional information absorption capabilities [1]. These indicators serve several purposes in strategic financial analysis. First, they

are instrumental in evaluating company efficiency, profitability, liquidity, solvency and capital structure. The categories of ratios commonly used include structure of capital, efficiency, liquidity, and profitability. This categorisation is supported by well-established reasoning: ratios like the equity-to-assets ratio or debt level reveal information about a business's financing structure, while ratios like ROA or net profit margin show how well a company may generate income off of its assets [8].

Financial ratios are applicable in a wide range of sectors, including public administration, IT, hospitality, transportation, and education. Their role in benchmarking is especially notable. Ratios allow organizations to measure themselves against sectoral averages or strategic targets, thus aiding in relevant decision making [1]. Financial ratios also play a critical role in company positioning by serving as standardised indicators. This standardisation is essential when comparing entities across sectors or countries with differing accounting standards. Future predictions may be realised using different statistical models and, therefore, vary by country or region due to economic and institutional factors. This is why certain ratios remain dominant and help provide context in various situations [9].

In practical applications, ratios have been essential in defining financial norms across industries. Because of vastly different capital structures and turnover speeds, different business sectors like logistics, agriculture, or financial service require tailored threshold values [8]. This sector-sensitive benchmarking is made possible by z-scores, which are expressed as standardised deviations from their means and make cross-company comparisons in clustering and classification models possible [10]. Cialone's [11] study supports this view by showing that profitability, liquidity, and capital structure ratios are especially relevant for distinguishing between viable and underperforming companies. His categorisation of indicators into structure of capital, liquidity, efficiency, and profitability reinforces their multifaceted utility in enterprise assessment.

Financial ratios are essential analysis instruments in firm performance measurement, peer comparison, and supporting data-driven managerial decision-making. Their use spans high-level profitability evaluation to detailed operational diagnostics, and as such, they are irreplaceable in modern BI frameworks. The choice of ratios for this work depended on the data that was available and is further explained in the methodology and Appendix I.

2.3 Machine Learning for Company Positioning

Traditional statistical methods typically require researchers to impose structures on various models, such as linearity in multiple regression analysis, and to construct the model by estimating parameters to fit the data or observation. This is the primary distinction between traditional statistical methods and machine learning methods. By automatically extracting knowledge from a data collection and creating several model representations to explain the data set, machine learning techniques enable learning the specific structure of the model from the data [12]. This flexibility can allow for higher accuracy rates which, in the end, is the critical deciding factor to using a model. Broadly, there are two types of machine learning approaches: supervised and unsupervised. Wherever the output information is recognised by the system, supervised learning utilises the tagged information. It uses algorithms such as Naive Bayesian (NB), Support Vector Machines (SVMs), Decision Trees, Linear and Logistic Regression among others to solve classification and regression problems. The output of unsupervised learning, which uses untagged data, depends on the turnout of perceptions. Using algorithms such as Clustering, Anomaly Detection, Neural Networks, Approaches for Learning Latent Variable Models, etc., it handles cluster and associative rule mining problems [3].

Supervised classification models have been widely used for predictive tasks in financial and industrial contexts. Syed [13] shows how predictive analytics using categorisation models can assist predictive maintenance occurrences and optimise resource allocation in the automotive industry. Data mining methods have been applied in marketing to predict consumer behaviour and enhance campaign results [14]. Uludag and Gürsoy [15] demonstrate good prediction performance in the manufacturing sector by using supervised learning algorithms with the Altman Z-score framework to classify financial distress. These uses highlight the versatility of classification results across different fields.

Supervised learning includes two major branches: regression, where outputs are continuous, and classification, where outputs are discrete. Classifiers play a particularly prominent role in supervised learning, where they map observations to predefined classes. The goal is often to generalise from historical, labelled data so that based on that, the model can accurately classify new data points. Such classifiers may include probabilistic summaries, SVMs, decision trees, and algebraic functions [16].

Once a meaningful clustering structure is established, supervised classification methods can be used to “replicate” these groupings in new data [17, 16]. Clusters of related entities can be used as a useful reference framework in applied benchmarking scenarios, which enables the models to generalise and allocate new observations to relevant peer groups according to learned patterns [17]. Although clustering starts without labels, supervised classification makes it possible to validate and spread these clusters across datasets, allowing for continuity in applied or longitudinal research [18].

In the context of company positioning, supervised learning can be used to classify firms by financial health or risk and unsupervised learning can be used to segment firms based on similar financial or business profiles. These approaches help identify patterns in data that support strategic comparisons across sectors or company sizes.

3. Data Processing and BI Methodology

The following section describes the methodology for processing and modelling financial data using Microsoft Power BI. The data was retrieved from Estonia e-Business Register, which contains the details of all legal businesses registered in Estonia in a single environment. The goal was to create an interactive dashboard that positions a company against its industry peers or companies of similar size using financial ratios. This process utilised the CRISP-DM (Cross Industry Process for Data Mining) framework to have a logical and structured flow.

3.1 Data Understanding

The source data was retrieved from Estonian e-Business Register's open data portal, where the datasets contain comprehensive financial information about Estonian companies. Understanding the scope and structure of this data was a critical step, as it influenced every aspect of the entire process.

Estonian e-Business Register is the official state platform that compiles data on all legal businesses registered in Estonia. Among other information, the register includes company-level data, annual financial reports, ownership structures and sectoral classifications. It also offers open datasets for public use, making it a reliable and transparent foundation for large-scale financial analysis.

In order to adequately position companies and compare their financials, it is essential to identify the firm's area of activity. In Estonia, this is done by using the Classification of Economic Activities in Estonia (EMTAK), which serves as the basis for determining the field of activity of enterprises and organizations and is to some extent the national adaption of the European NACE standard [19]. Data retrieved from the e-Business Register includes both quantitative financial indicators and sectoral identifiers, the EMTAK code being the primary one, which allows for structured analysis of Estonian companies.

The activity classification is hierarchical in its structure, divided into five levels. The first four levels have been taken over from the European Community Statistical Classification of Economic Activities (NACE), while the fifth level, which has taken into account the specific characteristics of the Estonian economy and the relevant legislation, is national [19].

Determining the field is crucial, because the financial position of a company is described differently across sectors from the perspective of ratios. For instance, a port, a dental clinic and a hamburger kiosk differ significantly by their need for current assets, long-term loans, seasonality, and balance sheet composition. Nevertheless, EMTAK does not offer a straightforward sectoral breakdown. When businesses are categorised using just the first digit (for example, “8” for the medical sector) or even the first two digits (for example, “86” for medical care), important subcategories can be extracted and irrelevant subcategories can be included. For instance “87101” (residential nursing care), a crucial component of the healthcare industry, would be left out if “86” was the only code-beginning used for medical treatment. However, if medical care was extended to codes that began with just “8”, “88911” (child-care activities), which is not a medical activity, can be inadvertently included [8]. In general, though, by using the EMTAK code, the counterparts of the same fields can be matched and through this, the validity and the fit of comparison can be increased [20].

Five different datasets were used: General Company Metadata, Financial Report Metadata, Sales Revenue by EMTAK, Sales Revenue by Region, and Financial Report Items 2019-2023. While the first four contained metadata that helped categorise the companies, the Financial Report Items all together contained 24 different financial indicators (excluding values that had “Consolidated” at the end of their variable name). These indicators can be found in Appendix II and were later used to calculate the financial ratios.

In total, the datasets contained thousands of fields across multiple years with considerable overlap and redundancy in variable names. Understanding this was crucial for modelling, since the same concept appeared under multiple names, and handling this properly significantly affected performance and usability later. Since there is only five years worth of data available, the building process of model structure had to consider scalability.

3.2 Data Preparation

Power BI’s architecture enables building reusable datasets, where each business process is represented by a fact table, usually in a star schema. Fact tables contain values for the same numeric columns and the same key columns to dimensions. Each fact table row comprises quantitative data and represents a distinct business process at a certain level of detail or granularity [20]. In the source database, each year’s financial data came as a separate table, one for each year

between 2019 and 2023. These were first appended together into a single unified fact table using Power BI's Append Queries functionality. Since the schema remained the same across years, the append operation was straightforward. This unified table became the base fact table (AruanneteElemendid (elemendid koos)) that holds actual numeric values, such as how much revenue a company reported in a specific year.

One of the major challenges was dealing with inconsistent field names. For example, the revenue field could appear as "Revenue", "Consolidated Revenue", "Käive", "Konsolideeritud käive" and so on. If each of these remained as separate fields or names in the fact table, the dataset would become bloated, and analysis would be confusing. To solve this, the project implemented a data normalisation technique. First, all unique field names (e.g., "revenue", "assets", "liabilities", etc.) were extracted from the original dataset. These names were collected in two separate metadata tables: one for Estonian names (AruanneteElemendid elementEST) and one for English names (AruanneteElemendid elementENG). Each unique name was assigned a unique numeric index (e.g. 1 for "revenue", 2 for "assets", etc.).

In order to give the numerical data in the fact table context, dimension tables typically include descriptive information such as product categories, dates, regions, which in turn can be used to filter, organise and categorise the data [20]. Therefore, in the fact table, instead of storing the long text strings for each data element, the index numbers were used. This mapping was stored in the dimension tables, and the fact table only contained the index and the corresponding numeric value (e.g., revenue = 1000€ becomes field 1 = 1000€). This drastically reduced data volume and improved performance. This process is known as data normalisation and it allows the use of dimension tables to interpret what each index number means, instead of repeating full text fields throughout the dataset.

Once the index fields were established, a Merge Queries step was performed to enrich the fact table with human-readable labels by joining it with the dimension tables. This was done three times to map table name index, element name in Estonian, and element name in English. These merges used Left Join operations. After merging, the redundant text fields were removed, and only the index values remained in the fact table. Relationships were then created in the Power BI data model, linking the fact table to the tree dimension tables. A visualisation of this can be found in

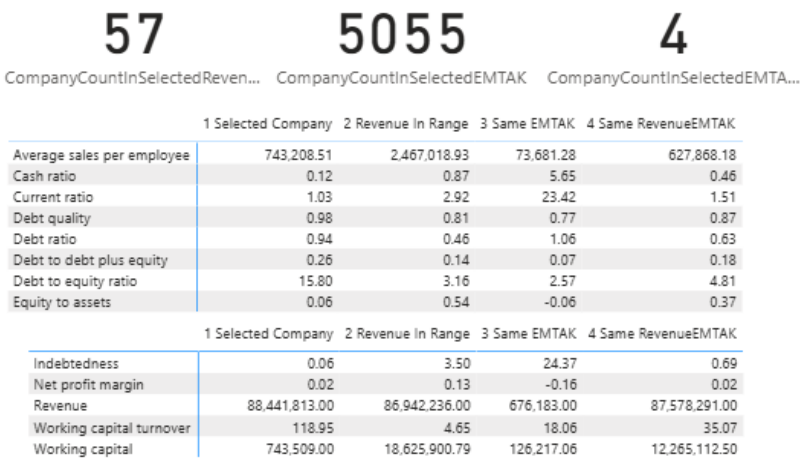
Appendix III. This ensured that measures and visualisations could dynamically pull the right labels without storing large text fields in every row.

3.3 Modelling

A crucial strength of Power BI is the semantic data modelling, which allows the creation of calculations like calculated columns, measures and tables using Data Analysis Expressions (DAX). DAX is a powerful formula language similar to Excel but optimised for tabular data models. DAX measures support dynamic calculations, adjusting based on user-defined filters or selections [7]. Taking this into consideration, once the data model was cleaned and normalised, the next step was building the logic for interactivity and comparison across companies. This was mainly done using DAX.

Power BI excels at creating interactive dashboards, which provide easily understandable visualisations of KPIs, which for financial analysis, are metrics that capture the health and performance of an organisation [7]. Common KPIs can include profit margins, equity ratio, liquidity and return on assets. Power BI allows these KPIs to be dynamically filtered by year, sector, or EMTAK code, which helps to enhance comparability across companies.

