6DATA007W – Final Project Report

Artificial Neural Network for Bankruptcy Prediction

Student: Jesus Daniel Martin – W1834001

Supervisor: Vitaly Fain

This report is submitted in partial fulfilment of the requirements for the BSc (Hons) in Data Science and Analytics at the University of Westminster

School of Computer Science & Engineering

University of Westminster

Date 30th April 2024

Declaration

This report has been prepared based on my work. Where other published and unpublished source materials have been used, these have been acknowledged in references.

Word Count: 6995

Student Name: Jesus Daniel Martin

Date of Submission: 1st of May

Abstract

The goal of this academic project is to examine how effective Artificial Neural Networks (ANNs) are in forecasting the probability of bankruptcy. To achieve this, a dataset from the UC Irvine Machine Learning Repository named "Polish Companies Bankruptcy" has been utilised. The study employs a dataset consisting of financial indicators to accomplish this purpose. The objective is to evaluate the predictive abilities of various ANN structures, all within the context of a Data Science and Analytics course.

The approach involved a series of crucial steps, including data preprocessing, which encompassed addressing missing data and imbalances, designing the model, and creating various neural network structures. Subsequently, progressed to training, validation, and testing. Although the process presented certain hurdles, primarily imbalanced datasets and missing data, these challenges were overcome by utilising effective techniques like SMOTE for oversampling and multiple imputation strategies.

The study revealed that different imputation methods displayed various levels of effectiveness. The results also revealed each model's advantages and shortcomings. Although some models showed good predictive accuracy, specific challenges were noticeable, the need to maintain a balance between precision and recall.

To summarise, this project showcased the obstacles and opportunities that arise when employing ANNs for bankruptcy forecasting. It emphasized the significance of conducting thorough testing and adapting to individual datasets and circumstances.

Table of Contents

6DATA007W – Final Project Report 1
Declaration 2
Abstract 3
Table of Contents 4
List of Figures 5
1. Introduction 6
1.1 Problem statement 6
1.2 Aims and Objectives 8
2. Background 9
2.1 Literature survey 9
3. Legal, social and ethical issues 11
4. Methodology 12
4.1 Data 12
4.2 Methods 12
5. Tools and skills 14
6. Model development 15
6.1 Data preprocessing 15
6.2 Data imputation 18
6.2.1 Mean imputation 18
6.2.2 Knn imputation 18
6.2.3 Linear regression 18
6.3 Data split and standardisation 18
6.4 Data imbalance 19
6.5 Neural Networks 19
7. Results analysis and discussion 21
8. Conclusions and reflections 23
9. References 25
Appendix I 26
Link to GitHub and video 26
List of Attributes 26

List of Figures

Figure 1 - Mind map.....7
Figure 2 - Sparsity matrix 16
Figure 3 - Heatmap 17
Figure 4 - Table of metrics 21

1. Introduction

Although bankruptcy prediction can be challenging, it is essential for many economic stakeholders. Banks and investors rely on it to identify potential financial risks and take necessary measures to avoid them. This helps minimise losses and allocate resources to more promising ventures, creating a healthier financial ecosystem. Addressing any financial weaknesses before they spiral out of control serves as a warning for businesses. This can be done through debt reorganisation, restructuring, cash management, or exploring alternate revenue streams to avoid bankruptcy. For investors, it is crucial to be aware of investment options and to monitor and evaluate the performance of their assets effectively. Such knowledge and control can facilitate the identification of profitable opportunities and improve the overall efficiency of their investment strategies.

While it is impossible to predict a company's future economic health with certainty, obtaining weighted forecasts can be incredibly valuable. By refining models and continuously analysing financial data, we can move closer to a more secure and prosperous economic environment for everyone involved.

1.1 Problem statement

Bankruptcy prediction aims to evaluate a firm's financial health and its prospects for sustained market operations over the long term. This domain encompasses a broad spectrum of finance and econometrics, integrating specialised knowledge concerning the phenomenon with historical data from successful and unsuccessful firms. Typically, companies are assessed using many indicators that encapsulate their business performance. These indicators are net profit, total liabilities, working capital, current assets, EBIT, EBITDA gross profit, net profit, short-term and long-term liabilities, etc. These indicators are then utilised to develop a mathematical model based on prior observations.

Several challenges accompany the prediction of bankruptcy. One primary concern is the prevalence of imbalanced data within historical observations utilised for model training. This imbalance stems from the disproportionate number of successful companies compared to bankrupt ones. As a result, the models developed under these conditions tend to categorise companies as successful (the majority class) even in instances where they are, in fact, distressed firms. Such a bias significantly impacts the overall predictive accuracy of these models. This is one of the reasons why accuracy should not be just the primary indicator when evaluating a model.

Neural networks have become increasingly popular in recent years due to various factors, such as the availability of enormous data, significant advancements in computing power and algorithms, and the broader range of practical applications. The effectiveness of these networks has also been enhanced by developments such as powerful GPUs, improved training techniques including dropout and Adam optimisation, and architectures like CNNs and transformers. Furthermore, the field has seen increased research and investment, driven by successes in high-profile areas such as machine translation and autonomous driving.

One of the biggest obstacles is choosing an architecture that balances complexity and generalisation while avoiding overfitting or underfitting. In addition, high-quality, diverse, and substantial data is crucial to training a neural network. Addressing these

challenges requires a deep understanding of machine learning principles, careful planning, and continuous experimentation.

The following image is a comprehensive mind map detailing the challenges encountered while creating the prediction model and the neural network. Through this structured approach, we aim to provide a clear and organised overview of the factors that must be navigated, offering insights into neural network development.

