HOKKAIDO UNIVERSITY

DOCTORAL THESIS

Bankruptcy Prediction Model Using Machine Learning

By

RASOLOMANANA Onjaniaina Mianin'Harizo

A thesis submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

in the

Division of Modern Economics and Management

Graduate School of Economics and Business

December, 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1. INTRODUCTION AND LITERATURE REVIEW

Predicting that a company is going bankrupt does not mean the company will go bankrupt. In studies about bankruptcy prediction, the goal is rather to get a model with enough predictive power to inform corporate representatives and investors of the current risk, to help evaluate the company, so each concerned party can take the necessary measures in advance.

The aim of the present thesis is to develop a neural network[1] model, which is a sub-branch of machine learning, that can predict corporate bankruptcy. Unlike traditional algorithms, which are hard-coded, a machine learning algorithm is trained on past data to learn the relationships within them and makes future predictions automatically. In the present study, we develop a comprehensive model, trained on not only quantitative data but also qualitative data that are related to the financial situation of the company.

2. Definition of bankruptcy

The term "bankruptcy" used in this research refers to companies that have received a legal order to undergo one of the 3 following procedures:

- Commencement of reorganization proceedings under the Corporate Reorganization Act[2]
- Commencement of rehabilitation proceedings under the Civil Rehabilitation Act[3]

---

[1] The theoretical aspects of neural networks are described in Chapter 2.

[2] According to the Corporate Reorganization Act in Japan, reorganization proceedings refer to the legal procedures "formulating a reorganization plan for a stock company, and implementation of a reorganization plan" when there is a significant risk that the company will be insolvent if it pays its debts.

[3] According to the Civil Rehabilitation Act in Japan, rehabilitation proceedings refer to the formulation of rehabilitation plans in the favor of debtors with financial issues, to ensure "the rehabilitation of the debtors' business or economic life".

- Commencement of bankruptcy proceedings under the Bankruptcy Act[4]

3. Previous studies related to bankruptcy prediction

Research in bankruptcy prediction dates back in the 1930s (Bellovary et al., 2007a). Several studies have been conducted since then, that can be simply distinguished according to the method used (statistical approach, machine learning, deep learning) and the type of data (quantitative, qualitative, financial, non-financial) used to build the prediction model.

Classic studies in bankruptcy prediction have primarily used statistical approach. The pillars in the field include Beaver (1966) which has been the first to determine the usefulness of financial ratios in bankruptcy prediction. Later, instead of the Univariate approach used by Beaver (1966), (Altman (1968), Altman et al. (1977), Blum (1974), Deakin (1972), Edmister (1972) used Multivariate Discriminant Analysis (MDA) to predict failure, also using financial ratios. Then, Ohlson (1980) pointed out the problems when using MDA and developed a Conditional Logit Analysis model to overcome those limitations, again using financial ratios.

Qu et al. (2019) summarizes studies until 2019 that used machine learning and deep learning models in the context of bankruptcy prediction and stated that several studies using machine learning and deep learning have been conducted to predict bankruptcy,

[4] According to the Bankruptcy Act in Japan, bankruptcy proceedings refer to the legal proceedings related to the "liquidation of a debtor's property, inherited property or trust property" in the case the debtor is deemed to be unable to pay his debts.

including Decision Trees[5], Support Vector Machines[6] (SVM), Genetic Algorithm (GA)[7], Ensemble learning[8], Neural Networks, and hybrid models. However, there are some articles not cited in their review but worth mentioning: we can name Shirata (2019), which used a decision tree algorithm, also called CART (Classification And Regression Tree) to identify the most predictive financial ratios for Japanese companies and build the Simple Analysis of Failure or SAF2002 model. There is also Min et al. (2006), they proposed the use of a hybrid model consisting of SVM and GA. Their work showed that the performance of SVM-based models can be improved by integrating a GA to optimize feature selection and parameter tuning. They used financial ratios as predictors.

Some studies have used other methods than statistics and machine learning. We can cite Slowinski & Zopounidis (1995) , they used Rough Set Theory[9] using financial ratios and qualitative factors as data to generate if-then rules to classify companies into high or low risk of going bankrupt. Zakharova (2013) attempted to calculate the importance of each pre-identified bankruptcy risk factors (including financial and non-financial factors) using a fuzzy SWOT analysis[10].

---

[5] Decision Trees are a method of classifying groups by dividing them by conditional branches using if-then rules.

[6] SVM is a pattern recognition model that uses supervised learning (use of labeled datasets to train algorithms, as opposed to unsupervised learning), and is a method of constructing a two-class pattern discriminator using linear input elements.

[7] GA is inspired by the evolution of living things. From a set of variables known as “genes”, some variables are randomly selected, then by repeating the process of selection of the fittest, crossover, and mutation, an optimal solution will eventually be obtained.

[8] Ensemble learning is described in Chapter 2.

[9] Rough Set Theory is a theory for classifying inaccurate or incomplete data, such as ambiguous or coarse data. By applying this theory to rule mining, it is possible to find the minimum set of variables necessary to distinguish one sample from another and derive rules from the selected set.

[10] Fuzzy SWOT analysis refers to the application of Fuzzy Theory to the classic SWOT (Strengths Weaknesses Opportunities Threats) analysis. In Fuzzy SWOT analysis, the importance of factors and

4. The gap

Most studies used only financial ratios in their prediction model. However, that presents some limitations. Financial ratios are based on past performance, although plans and performance forecasting play a crucial role in the success of a company. Also, these quantitative data are summarized indicators, but do not provide rich description. On the other hand, using only qualitative data is no better, although it is generally more detailed, it is based on subjective opinions of the analyst, thus, not always based on facts. In practice, decisions are not driven by the analysis of just one type of data, but by complementing all types of significant data.

Another gap identified in the literature is that previous studies do not transcend to industries and time periods. Classic bankruptcy prediction models such as Altman model, Ohlson model and Shirata's SAF2002 model have excluded some cases due to industry differences. Machine learning and deep learning models can overcome this limitation. However, related works have used voluminous data to train their models. Since the number cases of bankruptcy are sometimes limited, there is a need to investigate the usage of neural networks in such context.

5. Research significance and contributions

In this study, no exclusion was made in terms of industry and time frame. We build a model that can generalize well regardless of such data differences. We also show how quantitative and qualitative data can be combined in a single model and highlight the different yet complementary aspects of these data. Previous studies have focused on quantitative data, as it is based on hard facts, making it reliable. However, it is generally limited to past activities and do not provide rich description. On the other hand, qualitative data, although very detailed and often including future performance as well, is generally based on subjective opinion but not always factual. Quantitative and qualitative data are therefore based on different but complementary assumptions, and in this study, we show

---

strategies as the most significant criterion to select the most appropriate strategies can be done systematically.

that using a mixture of quantitative and qualitative data yield better prediction performance than using only one.

Lastly, unlike previous studies, we demonstrate the performance of neural networks when the dataset is small, meaning that the data presented here is highly dimensional since the number of training samples is far smaller than the number of predictors. High dimensionality is a major challenge in statistics and machine learning, making problems become exceedingly difficult as the number of dimensions in the data is high (Fan & Li, 2006). And neural networks are no exception, they are also affected by high dimensional data. However, neural networks can mild that effect when represented in deep and distributed levels (Goodfellow et al., 2016).

6. Research questions and problem formulation

The research questions addressed in this manuscript are formulated as follows:

- How different is the predictive power when using only financial quantitative data? Qualitative data? And both together?
- Which quantitative data are the most useful indicators of bankruptcy for the neural network model?
- Which qualitative data are the most useful indicators of bankruptcy for the neural network model?
- How to effectively combine quantitative and qualitative data?

7. Choice of algorithm

Amongst all machine learning algorithms, we chose neural networks for this research, as they are known to have the best generalization ability, meaning that with enough training, it predicts well even when using unobserved data. In fact, when trained using proper parameter optimization and feature selection method, neural networks yield a high level of prediction accuracy. This is probably why in the last decades, neural networks have been one of the main methods used in developing bankruptcy prediction models, although statistical approaches such as discriminant analysis used to be the preferred method (Alaka et al., 2018; Bellovary et al., 2007b).

8. Outline of the thesis

The present thesis is divided into 7 independent chapters. Following the general introduction, the second chapter will explain the theory behind neural networks, how they learn and why they are used in this research. The neural network models used in the following chapters will also be explained here. Then, in Chapter 3, we develop a neural network model using only financial ratios as input data and investigate its predictive power. Similarly, in Chapter 4, but using only textual information extracted from securities reports in the model, we will explore how useful such qualitative information can be in the context of bankruptcy prediction. In Chapter 5, we will use both data in the same neural network model and describe a novel approach on how to effectively combine them and still maximize the performance of the model. Chapter 6 will address in depth the usage of ensemble methods, and lastly; we will give a general conclusion in Chapter 7.

Chapter 2. OVERVIEW OF MACHINE LEARNING AND NEURAL NETWORK

This chapter explains the relevant concepts that are necessary to understand the analyses conducted throughout this manuscript. Artificial intelligence, machine learning and deep learning are related but not interchangeable concepts. Artificial intelligence refers to the ability of machines to mimic human intelligence, and machine learning, which involves applying statistics over observed data to automatically improve its function, is one way to achieve that. Deep learning, including artificial neural networks, is a branch of machine learning that learns from large amounts of data. Unlike other machine learning algorithms where the features are manually defined, neural networks determine the features automatically.

1. Machine learning

Machine learning makes computers intelligent enough to learn autonomously. While conventional statistics requires human intervention to specify all the knowledge that the computer needs, machine learning automates this process of mapping the inputs (real world knowledge) into output (decision) by identifying patterns from data.