A key part of the dashboard was comparing the selected company's financial metrics and ratios against various peer groups. Therefore, the final comparison table contains four columns: the selected company, all companies with similar revenue as the selected company (the tolerance can be chosen by the user, can be up to 20%), all companies with the same EMTAK code as the selected company, and all companies in both the same EMTAK and revenue range as the selected company. In the rows, all calculated financial ratios are shown, first the selected company's, then the average of the indicators of all companies that are in the selected range of the revenue, the average of all companies in the same EMTAK, and then the average of all metrics that are in the same EMTAK and same revenue range. This is illustrated in Figure 1.



	1 Selected Company	2 Revenue In Range	3 Same EMTAK	4 Same RevenueEMTAK
Average sales per employee	743,208.51	2,467,018.93	73,681.28	627,868.18
Cash ratio	0.12	0.87	5.65	0.46
Current ratio	1.03	2.92	23.42	1.51
Debt quality	0.98	0.81	0.77	0.87
Debt ratio	0.94	0.46	1.06	0.63
Debt to debt plus equity	0.26	0.14	0.07	0.18
Debt to equity ratio	15.80	3.16	2.57	4.81
Equity to assets	0.06	0.54	-0.06	0.37

	1 Selected Company	2 Revenue In Range	3 Same EMTAK	4 Same RevenueEMTAK
Indebtedness	0.06	3.50	24.37	0.69
Net profit margin	0.02	0.13	-0.16	0.02
Revenue	88,441,813.00	86,942,236.00	676,183.00	87,578,291.00
Working capital turnover	118.95	4.65	18.06	35.07
Working capital	743,509.00	18,625,900.79	126,217.06	12,265,112.50

Figure 1. Comparison of Maru Ehitus AS against its peers.

To make this work, special flags were calculated using DAX, such as [IsInSelectedEMTAK], [RevenueInRange] and [IsInSelectedYear]. These flags helped filter the data dynamically depending on user selections. An example of such calculations is provided in Figure 2. It must be noted that although companies usually have more than one EMTAK code since, for example, a mobile company can also be classified as a real estate company if they own their office space, only their main area of activity is taken into account here. This was done by filtering the Sales Revenue by EMTAK table and keeping only the values where the “main field” column is equal to “yes”, as there is only one of those for each company.

```
CompanyCountInSelectedEMTAKRevenue =
CALCULATE(
    DISTINCTCOUNT(AruanneteYldandmed[registrikood]),
    FILTER(
        ALL(AruanneteYldandmed),
        [RevenueInRange] = 1 &&
        [IsInSelectedEMTAK] = 1 &&
        [IsInSelectedYear] = 1
    )
)
```

Figure 2. Calculating the number of companies in both the same revenue range and EMTAK.

Calculation of these measures worked by removing the current report files using ALL(), then applying custom logic to simulate the desired comparison context.

What is displayed in columns 2-4 (Figure 1) are the averages of each financial ratio of all companies that fit into that category. To further explain this, when a company is chosen, first, the number of companies in each comparable category is calculated. Then, in the model, the sum of each indicator is calculated for the whole model, but when put into context, the sum picks out only the fitting companies. For instance, Maru Ehitus AS in Figure 1 is one of 5,055 companies that all operate in the same field, so for all metrics, the sum only considers these 5,055 companies. After that, it divides the sum with the company count, e.g., 5,055, and returns the average. This calculation is illustrated in Figure 3.

```
3 Same EMTAK =
CALCULATE(
    SELECTEDMEASURE(),
    KEEPFILTERS(
        FILTER(
            ALL(AruanneteYldandmed),
            [IsInSelectedEMTAK] = 1 &&
            [IsInSelectedYear] = 1
        )
    )
) / [CompanyCountInSelectedEMTAK]
```

Figure 3. Calculation of the average values in the same EMTAK group.

3.3.1 Additional Techniques

To prevent unwanted filter behaviour, parallel dummy tables were created. These tables helped simulate user selections, which include company name, report year and revenue tolerance without applying unwanted filters to the entire model. They were disconnected from the main model and used only to define filter context explicitly. This made it possible to compute KPIs and summaries without accidental interference from other filters.

Additionally, Power BI often misorders categorical fields like month names. To fix this, numerical sort keys were introduced. For example, months were sorted based on their numeric index (1 = January, 2 = February, etc.) rather than alphabetically. Dummy tables were used again here to isolate sorting logic and avoid disrupting filter contexts.

4. Machine Learning

After the data was prepared, normalised and interactively modelled, the process was followed by the machine learning phase. The preparation began with exporting the cleaned and structured financial data from Power BI in CSV format. The exported tables represented company-level financial summaries, ready for unsupervised and supervised learning tasks. This part of the project was aimed at positioning companies using clustering algorithms, and then building classifiers that could predict a company's cluster membership based on its financial profile. The transition from Power BI to Python for machine learning was a natural continuation of the CRISP-DM methodology, moving data preparation and modelling into deeper data mining and evaluation stages

4.1 Data Exporting and Loading

The datasets from Power BI included preprocessed financial values for Estonian companies from five consecutive years: 2019 through 2023. These were saved as CSV files (cluster2019.csv, cluster2020.csv, etc.) with a consistent structure. Each table contained derived financial metrics discussed earlier in this paper. All non-numeric fields, which were the company names and the EMTAK codes were removed at this stage to facilitate numeric analysis. For machine learning, only companies with a revenue of 1,000,000€ or higher were considered, since smaller companies may not have their financial statements audited and can submit misinformation, leading to inconsistencies in the model. Therefore, the clustering was performed on approximately 7,000-10,000 companies, depending on the year, with 15 numerical financial variables per year.

Once exported, the CSV files were loaded into Python using the pandas library. During loading, some additional data cleaning steps were performed, such as removing infinite values (resulting from divisions like equity/assets) and fully empty columns. After this, missing values were imputed using the median strategy with SimpleImputer, which was chosen for its robustness to outliers.

To ensure fair contribution to distance-based algorithms, all numerical values were standardised using StandardScaler. This transformed the dataset into z-scores, with each feature having a mean of 0 and a standard deviation of 1. This was necessary because it ensures that all features contribute equally to the clustering. Without this, features that have very large values (like "Revenue" or

"Assets" in this context) would have dominated the distance calculations, while small-scale features like "Profit margin" or "Debt ratio" would have been ignored, though they are as important as the previous ones. With StandardScaler, all features are on the same scale.

4.2 Unsupervised Learning: Company Clustering

Clustering is an unsupervised learning method used to group data sets that have not been classified based on similarity, offering businesses a way to uncover hidden structures in customer or company datasets without predefined labels [21, 22]. In strategic business contexts, clustering enables companies to identify naturally emerging customer or company segments to guide targeted actions such as positioning, marketing, or product development [23, 24].

Clustering was applied as the first analytical step. The main objective was to group companies into distinct financial profiles (or segments) based on their numerical characteristics. This would allow identification of strategic clusters, such as high-performing companies, distressed companies, or low-risk consistent earners. Three algorithms were explored for this: KMeans, hierarchical and DBSCAN.

4.2.1 KMeans Clustering

K-Means is a centroid-based algorithm that assigns each data point to the cluster with the nearest mean, updating centroids iteratively to minimise the sum of the distances between the points [21]. This method is straightforward, scalable, and widely used in market segmentation due to its ability to handle large datasets and reveal interpretable clusters [22, 25]. However, a key challenge is determining the optimal number of clusters, for which the Elbow Method is commonly used. This method plots the sum of squares for different K values and selects the point where after that, adding more clusters would provide diminishing returns [21, 26]. In addition, the Silhouette Score is used to assess cluster validity by comparing its tightness and separation; higher Silhouette values suggest better-defined clusters [27, 22]. In contrast to other clusters, the Silhouette value indicates how similar an object is to its own cluster. A high Silhouette value means that an object is well matched to its own cluster and poorly matched to neighbouring clusters. The Silhouette range is -1 to +1. As data becomes more dimensional, it becomes more challenging to attain high values due to the curse of dimensionality, which makes distances more similar [28].

The KMeans algorithm was run with different values of k (number of clusters) ranging from 2 to 10. To determine the optimal number of clusters, both inertia, using the Elbow Method, and Silhouette Scores were plotted for cluster counts. The Elbow Method indicated the optimal number of clusters, because, as shown in Figure 4, it faced a significant drop when working with a higher number of clusters.

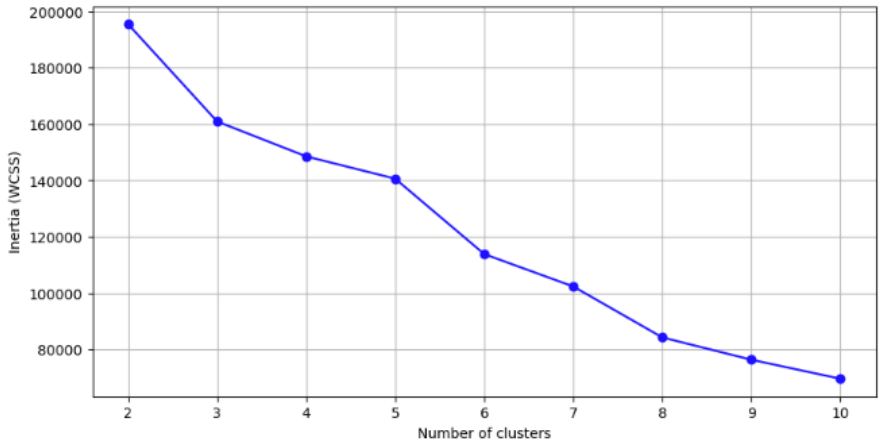


Figure 4. Elbow Method for optimal K.

The plot in Figure 4 suggests that k=5 might be the most reasonable choice to capture more variance without overfitting. The drop in inertia starts to slightly flatten compared to the previous ones and is followed by a bigger drop after that.

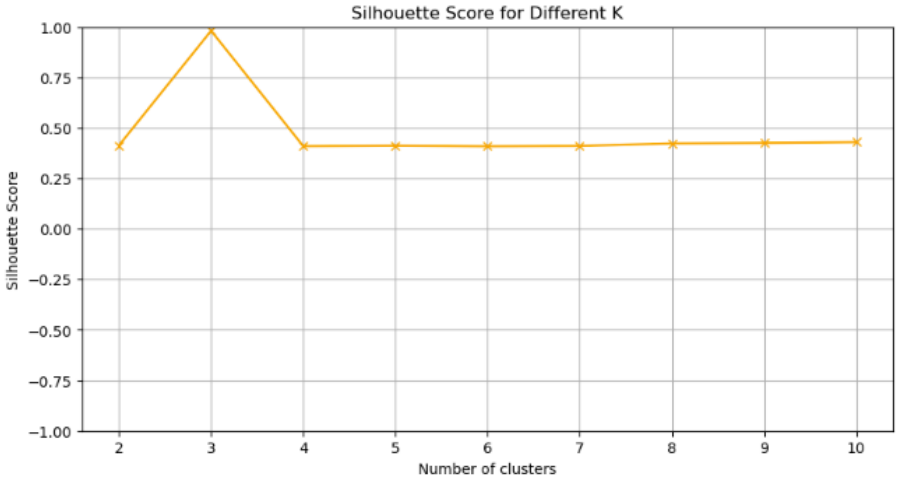


Figure 5. Silhouette Score for different K.

The highest Silhouette Score, as seen in Figure 5, is at k=3, reaching 1.00, which is the maximum possible score. For all other values, it stays relatively the same, at around 0.45, which indicates reasonably good clustering, with a few clusters overlapping, but having a reasonably sound structure. However, a score of 1.00 is unusually high and might indicate a possible artificial effect caused by isolated or very small clusters. Therefore, taking the results of both evaluation metrics to account, the most reasonable number of clusters is 5.

After assigning k=5 for the KMeans method as the initial cluster centers, each data point was assigned to the nearest cluster center using Euclidean distance. Then the centroid was recalculated as the mean of all points assigned to it. After that, KMeans clustering was applied to the scaled and imputed financial dataset. Once clustering was performed, the results were visualised using Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that transforms correlated variables into orthogonal principal components, enabling more efficient cluster visualisation as points in maps [29]. PCA compliments clustering by reducing noise and giving visual interpretation of cluster structures [30]. In Figure 6, it is clear that there are five reasonably well-separated groups, confirming that the KMeans algorithm was able to partition the dataset into meaningful clusters.