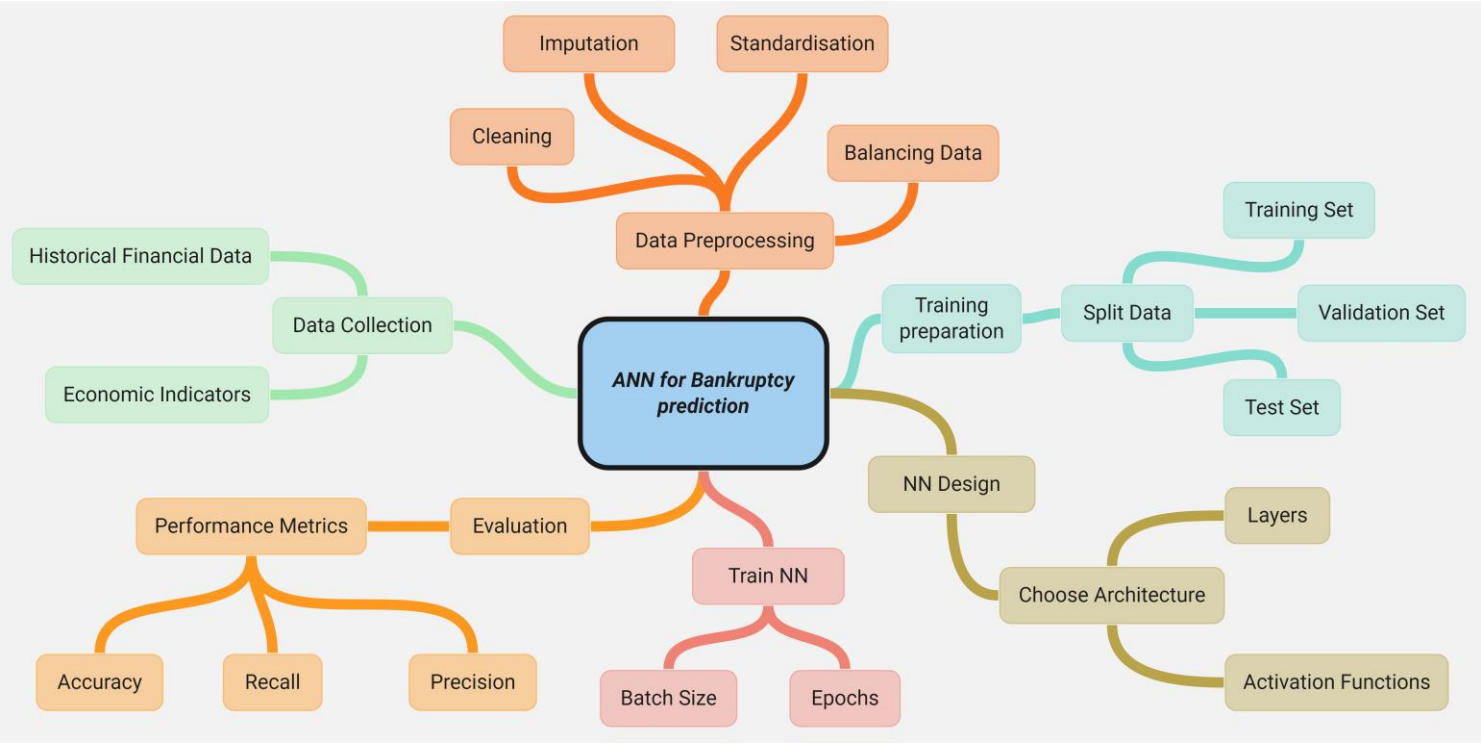

Figure 1 - Mind map

1.2 Aims and Objectives

The project evaluates how Artificial Neural Networks (ANN) can be applied to bankruptcy prediction. The project aims to achieve this by deploying different neural networks in a publicly labelled dataset with financial indicators of Polish companies.

This tool's success could benefit businesses, investors, financial analysts, and policymakers. To ensure a comprehensive evaluation and implementation, the project will be based on the following objectives:

Data Collection and Preprocessing

- Identify a comprehensive financial indicators dataset from bankrupt and solvent companies.
- Conduct data preprocessing to handle missing values and standardise data to improve the ANN's performance.

Model Design and Development

- Design an ANN architecture for bankruptcy prediction, considering factors such as layer number, neuron types, and activation functions.
- Implement the ANN using a suitable machine learning framework, ensuring the model is effectively trained, validated, and tested.

Training and Validation

- Train the ANN model using the pre-processed dataset.
- Fine-tune model parameters and architecture based on evaluation results as accuracy, precision, and recall.

Evaluation and Testing

- To validate its effectiveness and potential improvements, compare the ANN's performance with a different classification model like the random forest.

2. Background

Traditionally, bankruptcy prediction relied on financial ratios and statistical models. In the z-scored model was introduced (Altman, 1968), which quickly became one of the most widely used quantitative models in this field. The Z-score model assessed a company's likelihood of bankruptcy by utilising a set of five distinct financial ratios to generate a score.

The evolution of the finance sector has led to a surge in financial data that requires more advanced analysis techniques. This has prompted researchers and practitioners to explore innovative methods. During the 1980s and 1990s, machine learning techniques emerged as a prominent solution, providing novel ways to manage multidimensional data and complex nonlinear relationships crucial in financial contexts. Models like logistic regression, decision trees, and support vector machines were investigated for their potential to improve prediction accuracy beyond traditional statistical approaches.

In today's big data and advanced computing world, Artificial Neural Networks (ANNs) have emerged as powerful tools for predicting bankruptcy. ANNs are particularly favoured for their ability to model complex and non-linear relationships in financial data. Their adaptability to changing economic conditions and financial indicators makes them highly effective in dynamic environments.

2.1 Literature survey

In 2010, Wu, Gaunt, and Gray conducted research to explore different bankruptcy prediction models. Early models using financial ratios showed effectiveness in predicting bankruptcy within and outside sample data. However, over time, enhancements have been made, notably through integrating market data such as stock returns and using advanced mathematical frameworks like the Black Scholes Merton model.

This research paper evaluates the effectiveness of five prominent bankruptcy prediction models using updated datasets. The study concludes that while all models offer valuable insights into the likelihood of bankruptcy, their performance varies depending on the time of analysis. In addition, the authors introduce a new model that combines crucial elements from the examined models with a unique variable that measures a firm's diversification, which is found to have a negative correlation with bankruptcy risk. The new model outperforms existing models in several tests.