1.1. Techniques in machine learning

The machine learning technique used differs depending on the problem to be solved. The techniques can be categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning. And deep learning algorithms can be applied to all those three categories (see Figure 1).

1.1.1. Supervised learning

Supervised learning is a method of learning that requires the correct 'answers' in the training data. Regression and classification are the most common problems solved by supervised learning.

Regression is the prediction of a sequence of numbers. For example, by learning the relationship between data related to education level and annual income and predicting the level of annual income an individual.

Classification is the process of predicting which class an observed data belongs to. For instance, the present manuscript presents a classification problem. It aims to investigate the relationship between data and class (from data that has been classified as 'bankrupted' or 'non-bankrupted') and predicting whether a new observation represents a bankrupted company or not.

1.1.2. Unsupervised learning

Unsupervised learning is a method of learning that does not require providing the correct answers. It is useful in clustering task and dimensionality reduction, where it is difficult to find underlying patterns in the data.

For example, if many financial ratios are trained using unsupervised learning, it is possible to group them according to their similarities. But since it is not known whether they are financial ratios of a bankrupted or non-bankrupted company, it is necessary to interpret what kind of group is indicated.

1.1.3. Reinforcement learning

Reinforcement learning is a method of learning that rewards or penalizes an algorithm based on its actions in an environment. It is used in game-playing AI such as Shogi AI and Go AI to learn how to play, and in autonomous navigation to learn how to judge situations.

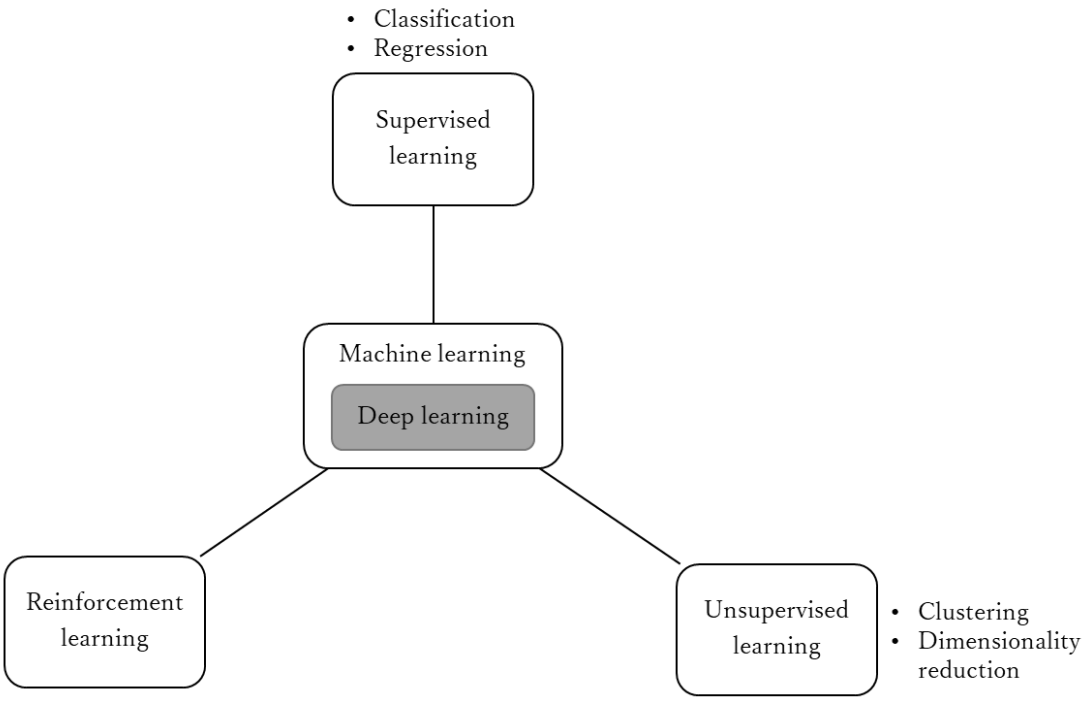

Figure 1: Machine learning techniques and its applications

Source: prepared by the author

1.2. Machine learning workflow

Figure 2 shows the different steps of a machine learning workflow. It starts with the preparation and preprocessing of the necessary data, then we train the appropriate model. After training, the model is cross evaluated on a new set of data, before it is deployed for real-world applications. In this thesis, the work stops at the testing process, the deployment is out of scope.

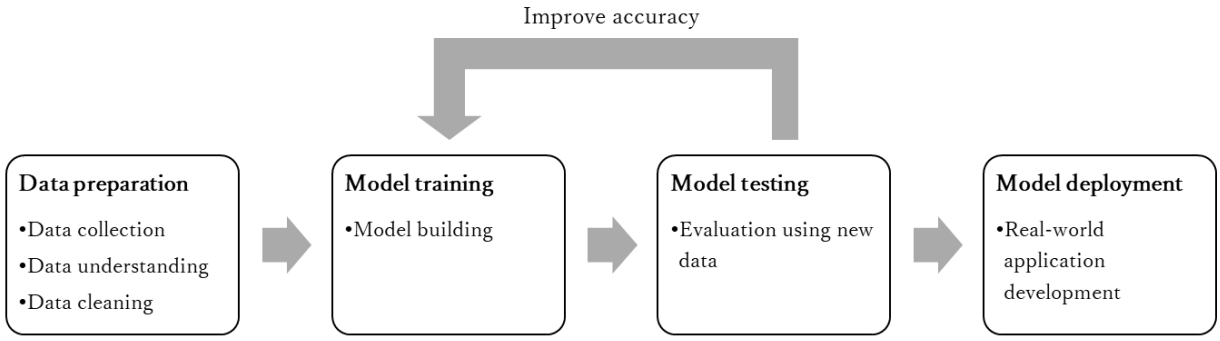

Figure 2: Simplified machine learning process

Source: Hurbans (2020)

1.2.1. Data preparation

This step is very important. Not only the quantity but data quality is also crucial in machine learning. The results of a machine learning algorithm rely largely on the data it is trained on. Therefore, domain knowledge is essential to understand the data to be acquired and how to process it. Here, the data is transformed when necessary, and cleaned of outliers to avoid more bias.

Before it is fed to the model for training, the whole dataset is divided into three subsets[11]: training dataset which is used to train the model parameters, validation dataset which is used to update the parameters and hyperparameters and testing dataset which represents the 'new data'[12] and used to evaluate the model and get its performance metrics.

1.2.2. Model training

After the data is properly cleaned and split, the training dataset is fed to the model. The algorithm varies depending on the problem to be solved. In this research, the task is to classify an observed as 'bankrupted' or 'non-bankrupted', so only algorithms that can solve such problem will be discussed.

Initially, the model parameters are set randomly or with default settings. Then, depending on the performance on the validation dataset, the parameters are iteratively adjusted, until an acceptable performance is reached.

1.2.3. Model testing

If the training performance is satisfying, the model is then evaluated on the testing dataset. If the results are not sufficient, it means that the model does not generalize well, and has probably overfitted the training dataset.

In the case of a classification task, a confusion matrix is used to obtain the performance metrics of the model (Table 1).

---

[11] It is also common to divide the data into two sets only: training and testing datasets.

[12] New data refers to data that has not been used in the training process, so it is 'new' to the trained model. It serves as a real-world test to check the generalization ability of the model.

In the context of bankruptcy prediction, it is relevant to assess the model focusing on bankrupted companies misclassified as non-bankrupted companies, besides the overall accuracy rate.

Table 1: Confusion matrix

		MODEL PREDICTION	
		Bankrupted	**Non-bankrupted**
ACTUAL CLASSIFICATION	**Bankrupted**	True positives	False positives (Type 1 error)
	Non-bankrupted	False negatives (Type 2 error)	True negatives

Source: prepared by the author

Here, the model is asked to determine if the presented data belongs to a company that has bankrupted. So, the above confusion matrix can be explained as following:

- True positives are bankrupted companies correctly classified by the model as bankrupted.
- False positives (or type 1 error) are bankrupted companies misclassified by the model as non-bankrupted.
- True negatives are non-bankrupted companies correctly classified by the model as non-bankrupted.
- False negatives (or type 2 error) are non-bankrupted companies misclassified by the model as bankrupted.

The accuracy rates and the type 1 error rates were calculated according to the following formulas:

$$Accuracy\ rate = \frac{True\ Positives + True\ Negatives}{True\ Positives\ +\ False\ Positives + True\ Negatives + False\ Negatives}$$

$$Type\ 1\ error\ rate = \frac{False\ Positives}{True\ Positives\ +\ False\ Positives}$$

2. Artificial neural network

A neural network is a computing system using machine learning algorithms to map complex functions. Its concept was inspired from how the human brain "learns" patterns from experience and uses it to solve a specific problem, such as predicting future events. Neural networks feed on large amounts of data to autonomously discern relevant features, unlike other machine learning models (Figure 3). It can be used to solve both classification and regression problems.

This section is dedicated to the understanding of the computational methods used in artificial neural networks which is a type of neural network in deep learning.