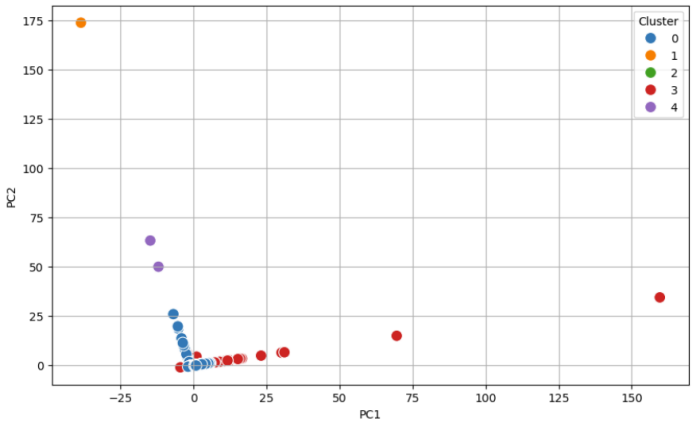


Figure 6. Clusters visualised with PCA.

4.2.2 Hierarchical Clustering

Hierarchical clustering is a clustering technique that builds a binary merge tree of clusters (a dendrogram) by progressively merging the closest clusters starting from the data elements (leaves) to the whole dataset (a root), providing a visual hierarchy of data structure [31]. The Ward linkage

method, in particular, minimises the total within-group dispersion at each step, producing compact and well-separated clusters ideal for business segmentation tasks. Hierarchical clustering does not require specifying the number of clusters in advance, and dendrograms allow for visual inspection to select an appropriate cut-off point. The Ward method's mathematical foundation is closely related to that of K-Means, both aiming to minimise within-cluster dispersion, which explains their complementary application in clustering-based positioning models [30].

Hierarchical clustering was also applied to the same data using the Ward linkage method. This starts with each point as its own cluster, computes distances between all pairs of clusters, merges clusters that minimise the Ward distance and is repeated until all data points are in one big cluster. The results are stored in a linkage matrix and were visualised with a dendrogram (see Figure 7).

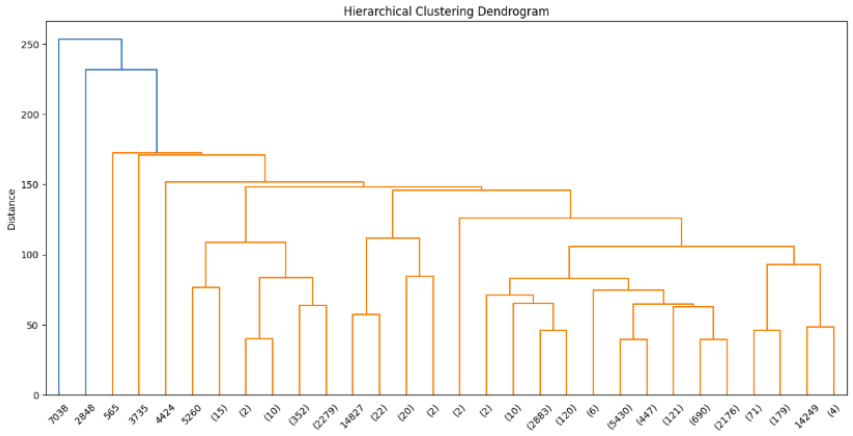


Figure 7. Hierarchical Clustering Dendrogram.

Although it seems in Figure 7 that the dendrogram should be cut somewhere around the distance of 150, this distance failed to provide meaningful and balanced clusters. Therefore, a distance of t=20 was chosen. This had much better balance, which is critical for training effective classification models and the top 10-15 are usable for supervised learning. Once the clustering was performed, the results were shown using PCA (Figure 8).

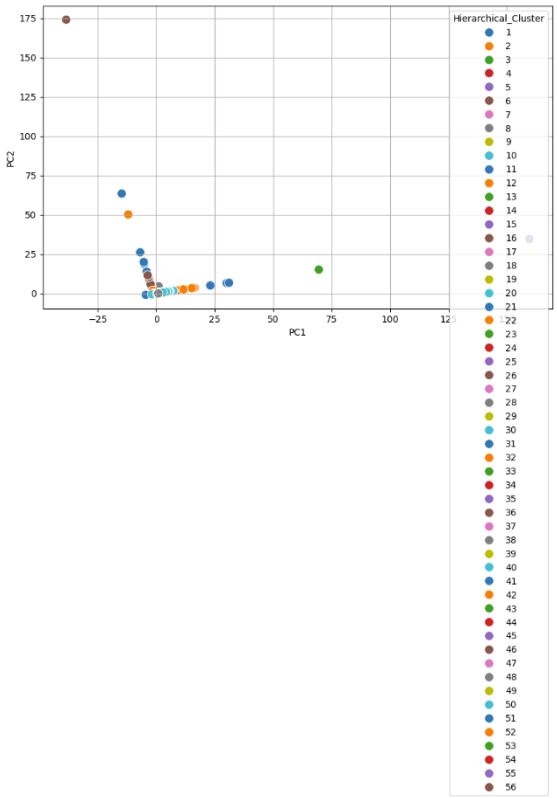


Figure 8. Hierarchical clusters visualised with PCA.

The most notable group sizes were Cluster 32 with 3,602 companies, Cluster 28 with 1,781 companies, Cluster 33 with 1,593 companies, and Cluster 42: 1,069 companies. Other clusters were smaller, with most consisting of only a few dozen companies and some even around 2 points.

4.2.3 DBSCAN

Unlike centroid- or connectivity-based algorithms, DBSCAN identifies clusters as areas of high data density and marks sparse regions as noise or outliers. This makes DBSCAN particularly effective in identifying niche companies or firms with outlier characteristics in an industry context, which might be overlooked by traditional clustering methods. DBSCAN does not require pre-specifying the number of clusters, but it can struggle with varying densities [32]. It requires two parameters: ε, which can be chosen by using a K-distance graph, and the minimum number of points required to form a dense region [33].

DBSCAN was also tested, although it could be expected that it would be less successful due to the fact that the data lacks well-separated dense regions. First, the method NearestNeighbors from

scikit-learn was used to find the five closest points for each item. Then, they were fit to the dataset using the previous scaled features. After that, a K-distance graph was plotted to find the most optimal distance. In addition to that, 95th and 98th percentiles of the distance distributions were calculated, so the results of the graph and percentiles could be compared and the most optimal epsilon value could be chosen.

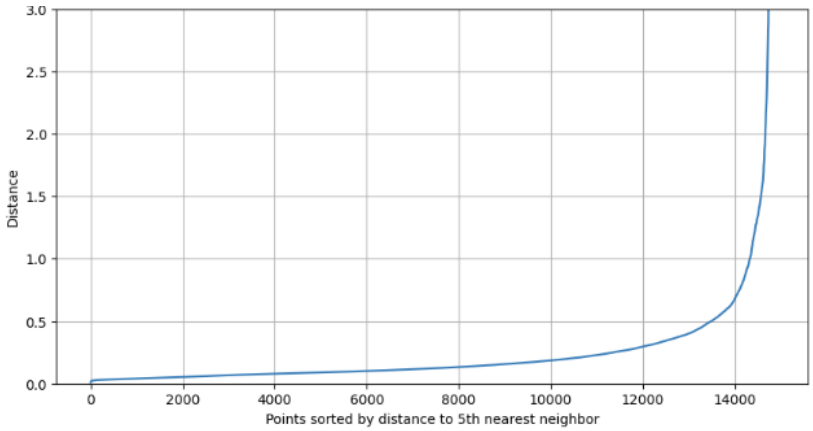


Figure 9. K-distance graph for DBSCAN.

The graph (Figure 9) demonstrated that the epsilon value should be between 0.5 and 1.5. 95th percentile demonstrated that with an epsilon equal to approximately 0.76, five meaningful clusters will form.

After assigning ε=0.76 for the DBSCAN method as the radius around a data point, it looked within this radius to determine which points are close enough to be considered neighbours. Then, each point was classified into one of the three groups: core point, which had at least 5 points (including itself) within ε, border point, which is within the ε neighbourhood of a core point, and a noise point, which is not a core point, nor a border point, so an outlier. Once clustering was performed, the results were once again visualised with PCA. In Figure 10, it is shown that the data formed a single dominant cluster.

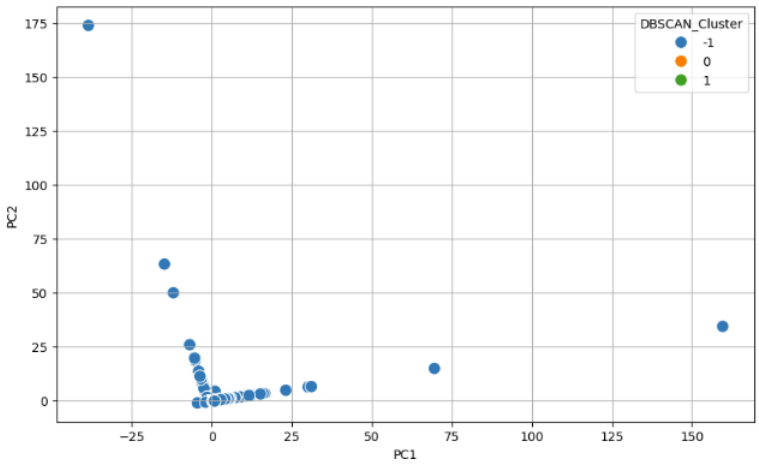


Figure 10. DBSCAN clusters visualised with PCA..

In addition to the dominant cluster, which contained 14,242 points, another cluster with only 4 points was created. This is likely an artificial cluster, because it contained only 4 points, when the minimum limit was 5 and it is statistically insignificant. A reasonable share of the dataset with 605 points was marked as outliers. Since only one dominant cluster was found, it fails to segment the data into multiple useful groups for training classifiers and will not be used in the classification process.

4.3 Year-By-Year Clustering and Scalability

To explore how company positioning evolved over time, clustering was performed for each year separately. This decision was based on the nature of the data: since companies do not always report consistently across years, and their financials may change due to external factors (like inflation, taxation policy, or global crises), clustering year by year allowed a more realistic snapshot of financial positioning.

The same preprocessing pipeline (imputation, scaling, clustering, PCA) was applied separately to each year. This not only ensured comparability between years but also made the process scalable for future updates. As new yearly reports are published and exported from Power BI, the same Python pipeline can be reused with minimal adjustments.

4.4 Supervised Learning: Predicting Company Cluster Membership

After assigning cluster labels to all companies for each year, the next step was to build classification models that could predict a company's cluster membership based solely on its

financial metrics. This allowed two important outcomes. Firstly, companies without complete cluster data (e.g., future predictions or new entrants) could still be classified. Secondly, the importance of different financial variables could be interpreted through model feature importance and performance metrics.

4.5 Dataset Preparation

For supervised learning, the clustered datasets were split into training (2019-2021) and testing (2022-2023) subsets. This is not ideal, however, is the most optimal approach in this context. Each year's data was processed separately and then concatenated. This yielded a clean training matrix X_train and test matrix X_test, along with two target labels: y_train_kmeans and y_test_kmeans for KMeans clusters, and y_train_hierarchical and y_test_hierarchical for hierarchical clusters.

4.6 Algorithms and Evaluation Metrics Used

For the algorithms, five different classification models were trained and evaluated: Gaussian Naive Bayes, Bernoulli Naive Bayes, Support Vector Machine, Random Forest Classifier, and Gradient Boosting Classifier. All models were trained twice, once for predicting KMeans clusters, and once for hierarchical clusters. Models were saved using joblib for potential reuse in deployment or dashboard integration.

Naive Bayes classifiers are probabilistic models grounded in Bayes' theorem, assuming conditional independence among features given in the class label [34, 16]. Two main variants, Gaussian and Bernoulli Naive Bayes, differ in how they handle feature distributions. The Gaussian Naive Bayes (GNB) classifier assumes that the continuous features follow a normal distribution and estimates the probability of an instance from the distance of its z-score to the class mean. This is particularly useful when feature values are normally distributed and real-valued [34]. In contrast, Bernoulli Naive Bayes (BNB) operates on binary or boolean features, modelling the presence or absence of attributes. This makes BNB more suitable for tasks such as text classification, where features may be represented as word occurrences [34].