The study analysed data from various sources, including New Generation Research, Compustat, and CRSP, covering firm bankruptcy between 1980 and 2006. The primary variables examined include profitability, liquidity, leverage, firm size, and market-based indicators, such as stock return volatility.

In 2014, a study conducted by Geng, Bose, and Chen used data mining techniques to predict financial distress in listed Chinese companies. The researchers analysed 107 Chinese companies that faced financial difficulties between 2001 and 2008. They used 31 financial indicators across different time windows to identify the indicators that could predict financial distress. The study found that neural networks were more effective than other classifiers such as decision trees and support vector machines in predicting financial distress. The study also emphasized the importance of specific

financial indicators, such as net profit margin and return on total assets, in predicting the deterioration of profitability.

The research emphasizes the significance of having a reliable early warning system to aid companies in taking necessary measures to steer clear of financial crises, as well as allowing investors to modify their strategies to minimize potential losses. This study not only adds to academic knowledge by presenting the effectiveness of various data mining techniques in predicting financial distress but also provides practical implications for corporate governance and investment strategies in China.

In 2020, Clement wrote an article that reviewed 32 articles from databases such as Web of Science, Scopus, and ScienceDirect. The review was based on various criteria such as data source, variable types and numbers, industry type, models used, dataset timeline, sample size, and accuracy of the best-performing model. The articles covered research and publications that were published between 2016 and 2020.

The research has identified two main types of models used in predicting bankruptcy: parametric and non-parametric models. Parametric models, such as logistic regression and different forms of discriminant analysis, have been traditionally preferred because of their simplicity and interpretability. On the other hand, non-parametric models, like artificial neural networks and support vector machines, are known for their ability to model complex nonlinear relationships without making assumptions about data distribution, thus providing more flexibility.

The review points out that there is no agreement on the best approach to predict bankruptcy, as no model consistently performs better than the others when tested on different datasets. However, it does note that there is a growing trend towards hybrid models, which utilize multiple machine learning techniques to take advantage of their individual strengths.

The research highlights the significance of tailoring models to specific datasets and the possibility of using non-financial variables to improve prediction accuracy. The paper suggests that future research should focus on exploring a wider range of model types and incorporating qualitative data to advance the field of bankruptcy prediction.

3. Legal, social and ethical issues

Accurately predicting bankruptcy necessitates a considerable amount of historical data. Yet, disclosing this information can leave companies vulnerable to reputational harm. To address this concern, a technique called anonymisation is utilised. This approach eliminates identifying details that could connect a business to its data.

Also, in complying with data protection regulations such as GDPR, anonymisation of company data is required to ensure the privacy and security of personal and corporate information used to train predictive models.

While anonymisation mitigates risks of reputational damage to companies identified as potential bankruptcy risks, it raises questions about transparency and accountability. This can influence stakeholders' perceptions in decision-making, potentially affecting creditworthiness and investment in businesses.

This highlights the significance of ensuring model accuracy and mitigating biases. Any errors may lead to unfair consequences, affecting businesses, employees, investors, and the economy.

Given that the dataset has been monitoring businesses for five years, it would have been beneficial to anticipate the likelihood of bankruptcy and the timeline for a company's downfall. Unfortunately, the anonymisation process has rendered it unfeasible to follow the financial standing of said companies.

4. Methodology

After researching Kaggle, Google Cloud Public Dataset, Data.World, The World Bank, etc. The dataset selected is Polish Companies Bankruptcy from the UC Irvine Machine Learning Repository.

4.1 Data

This dataset contains the bankruptcy data of Polish companies donated by Tomczak, 2016. This dataset includes an analysis of bankrupt companies from 2000 to 2012 and an evaluation of still-operating companies from 2007 to 2013. The data collection spans five years and is segregated into five annual datasets.

This dataset was chosen for its optimal size and the added complexity of missing data, which presented a welcomed challenge. Additionally, the attributes contained within the dataset are vital financial indicators for businesses, rendering it a prime candidate for modelling. Should the model yield favourable results with this dataset, it may serve as a valuable template for different datasets containing similar indicators.

4.2 Methods

The CRISP-DM (Cross-Industry Standard Process for Data Mining) method has been followed.

Business Understanding: The student is the key beneficiary of this project, which aims to replicate the role of a data scientist in finance and offer practical experience. The ultimate objective is to prepare the student with a valuable portfolio to enhance his chances of employment post-graduation. Therefore, developing neural networks for bankruptcy prediction is necessary and relevant to achieving the project's goals.

Within the project, stakeholders would include companies that can assess their own company's health through financial indicators and investors seeking opportunities to invest.

Data Understanding: It is crucial to understand financial indicators deeply to comprehend how companies operate. The Polish Companies Bankruptcy dataset comprises pertinent financial information such as net profit, total assets, total liabilities, working capital, EBIT, book value of equity, and more.

While intriguing and exciting, this information may not be particularly relevant given that neural networks function as "black boxes." It remains unclear which attributes significantly influence the predictions made by the neural network.

Data Preparation: During this stage, the data is cleansed and modified to make it appropriate for modelling. This process has included analysing the data type inside the dataset, encoding the class attribute, dealing with missing values, imputation, standardising the data, and splitting it into training and testing.

Modelling: The prepared data trains multiple artificial neural networks during the modelling phase. As no evidence exists on the optimal architecture, numerous neural networks are created with different structures to ensure a comprehensive evaluation. The result is the development of diverse models with varying structures, each of which is assessed to determine the model that yields superior results.

Evaluation: A distinct testing dataset will assess how well neural networks predict outcomes. To overcome the challenge of imbalanced data, we will employ metrics

such as accuracy, precision, and recall analysing the performance of the models. It's important to remember that a neural network that forecasts bankruptcy with a 95% accuracy rate is deemed ineffective if the majority class makes up more than 95% of the dataset.