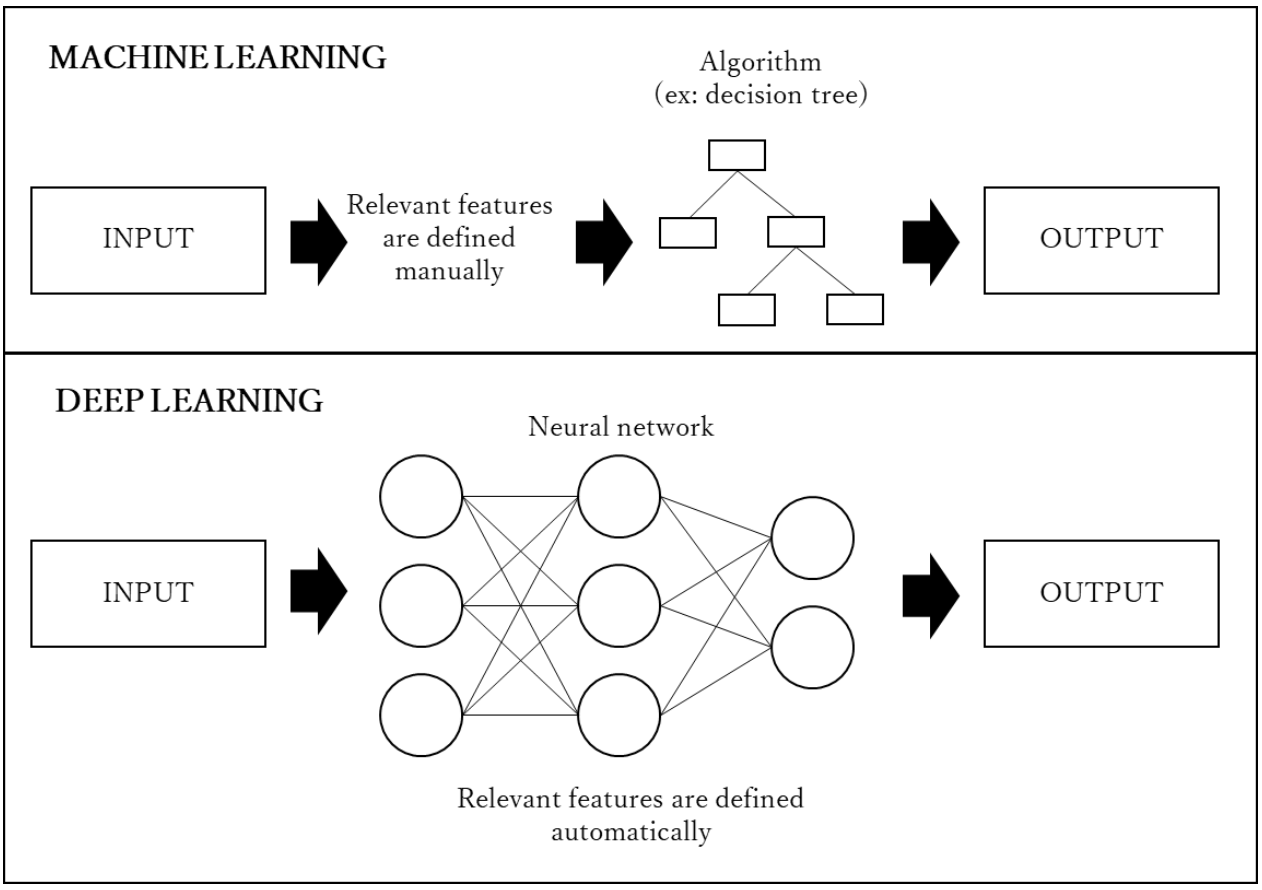

Figure 3: Difference between traditional machine learning and deep learning
Source: prepared by the author

2.1. Elements of an artificial neural network model

Like any other computer system, a neural network takes some input X and brings out an output Y, with some calculations in between. It is organized into densely connected layers.

There are three types of layers: input layer, hidden layer[13], and output layer. Each layer is composed of nodes which represent the computations. Each node in the hidden layer is connected to each node in the input layer and has a different connection strength that is called "weight".

2.2. Learning process

2.2.1. Feedforward propagation

A neural network is trained with large amounts of historical data, then, through a process called "feedforward propagation" (passing the information in order from the input layer to the output layer), it gives an initial prediction of the output. The steps required to provide the results in the hidden nodes are depicted in Figure 4.

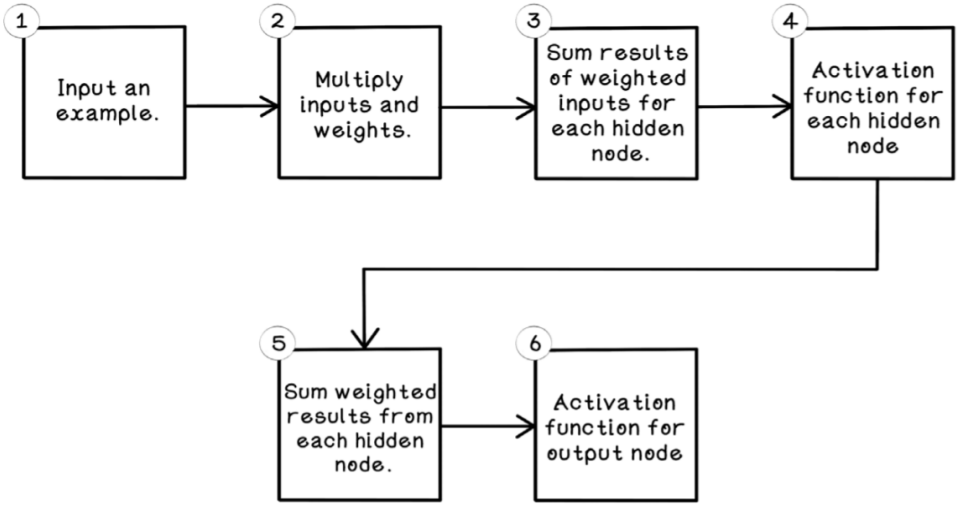

Figure 4: Feedforward propagation in an artificial neural network
Source: Hurbans (2020)

[13] The term "deep" as in "deep learning" is a technical term that refers to the number of the hidden layer (two or more).

A hidden corresponds to the results of activation:

$$Y = F\ (\overrightarrow{w_{nk}}.\ \overrightarrow{x_n}+b)$$

Where F= activation function; w_{nk}= weights; n= number of nodes in the layer; k= number of inputs for the layer and b= bias.

An activation function is the equation that represents one unit (called neuron) in a layer, determining the threshold to whether activate the neuron, and pass the information to the next layer. The ones used in this study are presented in the following table:

Table 2: Activation functions used in the present study

Function	Equation
Sigmoid or logistic function	$\sigma = \frac{1}{(1 + e^{-(\overrightarrow{w_{nk}}.\overrightarrow{x_n} + b)})}$
Rectified Linear Unit function (ReLU)	$R = max\ (0,\ (\overrightarrow{w_{nk}}\ .\ \overrightarrow{x_n} + b))$

Source: prepared by the author

2.2.2. Backward propagation

The predicted output is then compared to the actual output. The resulting error, also called cost or loss, is calculated as follows:

$$C(w,b) = \frac{1}{2k}\sum \|y(x) - Y\|^2$$

Where C= cost function; y= actual value and Y= predicted value.

Depending on how far the predictions were wrong or right, the network updates its parameters (weights and bias) and repeat the cycle until the error is minimized. That updating process is called "backward propagation" (passing the information backwards, from the output layer to the input layer).

3. Ensemble learning

Some models such as decision trees are easy to overfit or unstable such as neural networks, leading to poor performance. Ensemble learning consists of combining multiple poor models into one that performs better. This section describes two major methods of ensemble learning that will be used in Chapter 6 of this thesis: bagging and boosting.

3.1. Bagging

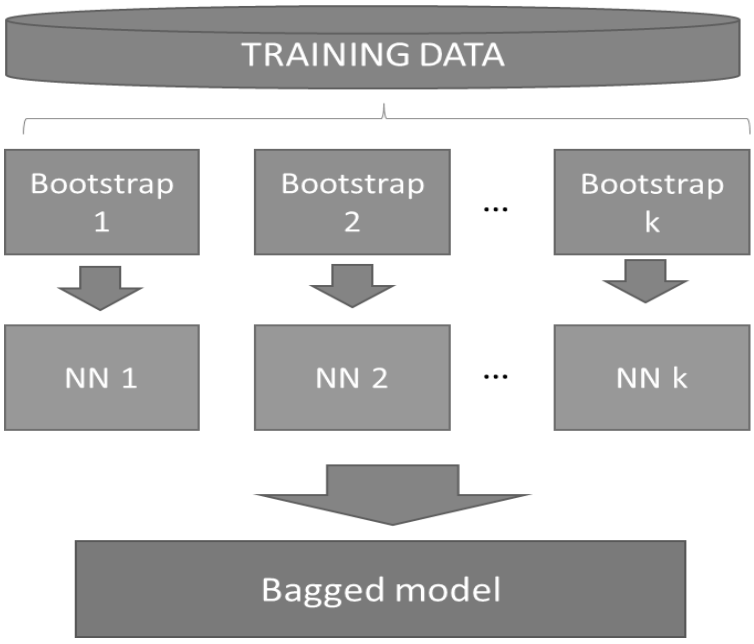

Figure 5: Basic framework of a bagged model

Source: prepared by the author

Bagging, or Boostrap Aggregating, is a technique that uses the overall results of multiple models trained in parallel. Generally, bagging aims to reduce model's variance. One representative approach using bagging is the Random Forest which uses another traditional machine learning model, Decision Trees, as base models. Figure 5 represents an ensemble that combines k neural networks.