Despite their simplifying assumptions, both models perform robustly in various domains. Uludağ and Gürsoy [15] applied Naive Bayes to financial risk modelling, where it achieved up to a 86% classification accuracy when predicting company insolvency using Altman Z-score parameters. Similarly, Ampomah *et al.* [35] demonstrated that GNB, especially when paired with techniques

like PCA and LDA, can effectively predict stock price movements, achieving top performance in key evaluation metrics such as AUC and F1-score.

Support Vector Machines (SVM) are powerful supervised learning models particularly known for their use of hyperplanes to separate classes in a high-dimensional space. The method seeks to maximise the margin between data points of different classes and has robust regularisation properties, which make it effective in both linear and nonlinear domains [36, 37]. SVMs also benefit from theoretical foundations in statistical learning theory and generalisation, contributing to their widespread adoption in areas like bioinformatics, pattern recognition, and predictive analytics [37]. Mohan *et al.* [38] underscore the utility of SVMs in separating nonlinear financial data structures and improving bankruptcy prediction performance.

Random Forest (RF) is an ensemble learning algorithm that aggregates the output of multiple decision trees to form dependable, high-performing classifiers. Every tree is trained on a bootstrap sample of the data and, for each split, a random subset of features is considered [39, 40]. This minimises overfitting and makes RF particularly efficient at handling large feature spaces [41]. RF's ability to accommodate both classification and regression tasks, as well as handle missing values and noisy data, account for its widespread application in predictive modelling [40].

Gradient Boosting Classifier (GBC) is another widespread method. It builds each new tree to correct the errors made by the ensemble so far [42]. This boosting technique helps produce highly accurate models by emphasising difficult-to-classify data points, and is particularly resistant to overfitting. Pawelek [43] highlights the efficiency of GBC (specifically, eXtreme Gradient Boosting) in bankruptcy prediction, finding that excluding outliers further improved its predictive accuracy. Roy *et al.* [44] similarly note that gradient boosted machines achieved comparably strong results in financial forecasting tasks. Kumar and Garg [42] classified GBC under the broader umbrella of predictive analytics tools, noting that its success hinges on robust statistical foundations and its capacity to handle complex feature interactions.

Once the classification models were implemented, their results needed to be validated. Evaluating ML classification models entails selecting metrics that balance interpretability, robustness, and context-specific relevance [45]. One of the fundamental tools is the confusion matrix, which presents the counts of true positives, false positives, true negatives, and false negatives for binary

or multi-class classification problems [46]. Based on this matrix, metrics such as accuracy, precision, recall, and F1-score are derived.

Accuracy measures the proportion of correctly classified instances but can be misleading in imbalance datasets because it does not reflect class-specific performance [47, 46]. Precision, the ratio of true positives to the total predicted positives, tells us how many of the positive predictions were actually correct [45]. Recall, or sensitivity, computes the proportion of the true positive instances correctly predicted and is particularly valued in contexts where missing a true positive has high cost, such as medicine [48]. When both precision and recall are important, F1 score, which is their harmonic mean, offers a balanced view, especially in skewed datasets [48, 29].

When models are probabilistic or need to be evaluated across various thresholds, Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) metric are widely used [50]. AUC is particularly advantageous as it summarizes performance across all classification thresholds, providing a single value that reflects the model's ability to rank positive examples higher than negative ones [51, 52]. It is especially valuable in imbalance settings where accuracy or F1-score may be insufficient [47].

However, ROC-AUC also has limitations. It may overestimate performance when false positive costs are not aligned with real-world stakes or when decisions depend on specific probability thresholds [53]. Furthermore, the growing use of multi-class classifiers necessitates strategies such as one-vs-all ROC-AUC evaluation, which can reveal per-class performance while maintaining interpretability [52].

5. Results

This chapter presents the results of the clustering and classification phases. The evaluation follows two main directions: first, unsupervised clustering results and their characteristics across algorithms (KMeans, DBSCAN, hierarchical); and second, the predictive accuracy and behaviour of supervised classification models trained to reproduce the clustering structure using financial ratios as input features.

5.1 Classification Results

The goal of classification models was to predict cluster membership based on financial features. Two sets of cluster labels were used: one from KMeans, and one from hierarchical clustering. Five classifications here trained on historical data (2019-2021) and tested on more recent data (2022-2023): Gaussian Naive Bayes, Bernoulli Naive Bayes, Support Vector Machine, Gradient Boosting, and Random Forest.

To understand the economic significance of each cluster, descriptive statistics were calculated per group. Starting from 2019, the KMeans revealed five distinct financial profiles among Estonian companies. Cluster 3 stands out, because it contains extremely high revenue firms (revenue of approximately €323M) with matching average sales per employee. They have a high working capital turnover and moderate leverage which makes them financially robust. On the contrary, Cluster 4 is problematic, the firms have negative equity and equity-to-assets ratios and a negative working capital, which suggest financial distress. Cluster 0 is more balanced with having high liquidity, healthy equity levels and stable profitability, indicating the companies are likely SMEs and are financially sound.

Hierarchical clustering is similar with some distinctions. Cluster 2, like KMeans Cluster 3 also groups the same high-revenue corporations indicating its structural dominance. Cluster 5 aligns broadly with KMeans Cluster 4, demonstrating financial instability with a debt-to-equity of 35.63 and a very low working capital turnover. Cluster 1 s and outlier with extremely high current and cash ratios, which are likely just holding companies. Cluster 3 contains midsized, profitable and liquid firms which is similar to KMeans Cluster 0. This indicates that both companies found core company profiles. The clustering results are shown in Figure 11.

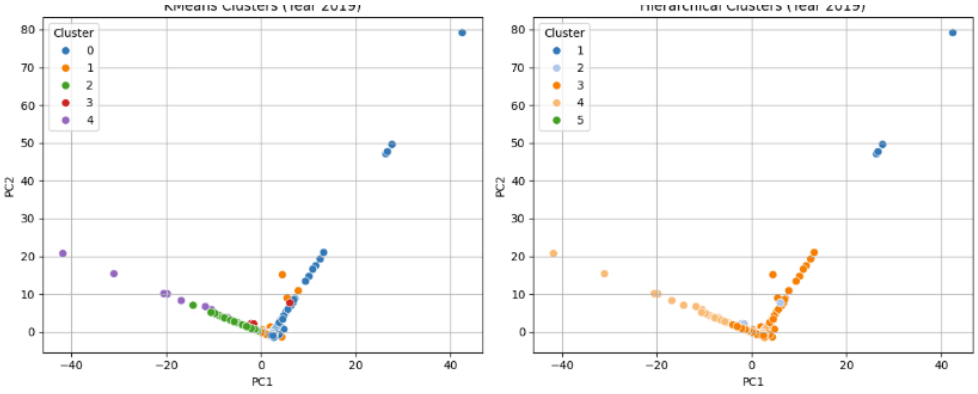


Figure 11. Clusters in year 2019.

In 2020 (Figure 12), KMeans clustering again identifies clear outliers and risk profiles. Cluster 2 is highly unusual, with extremely high cash and current ratios and perfect equity coverage. Cluster 1 shows negative equity ratio and an extreme debt-to-assets ratio of 43.87. Cluster 0 includes healthy mid size firms with good liquidity and decent profitability. Cluster 4 stands out for its very high revenue and profitability.

The hierarchical clustering mirrors this structure. Cluster 1 overlaps with KMeans Cluster 1, where the risk levels are relatively the same. Cluster 5, like KMeans Cluster 2 captures the liquid and passive entries. Cluster 4 aligns with KMeans Cluster 4, grouping high-equity firms, and Cluster 3 identifies stable midsize companies with reasonable metrics.

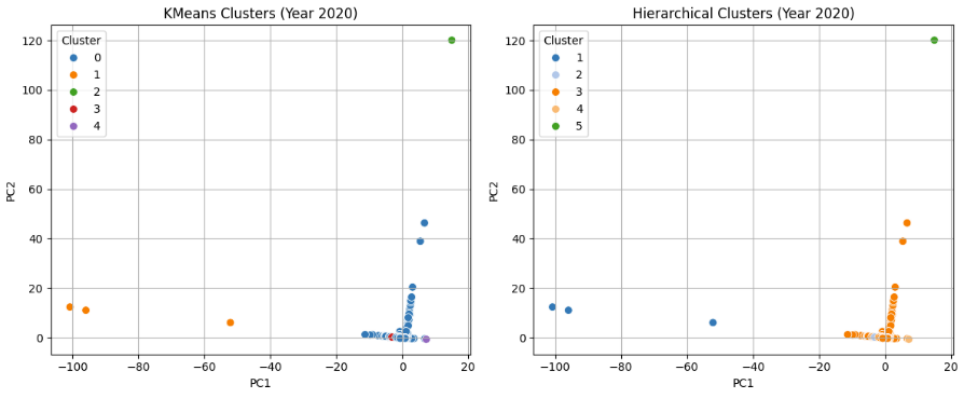


Figure 12. Clusters in year 2020.

In 2021 (Figure 13), KMeans Cluster 4 again contains an outlier group with extremely high liquidity (cash ratio of 5,321 and current ratio of 14,654), minimal debt and very high equity. Clusters 1 and 3 contain highly leveraged or bankrupt firms with Cluster 1 having a debt ratio of

1.0 and a debt-to-equity ratio of (4,968) and Cluster 3 having near-zero liquidity and equity. These patterns are consistent with earlier results. Cluster 0 is once again the healthiest and wealthiest, with a strong revenue, high current ratio and decent profitability,. Cluster 2 is mixed since it is profitable with a 30% margin, but has high debt and low equity. Hierarchical clustering reflects this. Cluster 1 overlaps with K Means Cluster 4. Clusters 3 and 5 group extreme outliers with negative equity and very high debt ratios.

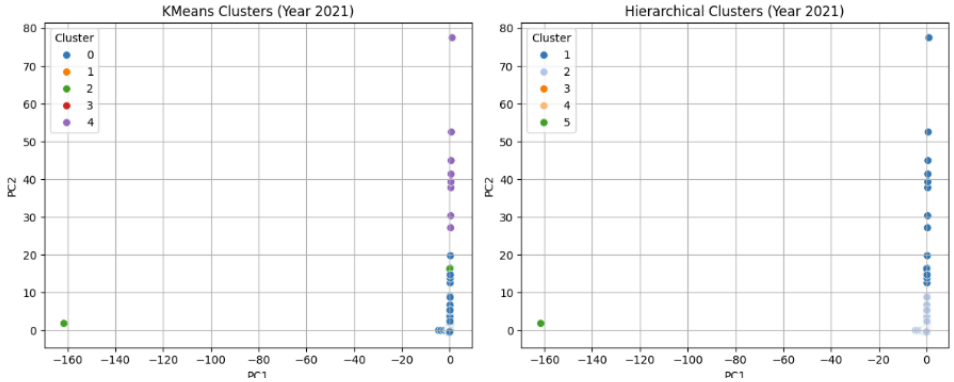


Figure 13. Clusters in year 2021.

In 2022 (Figure 14), the results are relatively the same. KMeans Cluster 0 contains healthy firms with good liquidity having a cash ratio of approximately 9%, low debt and high equity. Cluster 1 has zero liquidity and negative equity. Cluster 2 again represents the outliers that have a cash ratio of 17,475 and current ratios over 30,000, which are highly unlikely, if not impossible. Cluster 3 has a debt ratio of 1.86 and negative equity and working capital. Cluster 4 has the over-leveraged firms with the debt-to equity being 4,074.

Hierarchical cluster once again mirror these results, with Cluster 2 aligning for with KMeans Cluster 2, Clusters 3 and 4 reflecting strong firms with healthy cash and equity, Cluster 1 containing failing firms with high debt and negative equity, and Cluster 5 having the same pattern with KMeans Cluster 1 with zero liquidity and assets.