Deployment: Completing this project marks the pinnacle of the Data Science and Analytics student, which focuses primarily on academic exploration. The artificial neural network developed through this project is not currently intended for commercial use. The underlying goal is to showcase theoretical and practical concepts and methodologies in a controlled academic setting.

5. Tools and skills

Python, the primary programming language, provides robust data manipulation, analysis, and modelling tools. Furthermore, it provides access to numerous machine learning-based libraries, which expedite developing and deploying machine learning solutions.

Google Colab was chosen for its ease of use, accessibility, and robust cloud-based infrastructure. This eliminates installing Python locally and dealing with various versions or environments.

Pandas, a Python library, efficiently handles high-performance data frame objects for data manipulation and analysis. It has been used for data-preprocessing tasks such as concatenating datasets, changing class values and checking for missing data.

NumPy is a Python library that provides efficient array operations and mathematical functions. It has been used to change the dataset's data type to *numpy.float32*.

SciPy is a comprehensive library for scientific computing in Python. Used in this project to transform files from an *arff* format to a data frame.

Missingno is a Python library designed to visualise missing data in datasets. It provides insightful visualisations such as bar charts, heatmaps, and dendrograms to help users identify patterns and trends in missing values within their data. Used to create the sparsity matrix and the heatmap.

Sklearn, short for Scikit-learn, is a powerful machine-learning library that can be utilized in Python to construct and execute a diverse array of machine-learning models. In this project, it has been employed for tasks such as data imputation, standardization, partitioning datasets into training and testing sets, and measuring the performance of neural networks.

Keras, a Python-based deep-learning library offers a user-friendly interface for creating and training neural networks. It simplifies the process of building models, defining layers, and compiling networks. Keras has been used for the development of neural network models.

Imblearn, short for Imbalanced-learn, is a Python package that tackles class imbalance in machine learning datasets. The library offers the SMOTE function, which was used to address the imbalance in the Polish dataset.

Grammarly, an AI-powered writing assistant, improved the clarity, correctness, and effectiveness of this report.

ChatGPT is an AI-powered conversational agent that has assisted with coding tasks. It provides helpful code explanations, troubleshoots errors, suggests solutions, and even generates code snippets.

MS Word is a versatile word processing software that enables users to create professional reports with ease. This report has been written in Word.

6. Model development

Google Colab was the base platform for developing the bankruptcy prediction model. Then, the Polish dataset, which consists of five separate datasets, each representing a different year, was uploaded. This marks the beginning of the model development phase.

6.1 Data preprocessing

Having the data divided into yearly intervals poses a challenge due to the varying factors that can impact company performance from one year to the next. The aim is to create a model that can accurately predict bankruptcy based on company performance metrics independently of yearly factors. To achieve this, the five annual datasets have been consolidated into a comprehensive dataset; by doing so, the ANN can recognise and analyse the underlying patterns and correlations that signal potential bankruptcy.

Python was used to implement the whole project. During this step, *pandas* and *numpy* libraries were used to read and concatenate the datasets into one. Also, the data has been shuffled at random to simulate actual data.

The single dataset comprises 43,405 financial statements, each representing a unique record. The data is structured into 65 columns; the first 64 columns denote different features. The type of these columns is *numpy.float64*. For an in-depth understanding of the significance of each feature, kindly refer to Appendix I, listed under the section titled "List of Attributes".

The last column, “class,” categorises the entities into two groups. For companies that are running, *byte 0* has been assigned, and for companies that have bankrupted, *byte 1* has been assigned.

The difference in types will cause an issue, so the column “class” has been transformed to *numpy.float64*, which has been assigned 0.0 for running companies and 1.0 for bankrupted.

After the dataset had the same type, it was changed to *numpy.float32*, as *keras*, a Python library for neural networks, handles this type better than *numpy.float64*.

An exploration of missing data revealed that 23,438 instances out of 43,405 have at least one missing value. This represents 53.998 % of the dataset. This raises questions about where the decision to implement data cleaning and/or imputation strategies must be made. This is to ensure the dependability and applicability of our results.

Visual tools like a sparsity matrix created with the *missingno* library can help us understand the data's distribution.

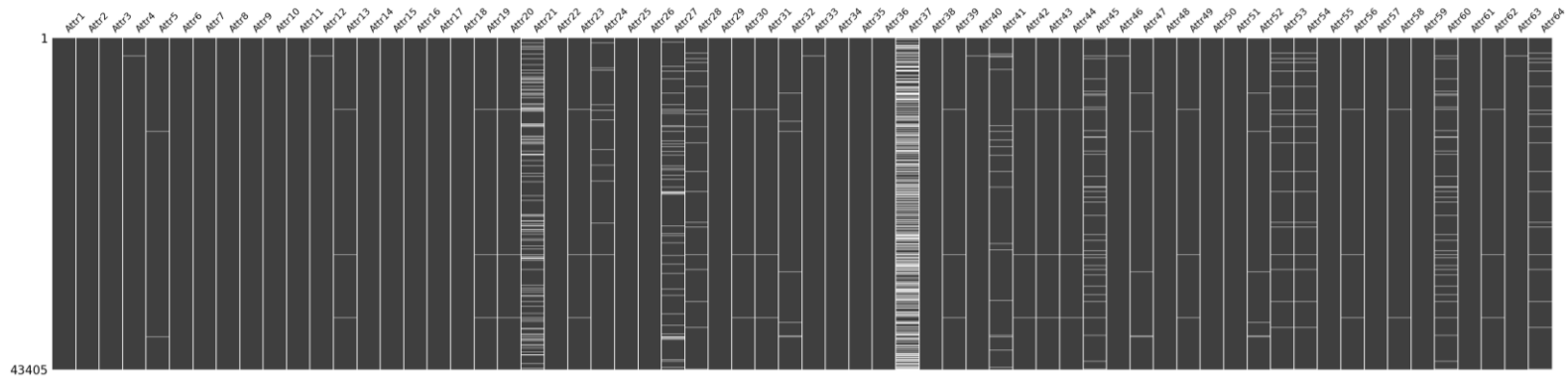

Figure 2 - Sparsity matrix

The graph shows that the 'Attr21' and 'Attr37' columns have the highest number of missing values. An analysis of the correlation between other variables is needed to determine the most effective approach for managing this missing data.