3.2. Boosting

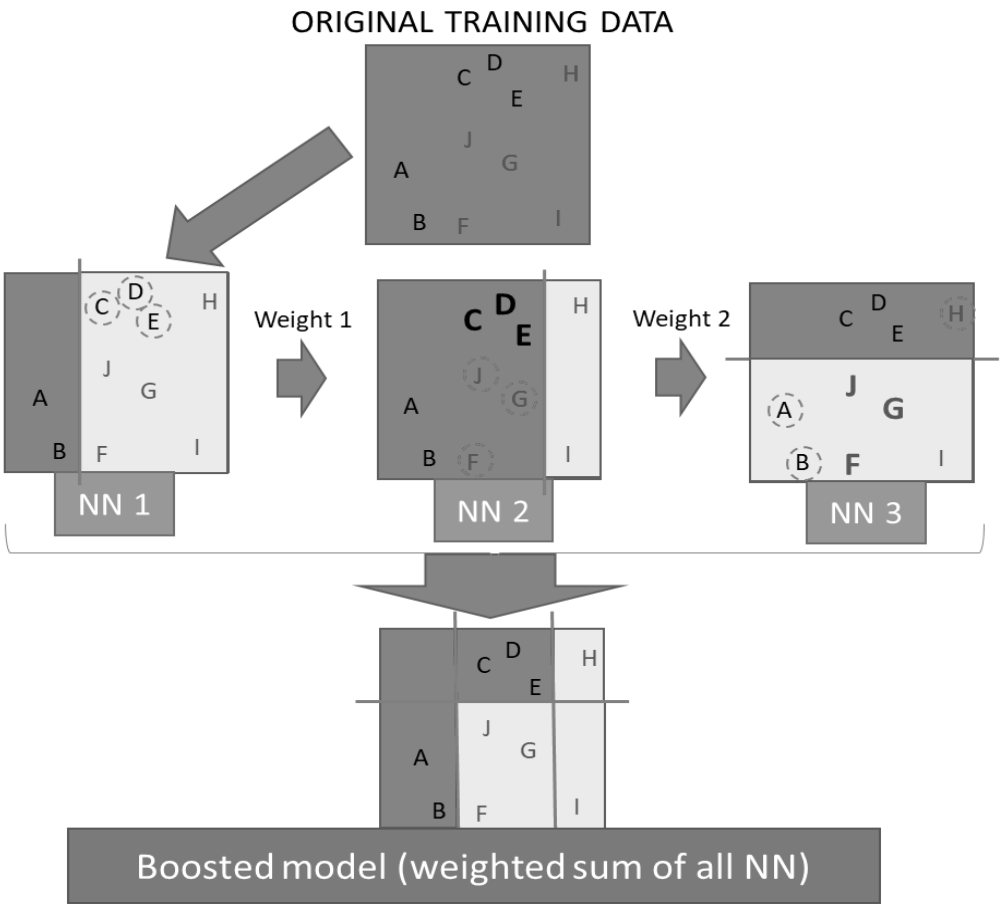

Figure 6: Basic framework of a boosted model

Source: prepared by the author

While the training of the base models in bagging is done in parallel, in boosting it is done sequentially. To estimate the data that were not estimated well by the first base model, we increase their weights before training the next model. And the final model is created by adding weights to the models with high accuracy. Boosting generally aims to reduce the bias. Figure 6 represents an example of boosted model.

Chapter 3. CORPORATE BANKRUPTCY PREDICTION USING NEURAL NETWORKS – USE OF FINANCIAL RATIOS

1. Introduction

Bankruptcy is a major concern not only for the company itself but also for its external stakeholders and has been a matter of interest for academics as well. The financial health of a company is generally assessed through one or both quantitative analysis method and ratio analysis method (Chitoshi, 2014).

Several studies have successfully developed models using financial ratios as predictors of corporate bankruptcy (Altman, 1968; Liang et al., 2016; Tian & Yu, 2017). They used statistical methods such as multiple discriminant analysis and regression analysis to predict bankruptcy. Some prioritized practicability over accuracy. However, in bankruptcy prediction, misclassifying a bankrupted company as a non-bankrupted one is an intolerable error (type 1 error). Therefore, in this study, we prioritized global accuracy and type 1 error over other aspects. Moreover, previous studies have omitted some industries in their model, whereas in the present research, all industries are included. The assumption established in this study is that regardless of the industry, the size or the time frame, a pattern exists in the financial ratios for bankrupted and non-bankrupted companies.

Neural networks, also called Artificial Neural networks, are known for their ability to find patterns in historical data, and provide a highly accurate classification based on them, which is why it will be used in this study. Thanks to the evolution in technology (computation power) and in accessibility of data, neural networks recently gained in popularity within the world of machine learning, although it was developed in the 1940s.

Compared to human beings, the singular benefit of computers is that they are good at calculations; computers are faster and more accurate. Conventionally, computer programs are written to perform a specific task, making it entirely based on the knowledge of the programmer (or instructions given to him). Neural networks, on the other hand, have a different approach. Neural networks are built in a way that it solves problems in a similar

way to the human brain. In that perspective, neural networks can be trained on large volume of data, and through different algorithms, "learn" to solve a problem (prediction, estimation, etc.), with little help from the external environment. The trained network finds itself the best model, which then can be applied to real-world data without having to change the algorithms or the training inputs.

Previous studies have also shown that neural networks can perform better than traditional statistical methods depending on the case and the data, so we compared the performance of the neural network model to the performance of other major bankruptcy prediction models, namely the Altman model and the SAF2002 model developed by (Shirata, 2003).

Therefore, the present research's purpose is to build a neural network model using financial ratios, to predict accurately if a company is in the risk of going bankrupt and compare its performance to other bankruptcy prediction models.

2. Data collection

In this study, we distinguished companies with individual data only and those with consolidated data. In machine learning, dealing with an imbalanced dataset may negatively impact the model performance, therefore, the number of samples in the non-bankrupted group was set to be the same as in the bankrupted group. For consolidated companies, 146 bankrupted and 146 non-bankrupted companies' data were collected. For standalone companies, 114 bankrupted and 114 non-bankrupted companies' data were collected.

The bankrupted companies were all listed Japanese companies before going bankrupt. The bankruptcy period ranges from 2002 to 2019, however, as we want to predict bankruptcy a year before, the data collected was from one fiscal year before bankruptcy. For example, for a company which bankrupted in 2019, the financial data from fiscal year 2018 were collected. To build a model beyond industry differences, we did not exclude any industry.

Table 3: Sample size per industry (bankrupted companies)

INDUSTRY	Individual	Consolidated
Services	17	13
Others (Manufacturing)	9	9
Others (Financial institutions)	8	5
Real estate	34	30
Warehouse / Transportation related business	2	2
Pharmaceutics	2	2
Wholesale	4	2
Retail	11	9
Construction	25	19
Information and Telecommunications	2	1
Machinery	15	11
Fisheries and Agriculture	1	1
Shipping	2	2
Airline	2	1
Textile	6	5
Steel	1	1
Food industry	5	1
TOTAL	**146**	**114**

Source: prepared by the author

The financial data were extracted from the EOL database. 46 financial ratios commonly used in financial analysis were initially collected for both individual and consolidated companies.

Basically, each company in the bankrupted group has its equivalent in the non-bankrupted group. The non-bankrupted companies were chosen randomly from the same database, following some proportion conditions:

- Same number of companies as in the bankrupted companies' group for each industry, as neural networks perform better when the classes in the dataset are balanced
- Same time frame as in the bankrupted companies' group
- Same asset size: the closest value within a range of 1,000,000 JPY was picked but in case there were more than one option, it was chosen randomly using Excel functions

2.1. Structure of the data and construction of the model

In this study, we aimed to determine if a company is in the risk of going bankrupt in the following year. For that purpose, a neural network model has been programmed to learn to classify a company as bankrupted or non-bankrupted, based on financial ratios of the previous fiscal year.

The data consists of financial ratios as inputs X_i and the corresponding classification as output (Y=0 if it is a bankrupted company and Y=1 if it is a non-bankrupted company). Results from standalone companies' data and companies having consolidated data were distinguished.

Neural networks do not perform well with high dimensional and multicollinear data, therefore, we proceeded to the screening of variables using Pearson's correlation analysis between all 46 financial ratios and chose the ones with the highest correlation value with the output variable Y. After screening, we obtained 31 variables for the individual group and 22 variables for the consolidated group.

Table 4: List of variables after screening

INDIVIDUAL (31 variables)		CONSOLIDATED (22 variables)	
X6	Dividend payout ratio	X19	ROE
X7	Dividend Yield	X20	Return on capital employed
X18	operating profit to operating capital	X29	Gross profit margin
X19	ROE	X47	Value Added Ratio
X20	Return on capital employed	X51	Labor sharing ratio
X23	Operating income to net sales	X54	Facilities Distribution Ratio
X27	Cost of sales ratio	X59	Actual capital distribution ratio
X29	Selling and adm. expenses to sales ratio	X62	Public Distribution Rate
X50	Labor sharing ratio	X63	Debt ratio
X51	Facilities Distribution Ratio	X64	Capital adequacy ratio
X52	Actual capital distribution ratio	X67	Interest-bearing debt ratio
X53	Public Distribution Rate	X69	Dependence on interest-bearing debt
X54	Share of other capital	X70	Sales growth rate
X58	Quick ratio	X71	Sales per employee growth rate
X62	Fixed ratio	X72	Gross profit growth rate
X63	Fixed long-term compliance rate	X73	Operating income growth rate
X67	Debt ratio	X74	Ordinary income growth rate
X69	Sales growth rate	X76	Net income growth rate
X70	Sales per employee growth rate	X77	Earnings per share growth rate
X71	Gross profit growth rate	X79	Total capital (total assets) growth rate
X72	Operating income growth rate	X80	Capital adequacy ratio
X73	Ordinary income growth rate	X84	Growth in net assets per share.
X74	Net income growth rate		
X76	Total capital (total assets) growth rate		
X77	Capital adequacy ratio		
X79	Growth in net assets per share.		
X80	Value-added growth rate		
X81	Sustainable Growth Rate		
X83	Dividend per share growth rate		
X84	Operating cash flow ratio		
X100	EBITDA margin		

Source: prepared by the author

The data was then normalized between 0 and 1 to speed up the learning process. Also, the differences in scale in our inputs can generate large values of weights, which affects the generalization performance of the model and slows down the learning, especially when dealing with large datasets.