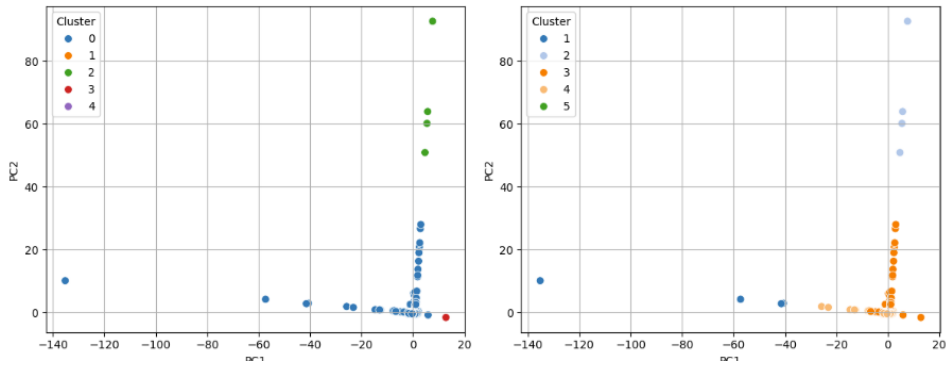


Figure 14. Clusters in year 2022.

In 2023 (Figure 15), the results remain consistent. KMeans Cluster 3 includes high-revenue firms (€315M) with strong liquidity and working capital, Cluster 0 contains stable midsize companies, Cluster 4 consists of financially strong companies that hold a lot of cash, Cluster 1 has insolvent firms and Cluster 2 includes underperformers, with the latter 2 being outliers. Hierarchical Cluster 1 matches KMeans Cluster 3, Cluster 5 reflects KMeans Cluster 1 and Clusters 3 groups underperformers.

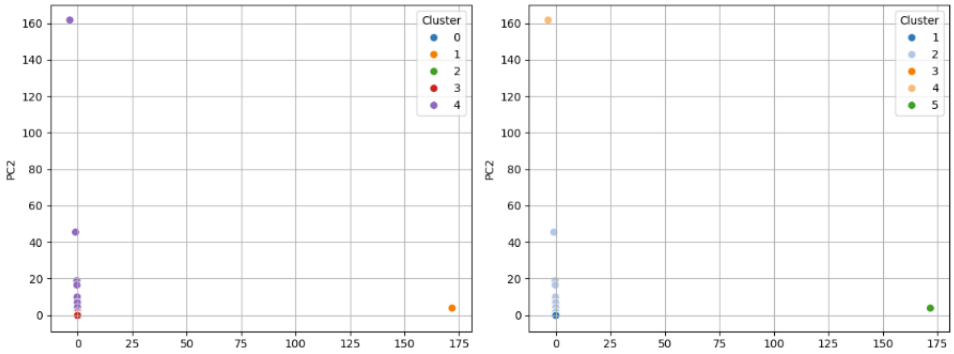


Figure 15. Clusters in year 2023.

5.1.1 Performance on Gaussian Naive Bayes

The classification performance of the GNB model on KMeans-generated clusters was successful in capturing the dominant patterns within the dataset. The model achieved an overall accuracy of 36.05%. It performed well in predicting the largest cluster, Cluster 0, which contained 12,456 instances, for which the model achieved a precision of 0.621, indicating that over 62% of samples predicted as Cluster 0 were indeed correct. More notable, the recall was 0.965, which means that nearly all true instances of Cluster 0 were successfully identified by the model. This high recall

led to a strong F1-score of 0.756, indicating that there is a nice balance between precision and recall. The good performance was also evident in the confusion matrix, where 12,045 out of all true instances were classified correctly.

However, the classifier struggled with smaller clusters. Cluster 4 with 7,408 instances had a recall of only 0.24%, meaning that despite its substantial size, the model failed to identify nearly all of its members. Clusters 1 and 2 had small sample sizes (2 and 5 instances, respectively), and although both had perfect recall of 1.0, their precision was near zero due to a large number of false positives, which is evident from the low F1-scores of 0.0004 and 0.019. Cluster 3 with 60 instances had modest results, with a precision of 0.30, recall of 0.42, and F1-score of 0.35, indicating that the model managed to detect some of its structure but was inconsistent and was leaning towards misclassification. The confusion matrix confirms that much of Cluster 3 was spread across other predicted labels, while most of the misclassified samples were wrongly attributed to Cluster 0.

The imbalance of the cluster sizes had a significant impact on the macro-averaged metrics which do not consider class frequency. Macro precision was 0.205, recall was 0.577, and the F1-score was 0.226. The weighted averages and micro-averaged scores demonstrated better results, but they are biased toward larger clusters.

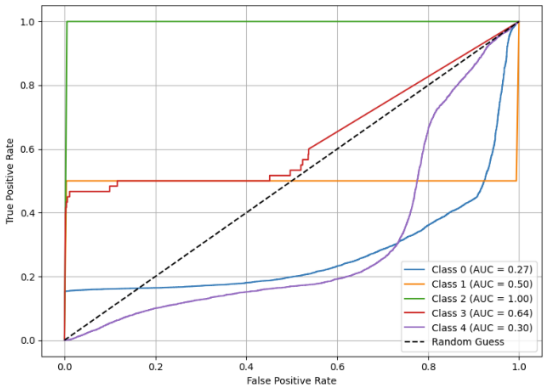


Figure 16. ROC curve of GaussianNB with KMeans clustering.

The ROC curve analysis further supports this conclusion. As seen in Figure 16, although Class 2 showed an AUC of 1.00, it must be considered that this is likely an artifact due to its extremely small sample size (5 instances) and does not reflect the model's strength. Class 1, with only 2 samples, had an AUC of 0.50, which is equivalent to random guessing. Class 0, which proved to be strong in the previous evaluation metrics, had an AUC of only 0.27, suggesting that the classifier

had problems in distinguishing it from other classes. AUC scores for Classes 3 and 4 were 0.64 and 0.30, respectively.

On hierarchical clustering labels, the GNB model performed substantially worse. The accuracy was just 0.01%, demonstrating the inability to generalise the hierarchical structure imposed on the data.

In this methodology, Cluster 2 with 7,993 samples was the only group that could be considered meaningful, with a precision of 0.40, recall of 0.96, and F1 score of 0.56. The problem was that the classifier predicted most test samples belonged to this cluster. All other clusters had an either near-zero recall or zero precision. Cluster 0, with 7,483 samples, had a recall of 0.8% and a F1-score of 0.015. Clusters 1 and 4, which were small in size and probably outliers, had near-zero precision and recall, making their metrics unreliable.

This is also reflected by the macro averages, where the precision was 0.201, recall was 0.294 and F1-score was just 0.118. On the contrary to KMeans clustering, the weighted and micro averages performed poorly here, too. The confusion matrix shows that almost all observations were misclassified as Cluster 1. For example, 7,383 out of 7,483 true instances of Cluster 0 were wrongly predicted as Cluster 1, which had a support of only 5.0. Therefore, it can be argued that the labels could be misaligned. ROC analysis, as shown in Figure 17, further confirmed these observations. The AUC for Class 4 was 0.51 and others were even worse, suggesting very poor class separation.

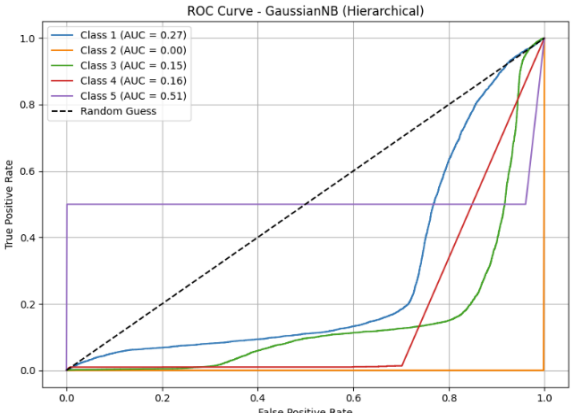


Figure 17. ROC curve of GaussianNB with hierarchical clustering.

5.1.2 Performance on Bernoulli Naive Bayes

In contrast to the GNB model, the following models in KMeans-generated clusters all have a similar pattern: the clusters have been misclassified. The accuracy of the Bernoulli Naive Bayes (BNB) model was 0.421, the second highest, but this is due to overpredicting Cluster 0. The performance of the Cluster 0 is, similarly to the previous model, satisfactory with a precision of 0.539, a recall of 0.696, and a F1-score of 0.608. However, all minority classes were ignored entirely, even though Cluster 4 is quite substantial in size. Clusters 1, 2, 3 and 4 all had a 0% precision and recall. BNB had even worse macro-averaged metrics than GNB. macro precision was 0.108, recall was 0.139 and F1-score was 0.122.

The ROC curve did not indicate this huge misclassification, though, and even had an AUC equal to 0.81 in Class 3 (Figure 18). This demonstrates that the ROC curves can be misleading in imbalanced data, because it shows relatively positive results even when the minority classes were not even predicted.

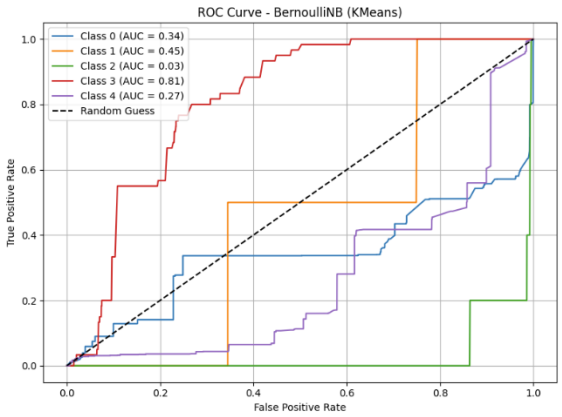


Figure 18. ROC curve of BernoulliNB with KMeans clustering.

On hierarchical clustering labels, the model had an accuracy of 0.58%. Clusters 0 and 2, the biggest ones in size, are the only groups that could be considered meaningful, with precisions of 0.404 and 0.073, recalls of 0.013 and 0.083, and F1-scores of 0.026 and 0.078, respectively. All other clusters had zero recall and zero precision, which could be expected from the very small clusters 1 and 4, but Cluster 3 with 4448 instances having these results is a clear indication of misclassification. The macro-averaged F1-score was just 0.02, which highlights how little the model captured the structure of the clustering. For this one, though indicating much better results than the

aforementioned evaluation metrics, the ROC curve (Figure 19) was more accurate, highlighting that it ultimately failed to create meaningful results.

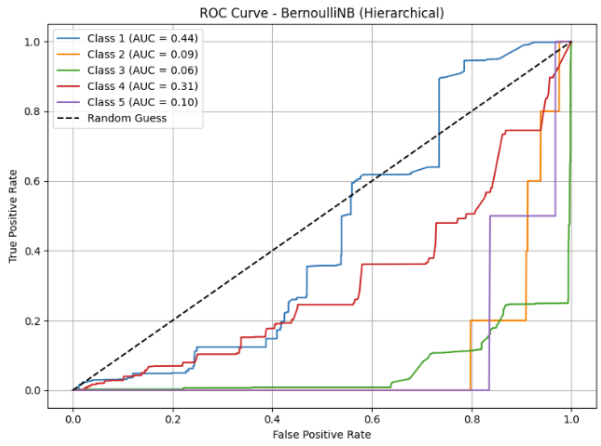


Figure 19. ROC curve of BernoulliNB with hierarchical clustering.

5.1.3 Performance on Support Vector Machine

SVM with KMeans clustering had the highest accuracy among all models at 42.5%. This, however, does not indicate it is the best model in this case, since it again was driven almost entirely by Cluster 0, with a precision of 0.53, recall of 0.68, and an F1-score of 0.60. Clusters 1 to 4 achieved zero precision, recall, and F1-scores, which means that the classifier completely failed to detect any meaningful structure in these groups. This claim is supported by the results of the confusion matrix, which indicates that nearly all samples from other clusters were misclassified as Cluster 0, with only a few data points from Cluster 0 itself being classified elsewhere. Although the micro- and weighted averages were a bit higher, the macro average remained low with the F1-score being 0.12, with which the model's bias towards the dominant Cluster 0 is further supported.