A heatmap on correlation helps determine whether there is any correlation between these two features and any other. The *missingno* library has been used to create the heatmap.

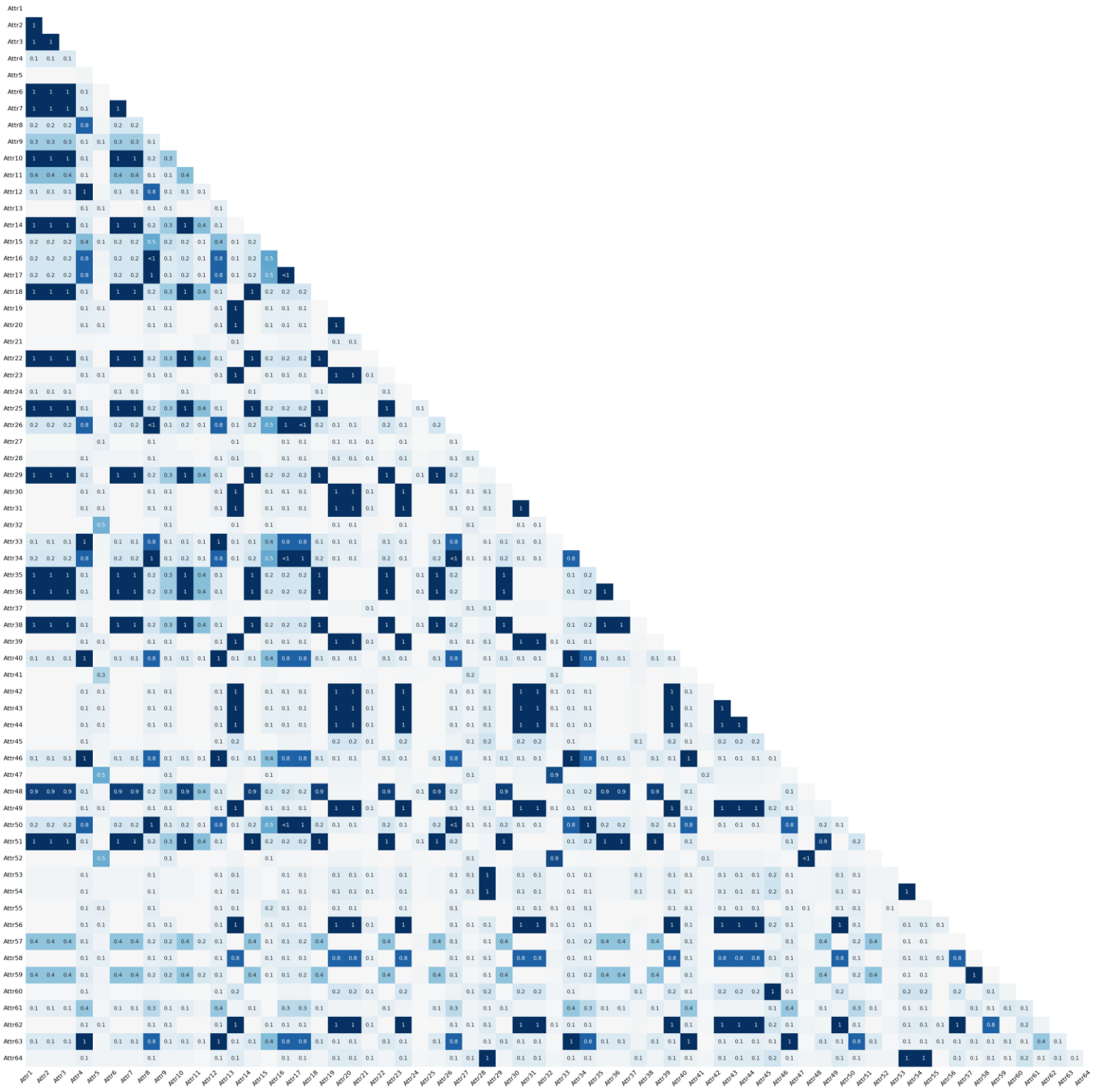

Figure 3 - Heatmap

The heatmap above illustrates the correlation between variables. Notably, 'Attr21' and 'Attr37' are unique variables with no strong correlation to any other variable. These columns are :

Attr21 - sales (n) / sales (n-1)
Attr37 - (current assets - inventories) / long-term liabilities

The lack of knowledge of why data is missing—some data is present, and others not—leads to the conclusion that it is Missing Completely at Random.

Listwise deletion has been discarded as an option; removing the rows with missing data would lead to a significant loss as the data is unique and not correlated with any other variable. Using different imputation models like mean, knn and linear regression and evaluating which is better seems like the best option.

6.2 Data imputation

Imputation is particularly useful in data preprocessing. It ensures that machine learning models, which typically do not handle missing values well, receive complete datasets for training, improving their accuracy and effectiveness.

Three techniques for data imputation have been used: mean imputation, Knn imputation, and imputation by linear regression. After each imputation, a new dataset is created and stored in a list called *dfs*.

6.2.1 Mean imputation

The mean imputation algorithm iterates through columns. For each column, it calculates the mean value, excluding any missing entries. Then, it uses this computed mean value to replace the missing values in that column.

This function's usefulness lies in its simplicity and effectiveness in handling missing data.

6.2.2 Knn imputation

The *sklearn* library was used for the Knn imputation. This algorithm imputes based on the mean of the k-nearest neighbours.

6.2.3 Linear regression

This model predicts missing values based on the values in other columns. This function is particularly useful in complex scenarios with missing data patterns. The linear regression estimator leverages the relationships between variables to make more accurate imputations than more straightforward methods (like mean imputation). This can lead to better performance in subsequent analyses or machine learning models as it tries to maintain the inherent relationships in the data.