Practice suggests dividing the dataset into training dataset and test dataset to avoid overfitting[14], thus, ensure the generalization of the model. Accordingly, before beginning the learning process, the data was divided into two sets:

- Training dataset (75% of the data per industry) to train the model
- Testing dataset (25% of the data per industry) to cross-validate the model after each learning iteration

The ratio 75% ・25% was chosen because the dataset is not large. Ultimately, the split was done randomly. However, because of the limited number of samples, to minimize the error, the industry proportion and the data period in each dataset were a priori set identically.

The hyperparameters [15]- number of hidden neurons and number of hidden layers - used in the model were set through a trial-and-error process. Also, to minimize the risk of overfitting and ensure the generalization of the model, the training is automatically stopped when the performance stops improving, but without undertraining the model (Marsland, 2009).

[14] Overfitting in machine learning means that the model has learned to treat every detail and noise in the training data as important, including the ones that are irrelevant, negatively impacting the performance of the model on new data. Besides data split, a complementary way to avoid this is to stop the training when the generalization error increases.

[15] Hyperparameters refer to the settings of the model: the number of hidden layers, number of hidden neurons, number of learning cycle (called epoch), etc.

3. Results

3.1. Evaluation of the model

The performance of the model was evaluated using a confusion matrix, which compares the model prediction with the actual classification.

Table 5: Summary of the results

	Confusion Matrix	Accuracy rate	Type 1 error rate
INDIVIDUAL DATA (n=292)	$\begin{bmatrix} 30 & 5 \\ 13 & 22 \end{bmatrix}$	74.29%	14.29%
CONSOLIDATED DATA (n=228)	$\begin{bmatrix} 17 & 7 \\ 2 & 22 \end{bmatrix}$	81.25%	29.17%

Source: prepared by the author

For individual data, 5 companies (14.29% of the bankrupted group) have been wrongly classified as non-bankrupted. For the consolidated data, 7 companies (29.17% of the bankrupted group) were misclassified as non-bankrupted companies. These results show that the neural network model could learn from past financial ratios to predict bankruptcy with high accuracy. However, there is a necessity to test the model with unseen data (data not included in the model) for validation.

3.2. Additional evaluation

To test the validity of the neural network model, financial data from two Japanese listed companies, RENOWN INCORPORATED and Nuts Inc., which bankrupted in 2020, were run into the trained model to check whether it could classify both correctly. We collected their data from fiscal year 2019.

Table 6: Bankrupted companies in 2020

Company name	Bankruptcy date	Industry	Assets (in million yen)
RENOWN INCORPORATED	30/10/2020	Apparel	32,000 and 29,000
Nuts Inc.	16/09/2020	Retail	1,386 and 1,386

Source: prepared by the author

The results are shown in the following table:

Table 7: Validation results

BANKRUPT GROUP N = 2	Confusion Matrix	Accuracy rate	Type 1 error rate
CONSOLIDATED (22 variables)	$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$	100%	0.00%
INDIVIDUAL (31 variables)	$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$	100%	0.00%

Source: prepared by the author

Both companies were correctly classified by the trained model, whether using consolidated data or individual data. Although the validation set contains only two companies, it proves that after our model learned the characteristics of bankrupted and non-bankrupted companies. It could then apply that knowledge to completely new data.

3.3. Significance of the results

In this research, we aimed to create a model that can accurately classify a company into a bankrupted or a non-bankrupted one. We showed that the use of financial ratios is effective when predicting bankruptcy. This is because neural networks are good at finding patterns in data and mapping the underlying function between a variable X and a variable Y, as the training goes, and the weights and biases are updated. It has also been proved that neural networks can be used with real-world data, making it a reliable tool every practitioner can use.

4. Discussion

4.1. Comparison with other published works

For comparability purpose, we analyzed the performance of two bankruptcy models (the SAF or Simple Analysis of Failure 2002 model and the Altman model) using the data we collected for the present study instead of the original data. Although it would be preferable to re-evaluate the models' parameters using the data of the present study, all the necessary information to perform such reproduction are not available or unclear, making it difficult. Besides, both models were built to be able to predict bankruptcy within a year even when using different data. Therefore, it is appropriate and significant to test the models as it is using new data. However, for a fair comparison, the industries initially used in both models were respected.

4.1.1. Comparison with the SAF2002 model

The SAF2002 model (Shirata, 2003) is a measurement of the risk of bankruptcy designed for Japanese companies. It uses four (4) financial ratios chosen through CART (Classification And Regression Tree) model and later by an Artificial Intelligence algorithm. Only individual data were used. The SAF2002 model demonstrated a discriminant power of 86% or more. The model is represented by the following multiple discriminant analysis formula:

$$SAF2002 = 0.01036\ X_1 + 0.02682\ X_2 - 0.06610\ X_3 - 0.02368\ X_4 + 0.70773$$

Where:

X_1: Retained earnings to Total assets

X_2: Net income before tax to Total assets

X_3: Inventory turnover period

X_4: Interest expenses to Sales

Table 8: SAF2002 performance (with original data)

	SAF2002 model	
	Sample size	Error rate
Bankrupted group	1436	15.22% (type 1 error)
Non-bankrupted group	3434	27.39% (type 2 error)
TOTAL	4870	

Source: Shirata (2003)

The neural network model included all industries; however, since the initial SAF2002 model excluded financial and construction industries, we compared both when using individual data and consolidated data, using the same data as in the neural network model. The results are shown in the following table.

Table 9: SAF2002 model performance results

Individual data				Consolidated data			
All industries n = 290		No financial/construction n = 224		All industries n = 228		No financial/construction n = 180	
Accuracy rate	Type 1 error rate	Accuracy rate	Type 1 error rate	Accuracy rate	Type 1 error rate	Accuracy rate	Type 1 error rate
74.14%	21.92%	74.11%	22.12%	71.49%	27.19%	72.22%	28.89%

Source: prepared by the author

Compared to the original paper, when using different data, the performance of the model has a bit decreased (accuracy rate from 86% to 74.11%), however, it is still a remarkable result.

The accuracy rates for the neural network model are 74.29% for individual data (type 1 error rate = 14.29%), and 81.25% for consolidated data (type 1 error rate = 29.17%). Whether or not financial and construction industries are left out, compared to the SAF 2002 model, the overall accuracy is higher for the neural network model, and the type 1

error rates are lower for individual data and almost the same for consolidated data. So, the neural network has, in overall, a better performance.

However, in terms of practicability and interpretability, the neural network model remains a complex one. The calculation process and the interpretation of the results are still challenging and require an extensive technical knowledge. Also, for a better performance, a huge volume of data is needed when using neural networks.

4.1.2. Comparison with Altman model (Altman, 1968; Altman et al., 1977; Bell et al., 2013)

The Altman model uses five financial ratios to predict bankruptcy. In the original study, 33 bankrupted companies and 33 non-bankrupted companies were used and only manufacturing companies were included. Only individual data were used in the original work, and the period ranges from 1946 to 1965. The model yielded an overall accuracy of 95%. The discriminant equation is as follows:

Z-Score = 0.012 X1 + 0.014 X2 + 0.033 X3 + 0.006 X4 + 0.999 X5

Where:

Z = overall index (Cut-off point = 1.81)

X_1 = Working capital / Total assets

X_2 = Retained earnings / Total assets

X_3 = Earnings before interest and tax / Total assets

X_4 = Market value of equity / Total liabilities

X_5 = Sales / Total assets

Table 10: Altman model performance (with original data)

	Altman model	
	Sample size	Error rate
Bankrupted group	33	6% (type 1 error rate)
Non-bankrupted group	33	3% (type 2 error rate)
TOTAL	66	

Source: Altman (1968)

The performance of Altman model against the data we collected in this study is presented in Table 11. One of the variables to calculate the Z-score requires the market value of equity, however it was not reflected in the consolidated financial statements; therefore, we analyzed the performance of the model using individual data only.

Table 11: Altman model performance results

	Individual data n = 290	
	Accuracy rate	Type 1 error rate
Manufacturing	55.13%	17.95%
Wholesale and Retail	63.33%	53.33%
Others	50.00%	27.17%
TOTAL	**52.74%**	**27.40%**

Source: prepared by the author

The accuracy rate of the Altman model when using different data is very low compared to the original performance (95% against 55.13% if including only manufacturing companies). In fact, the sample size in the original paper was relatively small (66 companies in total), so it is normal to see the performance fall when using a larger sample.

Compared to the neural network model, whether we consider only manufacturing companies, like in the original Altman model, the accuracy rate of the Altman model is very low, while the type 1 error rates are relatively low for manufacturing companies. This

means that the Altman model does not perform well when classifying non-bankrupted companies (there were 28 mistakes out of 39).

4.2. Interpretation of results

The core of the learning process of a neural network model relies on the update of the weight parameters. Through dozens of learning iterations, the model updates itself, so it fits to the data. It lowers or increases the weights after each iteration before reaching the best model, the model with the minimal error.

Since all the connections in the neural network have a weight, the whole network has hundreds of parameters, making it difficult to trace which input variable contributed the most to the prediction of bankruptcy. However, it is possible to determine which had the largest or lowest weight at the beginning, so the model performs well.