The ROC curve (Figure 20) further illustrates this imbalance. Even though Clusters 1 and 3 show AUC values of 0.99, which indicates almost perfect separation, they are misleading, because the classifier never predicts them. On the contrary, Cluster 0, which showed the highest scores in other evaluation metrics (though again, misleading), has a low AUC of 0.32. Therefore, despite being the most predicted, it has poor separability. Class 4 shows moderate distinction (AUC = 0.55), and Class 2 has an even better result (AUC = 0.74). The model is skewed towards specific minority classes and this makes AUC unreliable.

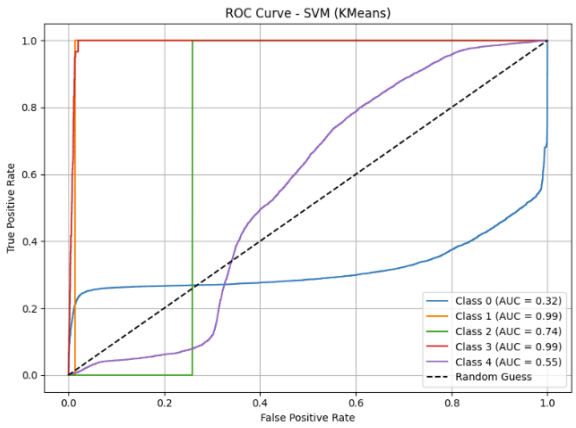


Figure 20. ROC curve of SVM with KMeans clustering.

On hierarchical clustering labels, the performance was again even worse, with an accuracy of 2.41%. This was significantly better compared to the previous methods, but is again biased towards the dominant cluster. Here, the dominant cluster was Cluster 2, with a precision of 0.11, recall of 0.07, and a F1-score of 0.08. The other ones, Clusters 0, 1, 3, 4 achieved, similarly to KMeans approach, zero precision, recall, and F1-scores. Macro F1 dropped to 0.017 and the confusion matrix shows that almost all instances were misclassified into wrong clusters, spread between Clusters 0, 1, 2 and 4 with Cluster 4 being the one that was not predicted at all. The ROC curve with its AUC values for each class are shown in Figure 21. Again, even though Class 5 has a strong AUC, the model ultimately failed to distinguish most clusters effectively.

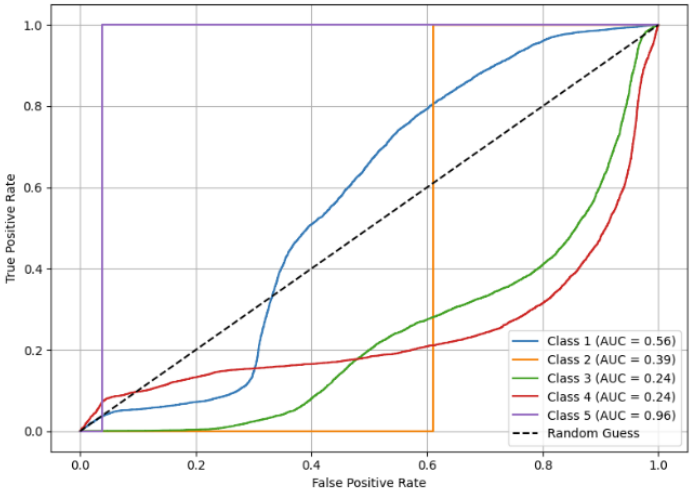


Figure 21. ROC curve of SVM with hierarchical clustering.

5.1.4 Performance on Gradient Boosting Classifier

The Gradient Boosting Classifier on KMeans clusters performed similarly to earlier models. Its accuracy was 39.5%, largely due to its performance on Cluster 0, where it achieved precision of 0.53, recall of 0.67, and F1-score of 0.59. However, it failed completely on Clusters 1-3, and barely recognised Cluster 4, with a recall of 0.04%. The macro-averaged F1 was just 0.12, reinforcing the model's bias toward the majority class and inability to generalise across minority clusters. The notable aspect about the ROC curve (Figure 22) is that when earlier, all models trained on KMeans clusters had at least one class with a near-perfect value, here the AUC values are more similar to each other, ranging between 0.31 and 0.51. This means that the classifier did not favour any single class and performed poorly across all of them.

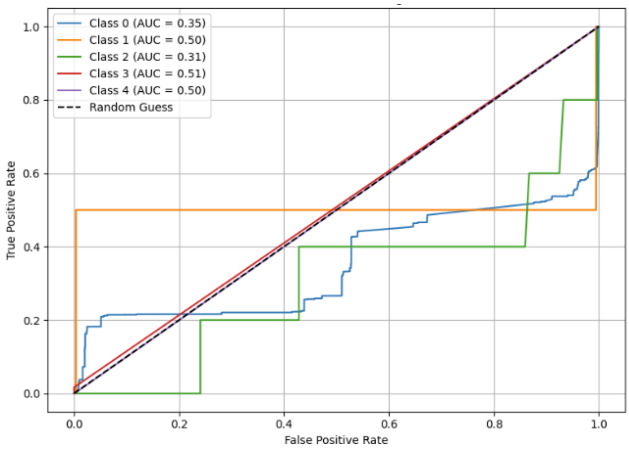


Figure 22. ROC curve of GradientBoosting with KMeans clustering.

On hierarchical clusters, the model's accuracy dropped to 3.5%. Only Cluster 2 saw partial success (F1=0.08), while other clusters had near-zero scores. Notably, even large groups like Cluster 0 had a recall of 0.03%. The curve shapes of the ROC curve as seen in Figure 23and their AUC value suggest minimal true positive gains across false positive rates, most notably so for Class 3.

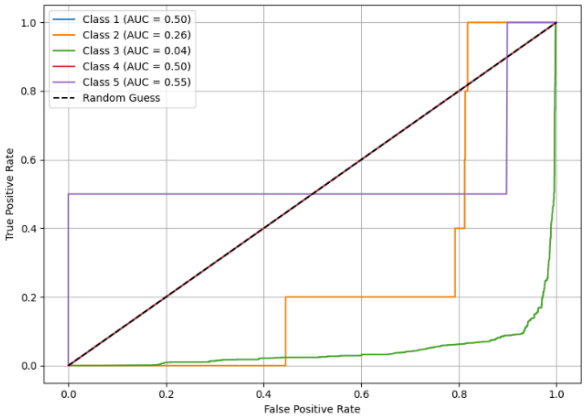


Figure 23. ROC curve of GradientBoosting with hierarchical clustering.

5.1.5 Performance on Random Forest Classifier

The Random Forest Classifier on KMeans clusters showed a similar pattern to previous models, with an accuracy of 39.7%. It performed well on Cluster 0, achieving an F1-score of 0.63 thanks to precision of 0.55 and recall of 0.74. However, it failed entirely on Clusters 1-3, and despite a perfect precision of 1.00 on Cluster 4, this was misleading, as the recall was just 0.03%. Overall, the model overfitted to the dominant class, as seen in the macro F1-score of only 0.13. In the ROC curve visualisation (Figure 24), Class 1 had a moderately higher AUC, but the rest remained close to or below 0.5.

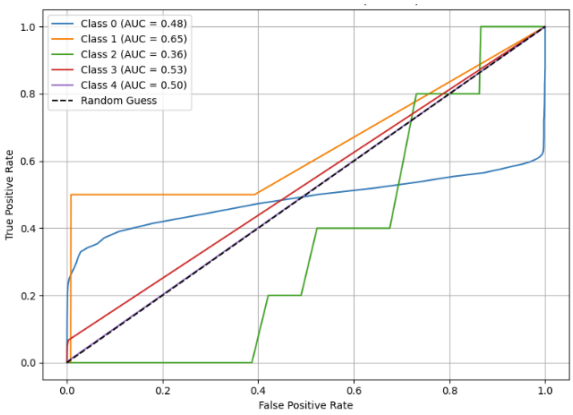


Figure 24. ROC curve of RandomForest with KMeans clustering.

On hierarchical clusters, the model fared very poorly, with an accuracy of 4.4%. Only Cluster 2 had some predictive success (F1 = 0.11), while clusters 0, 1, 3, and 4 were nearly completely ignored. The confusion matrix shows the classifier was mostly predicting Cluster 2 regardless of

true labels. Thus, despite Random Forest's overall power, it was unable to generalise in this multi-cluster setup. The ROC curve (Figure 25) further supported this, with the curves of Classes 2 and 3 being quite flat, highlighting the overall poor predictive ability on the hierarchical labels.

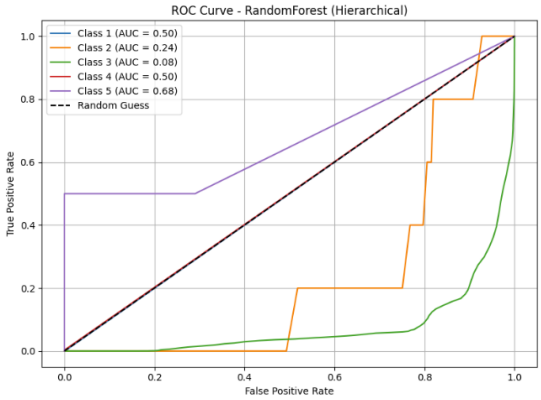


Figure 25. ROC curve of RandomForest with hierarchical clustering.

6. Discussion

Considering the results, it can be argued that KMeans clusters were easier to predict using classification algorithms. This argument can be supported by the fact that in the methodology part, it was found that the dendrogram cut somewhere around the distance of 150, but later, this produced timeout in the classification, and a very small t value had to be chosen instead. This meant that while KMeans produced mostly balanced and interpretable groupings such as stable midsized firms, good performers, outliers, suspiciously liquid firms, hierarchical clustering produced smaller and unbalanced clusters, which was confusing to the classification models and resulted in poor performance.

This, however, does not mean that the KMeans clustering was exemplary here. In the result analysis, it became clear that only the dominant cluster was learned, and in KMeans, all classifiers showed a strong bias toward predicting only Cluster 0, which was the biggest in size. Therefore, this cluster had relatively high recall and precision, but this resulted in nearly zero recall and precision for the other clusters. Taking this into account, the models did manage to capture the majority class structure, but failed to do so for the others and ultimately failed with generalising.

Although hierarchical clustering initially showed slightly better clustering metrics than KMeans, the labels proved nearly impossible to clarify. The accuracy for most models was between only 0.5% and 4.4%, which was largely due to the bias again. Therefore, hierarchical clustering in this context indicated almost total failure. In contrast to KMeans, even larger clusters (Clusters 0 and 2) were not correctly classified.

In addition to the clustering methods used for classification, DBSCAN was also tested, to see if it could be potentially applicable. However, since the data lacks well-separated dense regions, it created only one dominant cluster and a few outliers, which is why it was not used for classification. This, however, demonstrates that the clustering algorithms suit differently in various contexts and not all are suitable for later classification.

Another thing that can be noted from the results was how misleading ROC curves can be. AUC was higher than 0.7 multiple times, even when precision and recall were zero or near-zero. This highlights the fact that ROC measures ranking and has little to do with the correctness of classifications. Therefore, the use of precision, recall, f1-score and possibly more evaluation

metrics is vital for imbalance data, as these were significantly more honest about model failure and ROC curves can evidently be misleading.

Generally, the cluster interpretability was high, although it seems that problems started to arise early in the process. Although financial ratio patterns within clusters, like strong liquidity for some, high revenue for others, were consistent across years, there were many suspiciously high or low results. Appendix I contains the information about the range where the chosen financial metrics usually stay in, and in the clustering process, some values were sometimes over a thousand times higher / lower, or negative in a context where they should not have been negative. Therefore, though consistency was found, the interpretation was suspicious. However, since financial ratio patterns were successfully found, it can be argued that financial data alone is quite enough to reveal meaningful company groups, even without additional information about their sectors. This suggests potential for utilising this framework in benchmarking in the future.

However, it must be noted once again that machine learning failed to learn how to map from ratios to clusters. This is likely because of the aforementioned difficulty that the clustering process introduced artifacts and suspicious values, which contributed to the low results of the classification models. Even advanced models like Random Forest and Gradient Boosting provided weak results, which further proves that the clustering pipeline should consider additional techniques.

With that being said, the key takeaway should be that the cluster size imbalance matters. Since the clusters were quite imbalanced in size with Cluster 0 having 12,456 samples in KMeans, while Clusters 1 and 2 had 2 and 5 instances, respectively, the classifiers were heavily biased, which resulted in overfitting. In future, additional techniques like SMOTE, rebalancing or oversampling could be tried to help with balancing the data.