6.3 Data split and standardisation

After getting three new datasets with no missing values, the following steps are split into testing, training, and standardisation.

Splitting data into training and testing sets is essential to accurately evaluate the model's performance and ensure it generalises well to unseen data. The training set is used to train the model, allowing it to learn and adapt its weights and biases based on the provided inputs and corresponding outputs. Conversely, the testing set acts as new, unseen data for the model, providing a realistic assessment of how the model will perform in practical scenarios outside the training data environment.

This split helps detect overfitting, where a model might perform exceptionally well on training data due to memorisation or excessive complexity but poorly on new data. The model can be tested to generalise data by having a separate testing set.

Standardisation is crucial because it helps normalise the data within a specific range. This standardisation ensures that each input feature contributes equally to the analysis by having a mean of zero and a standard deviation of one, preventing any feature with more significant numeric ranges from dominating the model's learning process.

6.4 Data imbalance

It is essential to address unbalanced data before training a neural network. This is because imbalanced datasets can lead to biased models favouring the majority class, resulting in poor performance on the minority class. This is particularly problematic in applications such as disease diagnosis, fraud detection, and bankruptcy detection, where the minority class is often more significant despite its fewer examples. To mitigate this issue, techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) are crucial.

SMOTE is an over-sampling technique that creates synthetic instances instead of duplicating existing ones. This approach can help balance the dataset's class distribution and encourage the model to learn more robust features of the minority class, ultimately improving overall model performance.

From the training datasets, 33,062 instances are labelled 0.0 for ongoing companies, and 1,662 cases are labelled 1.0 for bankrupted companies. The majority class accounts for 95% of the entire dataset. After applying the SMOTE technique, the ratio between the two classes becomes balanced, with each class accounting for 50% of the total dataset.

6.5 Neural Networks

The neural network has been developed using the *Keras* library. The primary functions utilised by the *Keras* library are *Sequential* and *Dense*. *Sequential* is employed to construct models layer-by-layer in a neural network, creating a linear stack of layers. It is used with *Dense*, which generates a fully connected neural network layer where each input node is connected to each output node.

A neuron's activation function can be set differently based on the desired output. Here are the activation functions that have been consider:

- *ReLU* (Rectified Linear Unit): The most used activation function, it is known for its simplicity and efficiency. It adds non-linearity by returning zero for any negative input and returning the input directly if it is positive. This process helps to speed up the convergence of stochastic gradient descent.
- *Sigmoid*: The output layer generates values between 0 and 1 for binary classification.
- *Softmax*: When used in the output layer of a multi-class classification problem, the *softmax* function converts the network's output scores into probabilities by normalising the inputs' exponentials.
- *Tanh* (Hyperbolic Tangent): This activation function outputs values between -1 and 1, like the *sigmoid* function, but with a higher range. This can enhance the learning process in a network's hidden layers.
- *Linear*: This function is commonly used in regression problems or when the activation of the previous layer needs to be output without modification, as it doesn't alter the input and is suitable for values not bound to a specific range.

The task at hand involves binary classification, hence the selection of the *Sigmoid* function for the output layer. Additionally, the *Relu* function has been chosen to activate neurons in the hidden layers as it accelerates the convergence process.

The models in this problem have common structure characteristics in their input and output layers. The input layer has a dimensionality of 64, corresponding to the dataset's number of features. For this binary classification problem, only a single neuron is required, with a *sigmoid* activation function to output a probability. This is suitable for binary classification tasks, and its output can be classified as 0.0 if it is less than 0.5 or 1.0 if it is more significant than 0.5.

The predicted values will indicate if a company is classified as 0.0 (ongoing) or 1.0 (bankrupted).

Six models have been developed:

- Model 01(64 - 128 - 64 - 1)
  The model consists of an input layer that accepts 64 features, followed by a hidden layer with 128 neurons and another hidden layer with 64 neurons. Finally, an output layer with a single neuron using a sigmoid activation function.
  Time fitting the models: 5 minutes.

- Model 01(64 - 128 - 64 - 1) with *BatchNormalisation* and *Dropout*
  The structure of neurons is the same as the previous model, but two functions have been added between layers. *BatchNormalization* is applied to normalise the activations of the prior layer, helping to maintain a stable distribution of activations throughout the training process. The *Dropout* layer randomly sets 50% of the input units to zero during training to prevent overfitting.
  Time fitting the models: 6 minutes.

- Model 02 (64 – 128 – 128 – 128 – 64 – 64 - 32 – 32 – 1)
  This model consists of a neural structure that has seven hidden layers. The first three hidden layers have 128 neurons each, followed by two hidden layers with 64 neurons, then two more layers with 32 neurons, and finally, the output layer contains one neuron. This model evaluates a neural network's effectiveness based on the number of hidden layers used.
  Time fitting the models: 9 minutes.

- Model 03 (64 – 32 – 1)
  This model's structure is the simplest: an input layer of 64 neurons, a hidden layer of 32 and an output layer of 1.
  Time fitting the models: 5 minutes.

- Model 04 (64 – 256 – 256 – 32 – 1)
  This model consists of a neural structure with three hidden layers, the first two having 256 neurons and the last with 32.
  Time fitting the models: 5 minutes.

- Model 05 (64_1024_1024_32_32_1)
  In this model, the number of neurons per layer has increased significantly to gauge its potential impact on accuracy, precision, and recall. The model boasts four hidden layers, with the first two comprising 1024 neurons each and the final two housing 32 neurons each.
  Time fitting the models: 5 minutes.

7. Results analysis and discussion

The following table shows the results of evaluating the six models with the different imputations dataset.