Therefore, weights from the input layer to the first hidden layer were pulled out to determine which variables were given how much weights (Table 12 and Table 13).

As different variables were used respectively in the individual dataset and consolidated dataset, different weights were attributed although it may be the same financial ratio. This is because the model was trained on different data.

In the model using consolidated data, the ratio of fixed assets long-term capital was given the largest weight (0.74), following the return on capital employed ratio (0.64) and the quick ratio (0.62), which are used to assess the liquidity and profitability of a company.

In the model using individual data, liquidity ratios were given the largest weights: interest-bearing debt ratio (0.95), financial expenses to sales ratio (0.75), and current ratio (0.70).

Since neural networks operate depending on the input data, it is difficult to determine with certainty why a given variable were attributed a larger or lower weight compared to other variables. The network just determined that these values were the best to solve the problem it was presented to, classifying a company as bankrupted or non-bankrupted.

Nevertheless, we found out that the financial ratios commonly used in conventional financial analysis, which is mostly operated by human beings, are like the ones regarded

by the neural network model. This shows that the neural network model, could indeed learn from relevant variables (here, stability ratios) to assess the financial situation of a company, without external influence from the user, making it a practical tool for managers, external stakeholders, and academics.

Table 12: Input layer weights (Consolidated)

INPUTS		WEIGHTS
X62	Fixed assets to long-term capital ratio [%]	0.74
X20	Return on capital employed [%]	0.64
X59	Quick ratio [%]	0.62
X47	Value-added ratio [%]	0.52
X73	Ordinary profit growth rate [%]	0.49
X67	Interest-bearing debt dependence [%]	0.49
X54	Other capital allocation ratio [%]	0.48
X69	Sales growth [%]	0.48
X19	ROE [%]	0.47
X79	Growth rate of the number of employees [%]	0.44
X71	Gross margin growth rate [%]	0.43
X84	Operating cash flows to current liabilities [%]	0.42
X29	Financial expense to sales ratio [%]	0.42
X63	Debt ratio [%]	0.42
X72	Operating income growth rate [%]	0.42
X64	Equity ratio [%]	0.41
X76	Total assets growth rate [%]	0.37
X70	Sales-growth rate per employee [%]	0.36
X77	Equity growth ratio [%]	0.36
X51	Equipment productivity ratio [%]	0.35
X74	Net profit growth rate [%]	0.35
X80	Value-added growth rate [%]	0.22

Source: prepared by the author

Table 13: Input layer weights (Individual)

INPUTS		WEIGHTS
X67	Interest-bearing debt ratio [%]	0.95
X29	Financial expense to sales ratio [%]	0.75
X58	Current ratio [%]	0.70
X6	Dividend payout ratio [%]	0.53
X20	Return on capital employed [%]	0.52
X69	Sales growth rate [%]	0.46
X7	Dividend yield [%]	0.46
X80	Value-added growth rate [%]	0.45
X63	Debt ratio [%]	0.44
X79	Growth rate of the number of employees [%]	0.43
X71	Gross margin growth rate [%]	0.41
X84	Operating cash flows to current liabilities [%]	0.40
X70	Sales-growth rate per employee [%]	0.39
X53	Public distribution rate [%]	0.38
X27	Sales-cost ratio [%]	0.37
X76	Total assets growth rate [%]	0.37
X18	Operating profit to operating capital ratio [%]	0.33
X19	ROE [%]	0.33
X73	Ordinary profit growth rate [%]	0.33
X81	Sustainability growth rate [%]	0.32
X83	Operating cash flow ratio [%]	0.32
X74	Net profit growth rate [%]	0.31
X72	Operating income growth rate [%]	0.30
X100	EBITDA margin [%]	0.30
X77	Equity growth ratio [%]	0.30
X54	Other capital allocation ratio [%]	0.30
X62	Fixed assets to long-term capital [%]	0.29
X50	Labor distribution rate [%]	0.29

X23	Operating profit on sales [%]	0.29
X52	Added value to capital [%]	0.28
X51	Equipment productivity ratio [%]	0.26

Source: prepared by the author

5. Conclusion

The neural network model proved to be highly reliable and useful in terms of corporate bankruptcy prediction. As opposed to conventional model construction in the academia, where the researcher must manually build the model based on different analyses, by acquiring knowledge from past labeled examples, the neural network trained itself to produce an optimized prediction model.

However, there is still room for improvement for it to be of general application. Compared to other statistical models, the neural network model requires a large quantity of data to perform well. Besides, as there is no general rule, the model also depends on the technical knowledge and the understanding of the domain of the user. And as the neural network model only gives a classification statement, pulling out insights from the results can be difficult.

Nevertheless, despite its complexity, we demonstrated in this research that the neural network model yields higher prediction accuracy than other existing models. A first contribution of this work is the development of an accurate bankruptcy prediction model regardless of the industry, the period, or the size of the company. A second contribution is that it showed the outclassing capacity of neural networks compared to traditional statistical methods.

One of the important parts in developing neural network models is the choice of the parameters and the variables. In this study, we focused on financial ratios as variables as they were proven to be effective predictors by previous literature. Some studies have also used non-financial and qualitative information such as securities reports in their prediction model. To further improve the performance of the neural network model, the simultaneous use of financial and non-financial, quantitative and qualitative information

from reports such as "*kessan tanshin*"[16] reports, auditor's opinion report, and corporate social responsibility reports, needs further investigation.

Complementary future research would be to determine how early we can predict bankruptcy using neural networks allowing companies and their stakeholders to take preventive measures as early as possible.

Another challenge, which is common to researches in the field of neural network and deep learning, is that the calculation process, the learned features, and concepts in the hidden layer are not easily interpretable unlike other machine learning methods. Therefore, it is necessary to define a method for the interpretation of the model's behavior.

And finally, the model presented in this paper was implemented in Python, making it less accessible to those who are not familiar with this programming language or even programming at all. Therefore, to be truly of general application, it is necessary to release the model into a web application or software for practical usage. In that case, the only requirement for the user would be to upload the financial ratios data as input. Hence, this study is significant in the sense that it provides a basic model for developing such a tool.

[16] *Kessan tanshin* are flash reports of financial results of listed companies, that are disclosed to the market and investors in a timely manner. They are provided under the self-regulation of the stock exchange, which is an original aspect of stock exchange in Japan.

Chapter 4. BANKRUPTCY PREDICTION MODEL BASED ON BUSINESS RISK REPORTS: USE OF NATURAL LANGUAGE PROCESSING TECHNIQUES

1. Introduction

Many studies have proposed bankruptcy prediction models using statistical (Altman, 1968; Altman et al., 1977; Ohlson, 1980; Shirata, 2003) or machine learning models (Rasolomanana, 2021; Shin et al., 2005; Shirata, 2019; Tam & Kiang, 1992). Although some researchers attempted to incorporate non-financial information such as industry or number of employees, previous studies have mostly focused on quantitative information, namely financial ratios, to analyze the financial situation of companies. Calderon & Cheh (2002) summarized thoroughly the inputs and outputs used in previous studies. In practice, qualitative information is as much important and insightful for such analysis. Thus, it is necessary to implement a model using qualitative information as input variables (Chung, 2014).

Qualitative information can be internal disclosures, such as managerial reports, and external, such as macroeconomic data from news articles. Previous studies have shown that reports relative to auditor's opinion can be useful when it comes to bankruptcy prediction (Matin et al., 2019). In the case of Japanese companies, audit reports do not include information related to going concern assumption. Managers formulate this assumption, along other risks information, and are reported in the business risks section of the securities report. Therefore, this study presents the hypothesis that the risk of bankruptcy within the next year can be evaluated using information relative to business risk, from training a machine learning model.

For computers to learn features in textual data, it is first necessary to process the texts and extract their features, converting the textual data into their representative numerical values. Natural Language Processing (NLP) is a technology that allows computers to derive insights from human languages. Most NLP applications rely on machine learning methods to mine documents. Support Vector Machines (SVM) have been amongst the most popular techniques (H. Kim et al., 2005; Shin et al., 2005), followed by Naïve Bayes and neural networks (Amani & Fadlalla, 2017). The present study will be focused on the

latter. Neural networks are powerful learning models that have gained in popularity as the quantity of data increases constantly making it more complex to process (Goldberg, 2016). The performance of the neural network model will then be compared to the performances of SVM and Naïve Bayes.

The present research answers the questions: Is there a positive-negative polarity in business risk reports? Is the sentiment in the text extracted from NLP techniques linked to the financial situation, thus, the risk of bankruptcy? Which machine learning method performs best on data from Japanese listed companies?

We propose a bankruptcy prediction model for Japanese listed companies using qualitative information from managerial reports related to business risks. The performance of the model will be evaluated based on how accurately it classifies the texts as bankrupted or non-bankrupted. The objectives of this study are, therefore, to (1) find out if business risk reports are useful in predicting bankruptcy, and (2) to build a machine learning model that can classify texts.

2. Related works

2.1. Studies related to bankruptcy prediction using NLP and machine learning

Financial statements have long been the number one source of information to assess the financial situation of a company. However, companies are required to disclose different reports regularly, which constitutes another source of information. Texts are unstructured data that contain insights. Although some prior works have developed machine learning models using texts as input variables, it is still not has been discussed actively so far, making it underexploited. Human language is very complex and can sometimes be ambiguous even to humans, let alone to computers. NLP is, in fact, a rigorous series of many tough tasks, but it is a powerful tool for interpreting textual data.