Moreover, in the beginning, PCA showed reasonable results and confirmed that the clusters were well-separated. However, this was no longer indicated in the learning process. The models could not replicate the boundaries shown earlier, from which one can learn that the evident visual separation does not mean the structure itself is learnable. Therefore, PCA could be used to help with interpretation, but it does not aid with prediction.

Another thing that can be learned from this thesis is to use appropriate metrics for imbalance data. Accuracy and ROC curves may have provided an insight to some extent here, but it was not helpful

due to these metrics suggesting acceptable performance of the models, while recall, precision and F1-scores revealed failure.

6.1 Limitations

Because there is only five years worth of financial report data available, one major constraint throughout the data understanding phase was time coverage. Although this is already substantial, especially given that tens of thousands of rows are added annually, it still restricts the accuracy and universalizability of the model, especially for machine learning tasks like clustering and classification. In order to get meaningful results from machine learning, five years is clearly not enough and no cross-validation was done. If there was at least 5-7 years of training data and 2-3 years of testing data, the models would have generalised it better and would have provided better results. Currently, no longitudinal trend analysis could be done for firms with inconsistent reporting. Also, peer group dynamics, within an EMTAK group, for example, might shift year to year and are harder to model reliably with a short time window. Furthermore, there are still only a few comparable businesses for uncommon EMTAK codes, which creates more limitations in clustering.

Additionally, the dataset did not contain enough raw data (financial indicators) to calculate as many significant ratios as would have been necessary. For example, important ratios such as the quick ratio, interest coverage ratio, return on equity, accounts receivables turnover, inventory turnover, operating cycle, accounts payable turnover ratio, funding cycle, assets turnover ratio, to mention a few, were not possible to calculate, but all of those could have largely contributed to the evaluation of a company's efficiency, profitability and liquidity against its peers. Furthermore, companies submit reports with varying degrees of accuracy. It is still possible to submit a report in PDF format and only have to submit revenue and certain metrics about the stocks or assets. This creates a situation where data for many companies is missing, because it is not submitted in a machine-readable format.

Moreover, a possible limitation was the choice of the threshold, where only companies with revenue higher than €1 million were used for the machine learning process. Although this was justified with the argument that smaller companies may not have their financial statements audited and can submit misinformation, leading to inconsistencies in the model, the threshold excluded micro and startup firms, making it impossible to generalise smaller entities.

7. Conclusion

The purpose of this thesis was to explore how Estonian companies can be positioned against one another using financial ratios and how machine learning can be used to explain this positioning. It began by developing a business intelligence model in Power BI that allows users to benchmark a selected company against others in the same industry and / or similar revenue range. Following this, the data was exported and analysed using clustering and classification techniques in Python.

The clustering phase revealed several distinct company profiles, including healthy mid-sized companies, highly liquid companies, distressed firms with negative equity, and high earners. KMeans clustering performed the best out of the three methods, because hierarchical clustering produced less balanced groups and DBSCAN failed to meaningfully segment the dataset. The classification phase showed that only the dominant clusters could be predicted well. They struggled with minority clusters, which resulted in high recall and precision for one group, but near-zero for the rest. It became apparent that this is due to the limitations of the dataset. Firstly, the data currently spans only five years, which is not enough to find common patterns. Secondly, not enough financial ratios could be computed because the datasets did not contain enough variables, which in turn reduces the completeness of the financial profile. Thirdly, even though smaller companies that would have provided a lot more outliers were removed, the machine learning models suffered from significant class imbalance.

The findings reveal that ratios alone can be used to create clusters that have meaningful economic interpretations. However, the outliers and extreme ratio magnitudes undermined the reliability of these clusters and classification algorithms struggled due to severe class imbalance and short time coverage. The thesis provides a valuable foundation for future work. In order to assess a company's position with machine learning in the future, the annual reports have to span more years and contain richer financial indicators. Then, the current framework can be scaled and improved. Future versions of the model should include oversampling, feature selection or hybrid clustering-classification pipelines. In addition to that, if this research was to be continued as a Master's thesis, the computation power in the Power BI model could be improved and finding better cluster distributions would start by validating them externally, using trimming, robust scaling and isolation forests to handle the outliers better, choosing metrics with similar ranges so high revenues would not interfere with ratios that range between 0 and 1, optimising clustering parameters better,

and trying more algorithms. Ultimately, the positioning of companies is not just a statistical problem, but a strategic and comparative one. By combining interactive BI tools and data-driven models, this thesis shows a pathway toward more informed, transparent, and scalable financial analysis.

References

[1] D. S. Nadar and B. Wadhwa, “Theoretical review of the role of financial ratios,” *SSRN Electronic Journal*, Jan. 2019, doi: 10.2139/ssrn.3472673.

[2] J. L. Gallizo and M. Salvador, “Understanding the behavior of financial ratios: the adjustment process,” *Journal of Economics and Business*, vol. 55, no. 3, pp. 267–283, Apr. 2003, doi: 10.1016/s0148-6195(03)00022-5.

[3] R. Sharma, K. Sharma and K. Apurva, “Study of supervised learning and Unsupervised learning,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 6, pp. 588–593, Jun. 2020, doi: 10.22214/ijraset.2020.6095.

[4] B. Hočevar and J. Jaklič, “Assessing Benefits of Business Intelligence Systems – a case study,” *Management: Journal of Contemporary Management Issues*, vol. 15, no. 1, pp. 87-119, Jun. 11, 2010. Accessed: May 5, 2025. [Online]. Available: https://hrcak.srce.hr/53609

[5] H. Cheng, Y.-C. Lu, and C. Sheu, “An ontology-based business intelligence application in a financial knowledge management system,” *Expert Systems With Applications*, vol. 36, no. 2, pp. 3614–3622, Mar. 2008, doi: 10.1016/j.eswa.2008.02.047.

[6] N. Bolloju, M. Khalifa, and E. Turban, “Integrating knowledge management into enterprise environments for the next generation decision support,” *Decision Support Systems*, vol. 33, no. 2, pp. 163–176, Jun. 2002, doi: 10.1016/s0167-9236(01)00142-7.

[7] G. Deckler, *Learn Power BI: A beginner’s guide to developing interactive business intelligence solutions using Microsoft Power BI*. Packt Publishing Ltd, 2019.

[8] Team of Experimental Statistics, “An early warning services for businesses. Prototype and Business Analysis - Practical Execution,” Statistics Estonia, Apr. 2022. Accessed: May 15, 2025. [Online]. Available: https://realtimeeconomy.ee/sites/default/files/2022-08/Early%20warning%20system%20%28analysis%20and%20prototyping%29.pdf

[9] T. Kliestik, K. Valaskova, G. Lazaroiu, M. Kovacova, and J. Vrbka, “Remaining financially healthy and competitive: The role of Financial Predictors,” *Journal of Competitiveness*, vol. 12, no. 1, pp. 74–92, Mar. 2020, doi: 10.7441/joc.2020.01.05.

[10] H. Abdi, *Encyclopedia of Measurement and Statistics*, 3, pp. 1005-1058. 2007. Accessed: May 14, 2025) [Online]. Available: https://personal.utdallas.edu/~Herve/Abdi-Zscore2007-pretty.pdf

[11] Cialone, “Bankruptcy Prediction by Deep Learning,” *Stanford University CS30 Winter 2020*. 2018. Accessed: May 15, 2025. [Online]. Available: https://cs230.stanford.edu/projects_winter_2020/reports/32569269.pdf

[12] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, “Credit rating analysis with support vector machines and neural networks: a market comparative study,” *Decision Support Systems*, vol. 37, no. 4, pp. 543–558, Jul. 2003, doi: 10.1016/s0167-9236(03)00086-1.

[13] S. Syed, “Financial Implications of predictive analytics in vehicle manufacturing: Insights for budget optimization and resource allocation,” *SSRN Electronic Journal*, Jan. 2024, doi: 10.2139/ssrn.5028574.

[14] M. Karim and R. M. Rahman, “Decision Tree and Naive Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing,” *Journal of Software Engineering and Applications*, vol. 6, no. 4, pp. 196–206, Jan. 2013, doi: 10.4236/jsea.2013.64025.

[15] O. Uludağ and A. Gürsoy, “On the Financial Situation Analysis with KNN and Naive Bayes Classification Algorithms,” *Journal of the Institute of Science and Technology*, vol. 10, no. 4, pp. 2881–2888, Dec. 2020, doi: 10.21597/jist.703004.

[16] V. Nasteski, “An overview of the supervised machine learning methods,” *HORIZONS B*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/horizons.b.04.1.17.p05.

[17] X. Dai and T. Kuosmanen, “Best-practice benchmarking using clustering methods: Application to energy regulation,” *Omega*, vol. 42, no. 1, pp. 179–188, May 2013, doi: 10.1016/j.omega.2013.05.007.

[18] G. Milligan and S. Hirtle , “Clustering and Classification Methods” in I. Weiner, J. Schinka and W. Velicer (eds), *Handbook of Psychology: Research Methods in Psychology*, 2nd ed. John Wiley & Sons, Inc., pp. 189-206, 2013.

[19] “Eesti majanduse tegevusalade klassifikaator (EMTAK) | Statistikaamet.” https://www.stat.ee/et/esita-andmeid/andmete-esitamisest/ettevotete-uuringud/eesti-majanduse-tegevusalade-klassifikaator-emtak

[20] B. Powell, *Mastering Microsoft Power BI*. Packt Publishing, 2018.

[21] M. Cui, “Introduction to the K-Means clustering algorithm based on the Elbow Method,” *Accounting, Auditing and Finance*, vol. 1, no. 1, pp. 5–8, Oct. 2020, doi: 10.23977/accaf.2020.010102.

[22] L. S. Ling and C. T. Weiling, “Enhancing Segmentation: A comparative study of clustering methods,” *IEEE Access*, p. 1, Jan. 2025, doi: 10.1109/access.2025.3550339.

[23] D. Teslenko, A. Sorokina, K. Smelyakov, and O. Filipov, “Comparative Analysis of the Applicability of Five Clustering Algorithms for Market Segmentation,” *2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1–6, Apr. 2023, doi: 10.1109/estream59056.2023.10134796.

[24] S. Ramasubbareddy, T. A. S. Srinivas, K. Govinda, and S. S. Manivannan, “Comparative study of clustering Techniques in market Segmentation,” in *Lecture notes in networks and systems*, 2020, pp. 117–125. doi: 10.1007/978-981-15-2043-3_15.

[25] T. Kansal, S. Bahuguna, V. Singh, and T. Choudhury, “Customer Segmentation using K-means Clustering,” *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pp. 135–139, Dec. 2018, doi: 10.1109/ctems.2018.8769171.

[26] F. A. Mufarroha, I. O. Suzanti, B. D. Satoto, M. Syarief, N. Husni, and I. Yunita, “K-Means and K-Medoids Clustering Methods for Customer Segmentation in Online Retail Datasets,” *Proc. IEEE 8th Inf.Technol. Int. Seminar (ITIS)*, pp. 223–228, Oct. 2022, doi: 10.1109/itis57155.2022.10010135.

[27] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.

[28] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is ‘Nearest Neighbor’ meaningful?,” in *Lecture notes in computer science*, 1999, pp. 217–235. doi: 10.1007/3-540-49257-7_15.

[29] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jul. 2010, doi: 10.1002/wics.101.

[30] F. Murtagh and P. Legendre, “Ward’s Hierarchical Agglomerative Clustering Method: Which algorithms implement Ward’s criterion?,” *Journal of Classification*, vol. 31, no. 3, pp. 274–295, Oct. 2014, doi: 10.1007/s00357-014-9161-z.

[31] F. Nielsen, “Hierarchical clustering,” in *Undergraduate topics in computer science*, 2016, pp. 195–211. doi: 10.1007/978-3-319-21903-5_8.

[32] T. Ali, S. Asghar, and N. N. A. Sajid, “Critical analysis of DBSCAN variations,” *International Conference on Information and Emerging Technologies*, pp. 1–6, Jun. 2010, doi: 10.1109/iciet.2010.5625720.