		Model 01	Model 01 - Enhanced	Model 02	Model 03	Model 04	Model 05
Mean imputation	Accuracy	0.935	0.926	0.920	0.830	0.916	0.9407
	Precision	0.395	0.300	0.329	0.192	0.317	0.4243
	Recall	0.611	0.382	0.585	0.758	0.611	0.5618
Knn imputation	Accuracy	0.916	0.855	0.906	0.758	0.895	N/A
	Precision	0.256	0.138	0.246	0.136	0.233	
	Recall	0.371	0.366	0.438	0.725	0.492	
Linear regression imputation	Accuracy	0.893	0.616	0.923	0.794	0.925	N/A
	Precision	0.279	0.102	0.340	0.163	0.345	
	Recall	0.741	0.870	0.604	0.765	0.578	
	Time	5min	6min	9min	6min	10min	22min

Figure 4 - Table of metrics

Critical observations on the different imputations' methods:

Mean imputation is a reliable method for balancing recall and precision. It consistently provides high accuracy across multiple models. However, its precision may be comparatively low in specific models such as Model 01-Enhanced at 0.300. Despite this limitation, mean imputation remains valuable for managing missing data and optimising model performance.

KNN imputation has moderate recall values but lacks precision, making it less effective than other imputation techniques.

The effectiveness of Linear Regression Imputation is known to vary, with its optimal performance observed in recall across most models, primarily in Model 01-Enhanced at 0.870. This indicates the algorithm's ability to capture relevant instances effectively; however, it often comes at the cost of precision, with as low as 0.102 observed in Model 01-Enhanced. The accuracy rate of this imputation method is relatively low overall.

Key observations in the models:

- Model 01 has the best accuracy ratio versus time, with an accuracy of 0.935 and a time of 5 min. It also has the second-highest accuracy after Model 05. The precision is insignificant, with a low value of 0.395; this suggests that many non-relevant instances are incorrectly labelled as relevant. Its recall is moderate, with a value of 0.611, indicating it successfully identifies a good proportion of relevant cases.

- Model 01—Enhanced's accuracy drops to 0.926, the third-highest value. There is a notable drop in precision and recall compared with Model 01 with the mean imputed data. A recall value of 0.870 is achieved with the linear regression imputed dataset. This outcome is significant in situations where even a relevant instance cannot be missed.

- Model 02 demonstrates high accuracy, exceeding 0.9 when utilising all three imputation techniques. However, the precision is comparatively low, reaching a maximum of 0.340 in the linear regression imputed dataset. The recall, while moderate, achieves its highest value of 0.604 in the linear regression imputed dataset. It is worth noting that the computational time has increased by up to 9 minutes.

- Model 03 achieves its highest accuracy of 0.830 when using the mean imputed dataset. However, the precision values are shallow, with the best value of 0.192 also being with the mean imputed data. On the other hand, the recall values are moderate, with the linear regression imputed dataset giving the best value of 0.765 and the mean imputed dataset giving a value of 0.758. The total computational time for fitting the models to the datasets is 6 minutes.

- Model 04 attained a peak accuracy score of 0.925 upon being trained on the linear imputed dataset. Nonetheless, the precision values are suboptimal in all datasets, with the linear imputed dataset yielding the highest value of 0.345. As for the recall, values remain consistent in all datasets, with the mean imputed dataset registering the best score of 0.611. The computational time required to run this model is 10 minutes.

- The Model 05 is the most complex model, with the most neurons and connections. However, its complexity takes a toll on computational time, requiring 22 minutes for a single dataset. Because of this, the model has only been trained on the mean imputed dataset. Despite the time-consuming process, the model has achieved the highest accuracy score of 0.947, surpassing all other models. However, this model's precision and recall values are moderate and tend to be low, standing at 0.4243 and 0.5618, respectively.

The outcomes received are rather disappointing. When the dominant class makes up over 95% of the data set, an accuracy rate below 95 is deemed insignificant. Regrettably, none of the experimented models achieved the expected accuracy level, and the precision and recall values were significantly lower. As an individual who has delved into machine learning and neural networks and learned about their potential to transform the deep learning industry, these outcomes have underscored the intricacies of this field. ANNs are not the panacea for all challenges in this arena.

8. Conclusions and reflections

This project has prompted us to consider alternative approaches, and revisiting previous steps to achieve new outcomes may be worthwhile.

Dealing with missing data can be challenging. In this project, imputation data is used as a solution. The two primary missing attributes were:

Attr21 - sales (n) / sales (n-1)
Attr37 - (current assets - inventories) / long-term liabilities

It is worth considering the significance of two features that do not correlate with any other feature. If these features were removed, could the accuracy of the models be improved? Doing so could significantly decrease the amount of imputed data, resulting in more efficient models.

Exploring various imputation models for missing data can be advantageous. However, it is crucial to consider if the missing data is linked to any other variable in the dataset before using linear regression imputation. Linear regression imputation predicts the imputed value by analysing the relationship between the missing variable and other variables in the dataset.

When there is no apparent correlation, this method can generate inaccurate or biased results by creating a connection where none exists. This is evident in the dataset, where the lowest accuracy score of 0.616 was observed with linear regression imputation in Model 01 - Enhanced. Instead, simpler techniques such as mean imputation may be more appropriate since they do not assume any underlying pattern in the data.

During the project, the challenge of unbalanced data was encountered. To tackle this issue, a decision to oversample the minority class until it reaches a balanced proportion with the majority class. This involved generating 31,400 artificial instances, a significant amount and almost equivalent to creating an entirely new dataset. However, it's worth exploring whether under-sampling could have been an effective alternative. One technique that may have been useful in this scenario is the Balanced Bagging Classifier (Mashette, 2023). This approach leverages random under-sampling to balance the class distribution within each subset. By reducing bias towards the majority class, this method can potentially enhance the performance of the minority class.

Two additional algorithms, Random Forest and XGBoost classifiers, were utilized to expand the scope of the bankruptcy prediction project. The mean imputed dataset was used to enable further exploration.

The metrics of these two classifiers are:

Random Forest	Accuracy: 0.9623	Precision: 0.6483	Recall: 0.5198
XGBoost	Accuracy: 0.9762	Precision: 0.8101	Recall: 0.6760
XGBoost (0.9 thresh)	Accuracy: 0.9745	Precision: 0.9815	Recall: 0.4942

These metrics indicate that an accuracy above 95% can be achieved, proving to predict bankruptcy as a success.