Table 14: Literature related to bankruptcy prediction using NLP and machine learning

Article	Data		Text processing method	Results
	Textual data	Origin		
Cerchiello et al. (2017, 2018)	News article	European banks	Semantic vectors	Relative usefulness= 13%
Matin et al. (2019)	Auditors' reports and managements' statements	Denmark	Word embeddings and Convolutional Neural network	AUC and log score • NN_{aud}: 0.844 and 0.1064 • NN_{man}: 0.836 and 0.1078 • $NN_{aud+man}$: 0.843 and 0.1070
Ahmadi et al. (2019)	Annual business reports	Germany	DSCNN	Accuracy • SVM = 0.7679 • CNN = 0.5712 • LTSM = 0.6549 • DSCNN = 0.8414
Mai et al. (2019)	MD&A	US	Word embeddings and Convolutional Neural network	Accuracy and AUC • DL-embedding = 0.568 and 0.784 • DL-CNN = 0.428 and 0.714 • Logistic Regression = 0.434 and 0.717 • SVM = 0.422 and 0.71 • Random Forest = 0.733 and 0.716

Source: prepared by the author

The relative usefulness is a measure of the relative performance gain of the model compared to a perfect model (Sarlin, 2013). If relative usefulness is equal to 1, the model loss is equal to 0, meaning that the model is perfect.

The Area Under the receiver operating characteristics Curve (AUC) is a metric that tells how much the model is capable of distinguishing between classes.

Dependency Sensitive Convolutional Neural network (DSCNN) consists of a convolutional layer built on top of two LSTM networks, because after filtering, the texts are still too long for one-layer- CNN, making it difficult to capture dependencies.

Using the techniques of NLP, some studies (see Table 14) have shown that information from financial news brings insights not found in financial quantitative variables (Cerchiello et al., 2017, 2018). (Matin et al., 2019; Muñoz-Izquierdo et al., 2020) found out that statements from auditors and managements also contribute to the prediction of distress. Other works showed text segments in business management reports, such as Management Discussion and Analysis (MD&A), help detect financial distress (Ahmadi et al., 2019; Mai et al., 2019b).

The main task of sentiment analysis is to identify inner expression in the text (Ragini et al., 2018). In the present study, we demonstrate that texts segments related to the business risks contain decisive information that defines the sentiment of the text. Therefore, two sentiments are considered:

- Negative sentiment, meaning that the risk of bankruptcy is high, and
- Positive sentiment, meaning that bankruptcy is unlikely to happen within the next year.

In Japanese securities report, the risk information section is separated from the Japanese equivalent of the MD&A section, implying that both carry complementary but different information. Also, unlike previous research, sentiment analysis based on TF-IDF (Term Frequency - Inverse Document Frequency) scores is performed in this work. This is because instead of understanding the relationship between the words, we want to capture the lexical features that are descriptive of bankruptcy or non-bankruptcy; and TF-IDF scores measure the importance of each word in the corpus.

2.2. Machine learning techniques in NLP

Many machine learning techniques are used along NLP tools, but the most used in previous studies dealing with bankruptcy prediction are the SVM, Neural networks and Naïve Bayes (Qu et al., 2019).

Support Vector Machines, introduced by Cortes et al. (1995), are widely used supervised machine learning models for binary classification problems. They are easily interpretable and can work well even with a small data set. Its kernel function assigns a hyperplane that best divides a dataset into two classes, by transforming the data into a high-dimensional one. This is particularly useful for nonlinear data sets.

Neural networks and deep learning are powerful tools that learn by mistakes. They have become popular in both academic and practical applications. The classical Neural network is a group of multiple neurons organized in layers. It can learn linear and non-linear functions, making it a proper choice when the relationship between input and output is complex. Another typical model is the Recurrent Neural network (RNN) which is commonly used in stock price prediction since the RNN is suitable for time series analysis, where sequence is key. It is also used in text classification. Another major model in deep learning is the Convolutional Neural network (CNN), generally used for image recognition, as it was initially designed for 3-dimensional data, later successfully experimented on sequential data as well.

Naïve Bayes is also a widely used learning algorithm, it assumes that all predictors are independent. Naïve Bayes classifiers are computationally efficient and easy to implement. It assigns a probability that a given word or text is positive or negative. Naïve Bayes classifiers typically need lots of training examples to perform well.

Previous studies have shown that the performance of a model is not independent of the used data. It relies heavily on the representation of the data (Goodfellow et al., 2016). A model can perform well on a certain dataset but can have a poor performance on a different dataset. Therefore, in the present study, a benchmark of the performance of the three above-mentioned machine learning techniques is carried out.

3. Text processing and analysis

3.1. Data

The data used in this study is the business risks section from securities reports, imported from EOL Database. The total sample of 138 companies from fourteen industries includes 69 bankrupted and 69 non-bankrupted companies equivalent in terms

of asset size, industry, and time frame of collected data. The period ranges from fiscal year 2004 to fiscal year 2017.

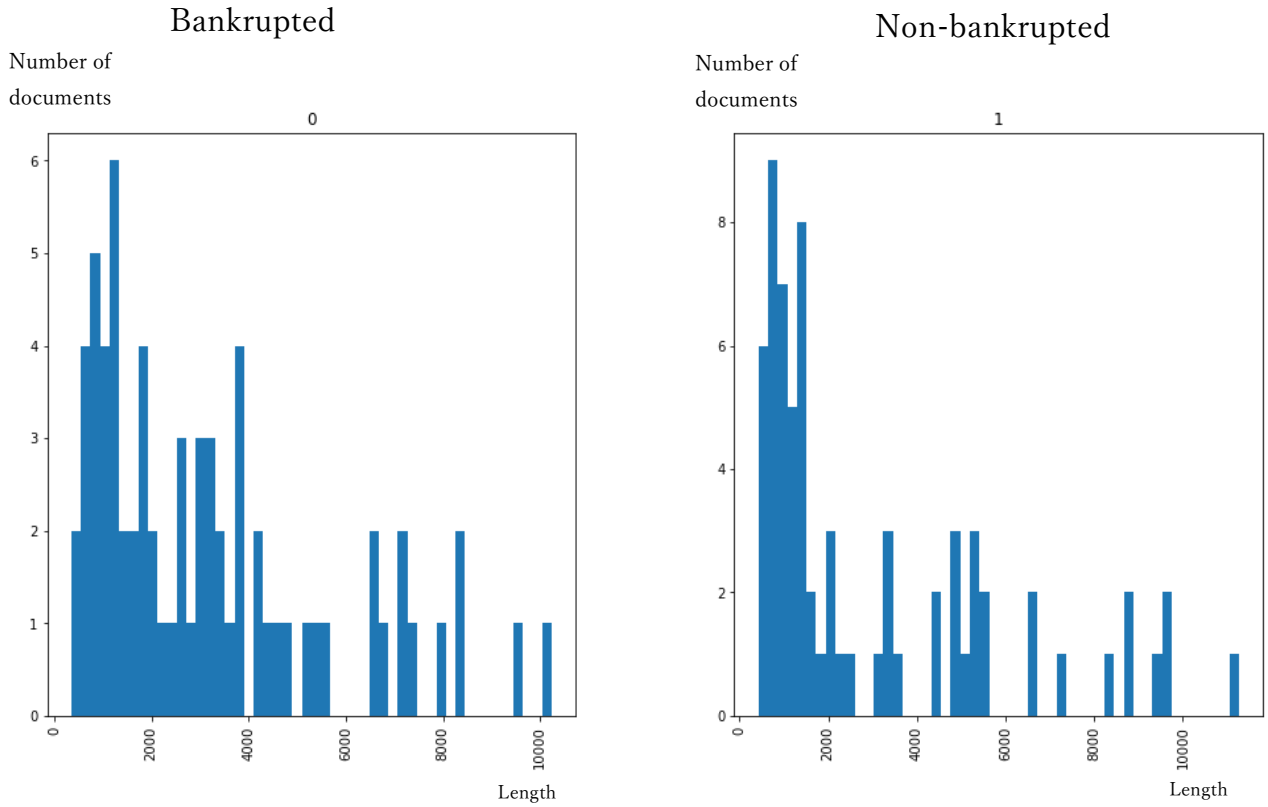

Figure 7: Length of documents in both classes
Source: prepared by the author

3.2. Model construction

Before feeding the data to the model, it is necessary to preprocess the texts first by the means of NLP techniques. The texts are in written in Japanese language. Unlike English or German, the words are not separated with a space. Besides, in Japanese, the characters are not limited to alphabet, numbers, and symbols, in addition to those, there are three kinds of characters in Japanese language: kanji, hiragana and katakana. Therefore, the morphological analysis is more complex. In the present study, we used the pure python

package Janome version 0.4.1 (Uchida, 2020)[17], which is a Japanese morphological analysis engine (also called tokenizer) including the built-in dictionary and the language model. It uses mecab-ipadic-2.7.0-20070801 as the built-in dictionary.

The first processing step is to clean the texts using the Analyzer framework of Janome (Uchida, 2020). To do so, it is necessary to isolate each word or compound word. Each word or compound word constitutes a token. This implies removing characters that do not bear any useful information, such as punctuations, symbols, or other useless characters (numberings, etc.), and stop words. Stop words are languages that are not useful for the analysis. It can be general words from the language itself (such as “the”, “to”, “I”, etc.), or domain-based, which are vocabularies related to accounting, management, or finance (such as “corporation”, “financial statements”, etc.).

The stop words used in the present study was from a programming library called SlothLib, developed by Ohshima et al. (2007) added by domain-based words. Non distinctive words, which are words common to both bankrupted companies’ reports and non-bankrupted companies’ reports were ignored, hence, removed from the corpus.

[17] Janome is an analyzer framework for Japanese language. It is written only in Python, and since we used Python as the programming language in this study, it has the advantage be easily installed, compared to other analyzers.