[33] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN revisited, revisited,” *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, Jul. 2017, doi: 10.1145/3068335.

[34] S. Sushma, TT. Keerthan, “Comparative study of naive bayes, Gaussian Naive bayes classifier and decision tree algorithms for prediction of heart diseases,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 3, pp. 475–486, Mar. 2021, doi: 10.22214/ijraset.2021.33228.

[35] E. K. Ampomah, G. Nyame, Z. Qin, P. C. Addo, E. O. Gyamfi, and M. Gyan, “Stock Market Prediction with Gaussian Naïve Bayes Machine Learning Algorithm,” *Informatica*, vol. 45, no. 2, Jun. 2021, doi: 10.31449/inf.v45i2.3407.

[36] H. Bhavsar and M. H. Panchal, “A Review on Support Vector Machine for Data Classification,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, Dec. 2012, Accessed: May 7, 2025. [Online]. Available: https://text2fa.ir/wp-content/uploads/Text2fa.ir-A-Review-on-Support-Vector-Machine.pdf

[37] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. J. Rajabi, “Advantage and drawback of support vector machine functionality,” *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 63–65, Sep. 2014, doi: 10.1109/i4ct.2014.6914146.

[38] L. Mohan, J. Pant, P. Suyal, and A. Kumar, “Support Vector Machine Accuracy Improvement with Classification,” *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 477–481, Sep. 2020, doi: 10.1109/cicn49253.2020.9242572.

[39] M. Pal, “Random forest classifier for remote sensing classification,” *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, Jan. 2005, doi: 10.1080/01431160412331269698.

[40] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, “Improved random forest for classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, May 2018, doi: 10.1109/tip.2018.2834830.

[41] A. Prinzie and D. Van Den Poel, “Random Forests for multiclass classification: Random MultiNomial Logit,” *Expert Systems With Applications*, vol. 34, no. 3, pp. 1721–1732, Feb. 2007, doi: 10.1016/j.eswa.2007.01.029.

[42] V. Kumar and M. L, “Predictive Analytics: A review of trends and techniques,” *International Journal of Computer Applications*, vol. 182, no. 1, pp. 31–37, Jul. 2018, doi: 10.5120/ijca2018917434.

[43] B. Pawełek, “EXTREME GRADIENT BOOSTING METHOD IN THE PREDICTION OF COMPANY BANKRUPTCY,” *Statistics in Transition New Series*, vol. 20, no. 2, pp. 155–171, Jun. 2019, doi: 10.21307/stattrans-2019-020.

[44] S. S. Roy, R. Chopra, K. C. Lee, C. Spampinato, and B. M. Ivatlood, “Random forest, gradient boosted machines and deep neural network for stock price forecasting: a comparative analysis on South Korean companies,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 33, no. 1, p. 62, Jan. 2020, doi: 10.1504/ijahuc.2020.104715.

[45] R. Yacouby and D. Axman, “Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models,” *2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, Jan. 2020, doi: 10.18653/v1/2020.eval4nlp-1.9.

[46] N. J. C. Obi, “A comparative study of several classification metrics and their performances on data,” *World Journal of Advanced Engineering Technology and Sciences*, vol. 8, no. 1, pp. 308–314, Feb. 2023, doi: 10.30574/wjaets.2023.8.1.0054.

[47] N. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, “Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data,” *2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, pp. 14–18, Oct. 2019, doi: 10.1109/ic3ina48034.2019.8949568.

[48] D. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” *International Journal of Machine Learning Technology*, vol. 2, no. 1, pp. 37-63, Jan. 2020, doi: 10.48550/arxiv.2010.16061.

[49] Q. Gu, L. Zhu, and Z. Cai, “Evaluation measures of the classification performance of imbalanced data sets,” in Z. Cai, Z. Li and Z. Kang (eds.), *Communications in computer and information science*, vol. 51, 2009, pp. 461–471, Pinger, Berlin, Heidelberg. doi: 10.1007/978-3-642-04962-0_53.

[50] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Dec. 2005, doi: 10.1016/j.patrec.2005.10.010.

[51] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997, doi: 10.1016/s0031-3203(96)00142-2.

[52] GeeksforGeeks, “AUC ROC curve in machine learning,” *GeeksforGeeks*, May 12, 2025. https://www.geeksforgeeks.org/auc-roc-curve/

[53] A. M. Carrington *et al.*, “Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329–341, Jan. 2022, doi: 10.1109/tpami.2022.3145392.

Appendix I. The Economic Ratios Used

Ratio / formula used to find the ratio	Explanation of the function of the ratio	Interpretation of the ratio
Average sales per employee Net sales / Total employment	Shows how effectively the company utilises its human resources to generate revenue.	Companies with higher figures are generally considered more efficient than those with lower figures, because a higher ratio indicates that the company can operate on low overhead costs and do more with fewer employees.
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.
Current ratio 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).
Debt quality Current liabilities / Total debt	Indicates what proportion of a company's total liabilities are short-term, due within the next year. This reflects the maturity structure of the company's debt obligations.	A good value depends on the industry and the operating model, but usually the range of 0.2-0.5 is considered healthy and well-balanced, while below 0.2 suggests a company is mostly financed with long-term debt, and above 0.6 signals that the company has a high proportion of short-term debt.
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	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

	risk of potential payment difficulties arising therefrom	the type of the liability. In some cases, even 30% is a high figure; in other cases, this figure may rise up to 80% without substantial issues.
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 focused 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 the 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.
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 / total assets	Shows the share of the assets belonging to the owners of the company in all assets of the company. The lower the 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.
Indebtedness Equity capital / total debt	Shows how much of the company's debt is backed by the equity capital. This reflects the company's long-term financial stability and indicates how well owners could cover the company's liabilities if needed.	The higher the value, the better. A higher value means that a large portion of the company's financing comes from equity rather than debt. However, if the ratio is too high, it could indicate an underutilisation of leverage. A very low

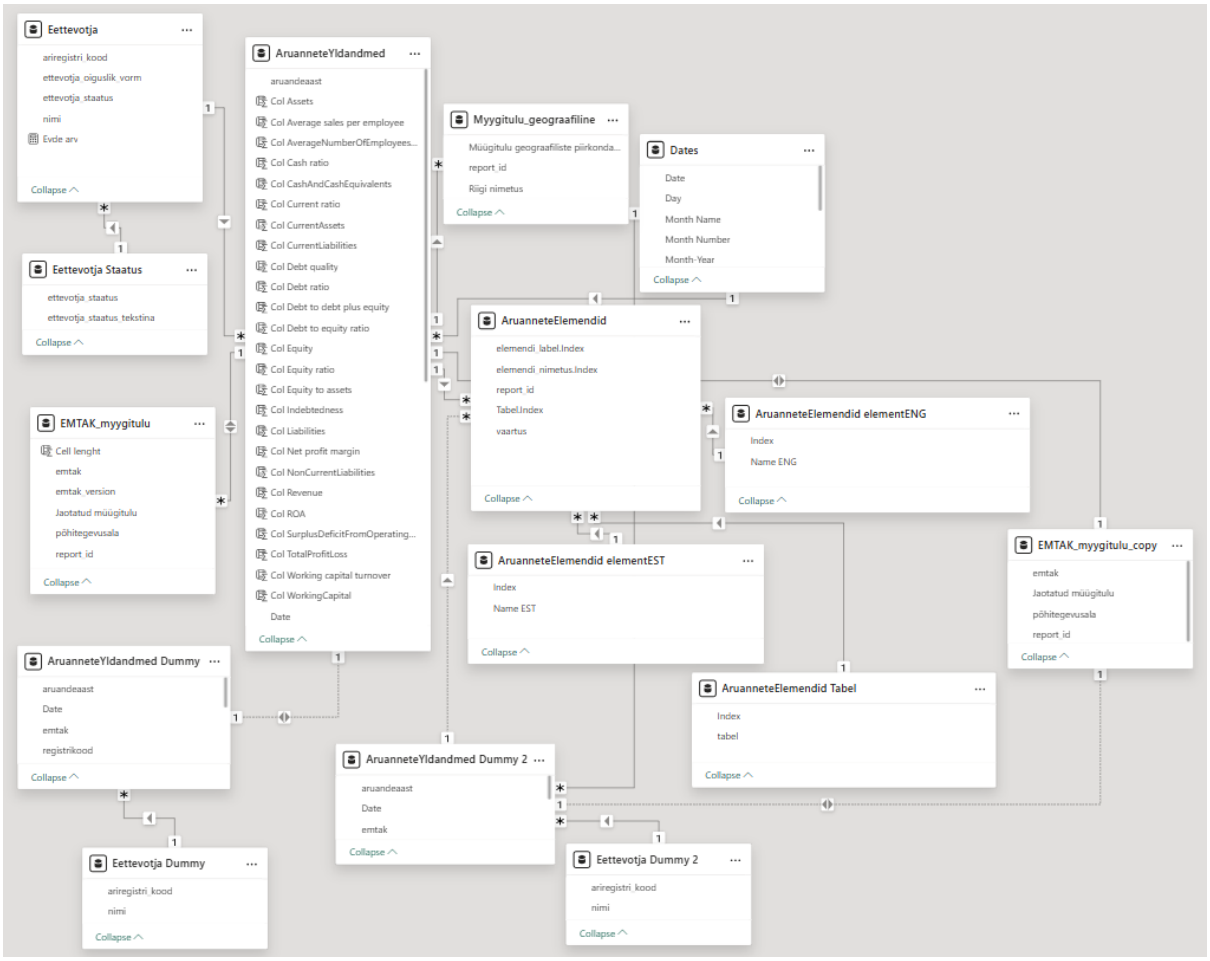
		value means the company relies heavily on debt.
Net profit margin Net profit / sales revenue	Shows the share of net profit in the sales revenue, i.e. the amount of net profit for the company from each euro of the turnover of the company, the share of the company in this turnover. Enables analysing how 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 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.
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).
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 capital a company has, the better. Whether or not the amount of working capital is sufficient can be assessed together with other economic indicators.

Source: Adapted from Team of Experimental Statistics, „An early warning ervices for businesses. Protorype and Busines Analysis – Practical Execution", 2022.

Appendix II. Financial Indicators in The Annual Reports

Name ENG	Index
CurrentAssets	1
Equity	2
IssuedCapital	3
NonCurrentAssets	4
Assets	5
RetainedEarningsLoss	6
TotalAnnualPeriodProfitLoss	7
Revenue	8
DepreciationAndImpairmentLossReversal	9
TotalProfitLoss	10
SurplusDeficitFromOperatingActivities	11
LaborExpense	12
TotalRevenue	13
AverageNumberOfEmployeesInFullTimeEquivalentUnits	14
CurrentLiabilities	15
CashAndCashEquivalents	16
EmployeeExpense	17
DepreciationAndImpairmentLossReversalNeg	18
LiabilitiesAndNetAssets	19
NetAssets	20
NetSurplusDeficitForPeriod	21
NonCurrentLiabilities	22
BusinessIncome	23
	24
IssuedCapital2	25
ServiceFeeIncome	26
EquityConsolidated	27
IssuedCapitalConsolidated	28
NonCurrentLiabilitiesConsolidated	29
NonCurrentAssetsConsolidated	30
CashAndCashEquivalentsConsolidated	31
AssetsConsolidated	32
TotalAnnualPeriodProfitLossConsolidated	33
RevenueConsolidated	34
DepreciationAndImpairmentLossReversalConsolidated	35
EmployeeExpenseConsolidated	36
TotalProfitLossConsolidated	37
DepreciationAndImpairmentLossReversalNegConsolidated	38
LaborExpenseConsolidated	39
AverageNumberOfEmployeesInFullTimeEquivalentUnitsConsolidated	40
RetainedEarningsLossConsolidated	41
IssuedCapital2Consolidated	42
CurrentAssetsConsolidated	43
CurrentLiabilitiesConsolidated	44
ServiceFeeIncomeConsolidated	45

Appendix III. Relationship Visualisation



License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Lisete Müürsepp,

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis “Positioning Estonian Companies with Power BI and Machine Learning”, supervised by Fredrik Milani;

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights or rights arising from the personal data protection legislation.

Lisete Müürsepp

15/05/2025

.