It is worth considering the limitations of publicly available anonymised datasets. Such datasets tend to lack exciting information. For instance, in the Polish dataset, if companies had an identifier and a track record spanning over five years, it would be possible to monitor their financial health. This could lead to creating a multi-classifier that can predict bankruptcy and estimate the time remaining before a company becomes insolvent. This is a fascinating idea that has a lot of potential.

To sum up, the realm of data science and analytics is expansive, necessitating individuals to possess a broad understanding to remain current with the continuous emergence of new libraries and resources. Merely mastering one area, such as neural networks, does not guarantee a prosperous path. Being a data scientist and analyst requires acknowledging that a problem can have numerous solutions.

9. References

Tomczak, S. (2016) 'Polish companies bankruptcy data', UCI Machine Learning Repository. Available at: https://doi.org/10.24432/C5F600 (Accessed: 1st of May 2024)

Altman, E.I. (1968) 'Financial ratios, discriminant analysis and the prediction of corporate bankruptcy', *The Journal of Finance,* 23(4), pp. 589-609 Available at: https://doi.org/10.1111/j.1540-6261.1968.tb00843.x (Accessed: 1st of May 2024)

Wu, Y., Gaunt, C. and Gray, S. (2010) 'A comparison of alternative bankruptcy prediction models', *Journal of Contemporary Accounting & Economics,* 6(1), pp. 34-45 Available at: https://doi.org/10.1016/j.jcae.2010.04.002 (Accessed: 1st of May 2024)

Clement, C. (2020) 'Machine Learning in Bankruptcy Prediction – A Review', *Journal of Public Administration, Finance and Law*, 17, pp. 178-197. Available at https://www.ceeol.com/search/article-detail?id=941852 (Accessed: 1st of May 2024)

Geng, R., Bose, I., Chen, X. (2015) 'Prediction of financial distress: An empirical study of listed Chinese companies using data mining', European Journal of Operational Research, 241(1), 236-247. Available at: https://doi.org/10.1016/j.ejor.2014.08.016 (Accessed: 1st of May 2024)

Surya, S., Maddikonda, T. and Keerthi Matta, S. (2018) *Bankruptcy Prediction: Mining the Polish Bankruptcy Data.* College of Computer and Information Science, Northeastern University. Available at: https://www.scribd.com/document/602916267/Bankruptcy-Prediction-Report (Accessed: 1st of May 2024)

Mashette, N. (2023) 'Balanced Bagging Classifier (Bagging for Imbalanced Classification)', *Medium*. Available at: https://medium.com/@nageshmashette32/balanced-bagging-classifier-bagging-for-imbalanced-classification-dfba66c44c14 (Accessed: 1st of May 2024).

Appendix I

Link to GitHub and video

https://github.com/KorvenDalas/FYP_w1834001/blob/main/FYP_w1834001.ipynb

https://drive.google.com/file/d/1AVxlX5ilxx0eg2F4pNtnONuZnYlnGmuJ/view?usp=sharing

List of Attributes

Attr1 - net profit / total assets

Attr2 - total liabilities / total assets

Attr3 - working capital / total assets

Attr4 - current assets / short-term liabilities

Attr5 - [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365

Attr6 - retained earnings / total assets

Attr7 - EBIT / total assets

Attr8 - book value of equity / total liabilities

Attr9 - sales / total assets

Attr10 - equity / total assets

Attr11 - (gross profit + extraordinary items + financial expenses) / total assets

Attr12 - gross profit / short-term liabilities

Attr13 - (gross profit + depreciation) / sales

Attr14 - (gross profit + interest) / total assets

Attr15 - (total liabilities * 365) / (gross profit + depreciation)

Attr16 - (gross profit + depreciation) / total liabilities

Attr17 - total assets / total liabilities

Attr18 - gross profit / total assets

Attr19 - gross profit / sales

Attr20 - (inventory * 365) / sales

Attr21 - sales (n) / sales (n-1)

Attr22 - profit on operating activities / total assets

Attr23 - net profit/sales

Attr24 - gross profit (in 3 years) / total assets

Attr25 - (equity - share capital) / total assets

Attr26 - (net profit + depreciation) / total liabilities

Attr27 - profit on operating activities / financial expenses

Attr28 - working capital / fixed assets

Attr29 - logarithm of total assets

Attr30 - (total liabilities - cash) / sales

Attr31 - (gross profit + interest) / sales

Attr32 - (current liabilities * 365) / cost of products sold

Attr33 - operating expenses / short-term liabilities

Attr34 - operating expenses / total liabilities

Attr35 - profit on sales / total assets

Attr36 - total sales / total assets

Attr37 - (current assets - inventories) / long-term liabilities

Attr38 - constant capital / total assets

Attr39 - profit on sales / sales

Attr40 - (current assets - inventory - receivables) / short-term liabilities

Attr41 - total liabilities / ((profit on operating activities + depreciation) * (12/365))

Attr42 - profit on operating activities / sales

Attr43 - rotation receivables + inventory turnover in days

Attr44 - (receivables * 365) / sales

Attr45 - net profit / inventory

Attr46 - (current assets - inventory) / short-term liabilities

Attr47 - (inventory * 365) / cost of products sold

Attr48 - EBITDA (profit on operating activities - depreciation) / total assets

Attr49 - EBITDA (profit on operating activities - depreciation) / sales

Attr50 - current assets / total liabilities

Attr51 - short-term liabilities / total assets

Attr52 - (short-term liabilities * 365) / cost of products sold)

Attr53 - equity / fixed assets

Attr54 - constant capital / fixed assets

Attr55 - working capital

Attr56 - (sales - cost of products sold) / sales

Attr57 - (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)

Attr58 - total costs / total sales

Attr59 - long-term liabilities / equity

Attr60 - sales / inventory

Attr61 - sales / receivables

Attr62 - (short-term liabilities *365) / sales

Attr63 - sales / short-term liabilities

Attr64 - sales / fixed assets.