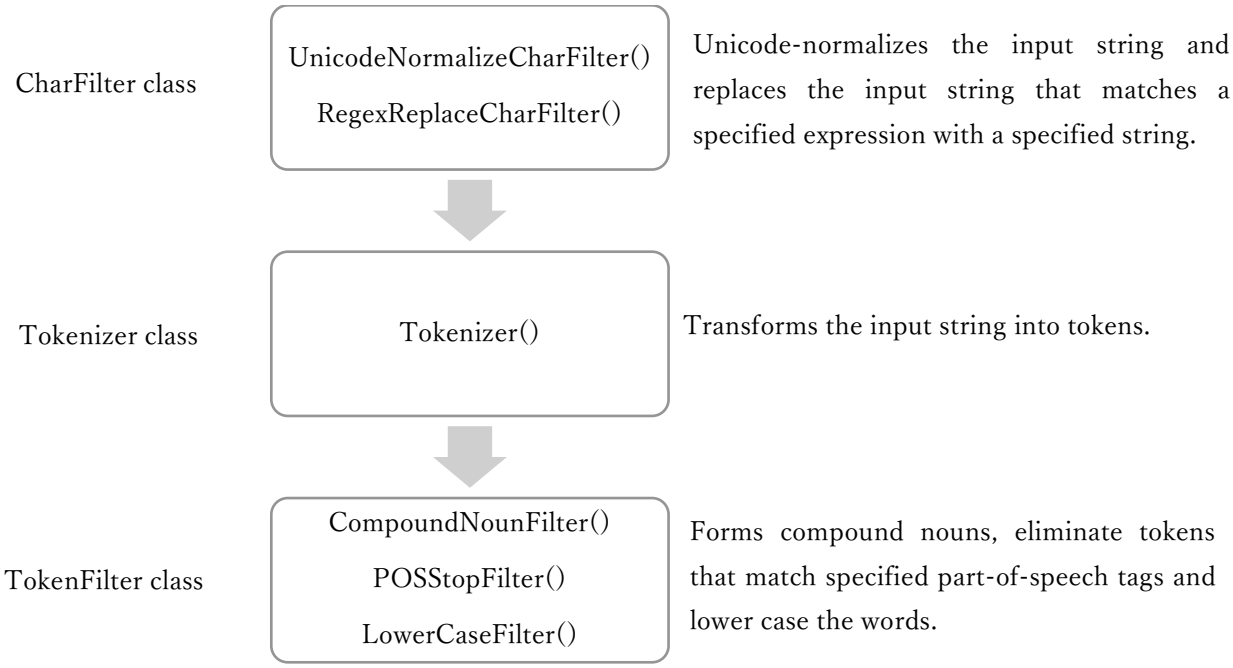

Figure 8: Analyzer framework of Janome

Source: prepared by the author

The next step is to transform the words into numerical values that mathematical models can understand. The values assigned to each word will be its TF-IDF score. TF-IDF evaluates the originality of a word by analyzing how relevant it is to a collection of documents. TF-IDF does not consider the words order or the relationship between words, it is generally used as a lexical feature.

It is calculated by multiplying term frequency and inverse document frequency:

$$w_{i,j} = tf_{i,j} \times log \frac{N}{df_i}$$

$tf_{i,j}$ = Number of occurrences of i in j

df_i = Number of documents containing i

N = Total number of documents

In the present study, the model will be trained and evaluated using three (3) classifiers: neural networks, Support Vector Machines and Naïve Bayes.

The classification was performed through a fully connected neural network, using the Keras library. The parameters of the model were chosen through a trial-and-error process[18]. Hence, the neural network has one (1) hidden layer only.

To minimize the error rate of the model in the prediction, optimization function is used. The optimizer chosen is 'rmsprop', and the loss function corresponds to 'binary_crossentropy'. Loss function is used to estimate the error of the model during the training. Using the error backpropagation (Rumelhart et al., 1988), the weights in particular layer are updated in such manner, that the error rate decreases in following evaluation. We used Binary Cross-Entropy (BCE), which can be defined as:

$$\mathrm{BCE} = -\left(y \log \hat{y} + (1 - y) \log (1 - \hat{y})\right)$$

Where y is the target value (0 for bankrupted and 1 for non-bankrupted), and $\hat{y}$ is the predicted probability of the class to be 1.

Neural network models have many parameters, and overfitting can easily occur. Overfitting can be alleviated to some extent by regularization. Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error (Goodfellow et al., 2016). A recently proposed alternative regularization method is "dropout" (Srivastava et al., 2014). The dropout method is designed to prevent the network from learning to rely on specific weights. It works by randomly dropping a pre-determined percentage of the neurons in the network (or in a specific layer) in each training example. Ignoring, or "dropping-out" of specific neurons can prevent their over-adaptation, which could lead to over-fitting. The dropout technique is effective in NLP applications of neural networks. It is necessary to set the parameter defining the probability of selection of several neurons to drop out from the network. In this study, we applied a dropout of 0.5 rate in the first layer. Along the dropout technique, we used another common regularization technique called early stopping. This means

[18] The choice of the parameters and hyperparameters of the model was done based on the best performance. The model showed the most sensitivity and variation in performance when changing the optimization function.

stopping the training before overfitting occurs but not too early to so that the network has learnt something.

Another way to check if the network generalizes well is to test it. Therefore, a set of the data is hold out for testing (pairs of *(input, label)*) but not used in the training. However, it is also necessary to check how well the network has trained, so it is necessary to reserve a third set of data for cross-validation. This third set will be called validation set. Hence, the data was split into training (n = 92), validation (n = 20) and test (n = 25) sets, with respect to industry proportions. If the number of samples in the industry is enough to be split following a 65/10/25 ratio, then those samples were distributed randomly amongst the three sets. The validation set is also useful to determine when to stop the training before it overfits. As we train the network, the validation set helps determine how well the model is generalizing, and at some point, the validation error will start increasing indicating that the network has started to learn the noise and inaccuracies in the data and the function. And that is when early stopping is used (Marsland, 2009).

The output activation function is the sigmoid function (Leshno et al., 1993). It is mostly used because of its non-linearity and simplicity of the computation. The function is defined as:

$$f(x) = \frac{1}{(1+e^{-x})}$$

The network was compared with SVM and Naïve Bayes. The data was split into training (n = 112) and test (n = 25) sets for both. For the SVM, the kernel function used is the linear kernel. The Naïve Bayes algorithm was the Multinomial Naïve Bayes.

4. Results and discussion

The point of machine learning is that the algorithm must perform well on new previously unsee inputs, not just on the data used for training. This ability is called generalization. The new data corresponds to the testing set, which is why the evaluation of the models is based on the results using this testing set.

There are several metrics to measure the performance of a machine learning model. The choice of the metrics depends on the domain and the data. In the previous literature,

some researchers have used AUC or F1-score, which are useful especially if the dataset is imbalanced. In the present study, since the dataset is balanced (same sample size for bankrupted and non-bankrupted classes) and since in bankruptcy prediction it is more important to spot the risk of bankruptcy that not spot it, the performance of each model was evaluated using accuracy rate (overall percentage of correct classifications), type 1 error rate (percentage of bankrupted companies mistakenly classified by the model as non-bankrupted) and type 2 error rate (non-bankrupted companies mistakenly classified by the model as bankrupted).

The results are summarized in the following table:

Table 15: Summary of results

N = 12 + 13	Confusion matrix	Accuracy	Type 1 error	Type 2 error
Neural networks	$\begin{bmatrix} 10 & 2 \\ 0 & 13 \end{bmatrix}$	92.00%	16.67%	0%
SVM	$\begin{bmatrix} 8 & 4 \\ 0 & 13 \end{bmatrix}$	84.00%	33.33%	0%
Naïve Bayes	$\begin{bmatrix} 7 & 5 \\ 0 & 13 \end{bmatrix}$	80.00%	41.67%	0%

Source: prepared by the author

Previous studies have shown that there is no universal model for all data, the performance of the model is closely related to each data.

In the present case, the neural network had the best performance. A possible cause is related to the quality of the data. The classes (bankrupted class and non-bankrupted class) are balanced as well. As Batista et al. (2004) argued, learning from imbalanced datasets might be difficult. In the present study, the classes are evenly distributed not only in terms of sample size (68 and 69 samples in each class) but also in terms of the data itself since the samples in the non-bankrupted class are the equivalent of the samples in bankrupted class in terms of industry proportion, data period and asset size of the company. Therefore, it is possible that the network could clearly distinguish and learn the characteristics of each class. Moreover, if the model is trained and validated with samples from service industry only, it will not perform well on a testing set with only data from manufacturing industry,

since they do not share the same inherent features. Thus, although the split of the data into three sets is usually done entirely randomly, here, the proportions per industry were controlled so that each set includes samples from each industry.

The choice of the features might also constitute another reason. In text classification, semantic quality and statistical quality are generally the main concerns (Sulo et al., n.d.; Zhang et al., 2011). Semantic quality refers to the ability of the indexing method to analyze the relationship between words. Statistical quality refers to the ability to classify the term in the domain it belongs to. TF-IDF measures the rarity of each word in a document; and as the objective is to retrieve the terms that contain bankruptcy-related sentiment, TF-IDF helps determine lexical features for bankrupted and non-bankrupted classes.

5. Conclusion

Every year, companies disclose large amounts of textual data, which requires time and effort to process for the human being, hence, the need to involve intelligent models to perform such task efficiently. The present study introduced the use of machine learning and NLP techniques using unstructured qualitative data (business risk information). It has been shown that the business risk section in securities reports carries predictive information on imminent risk of bankruptcy, and that machine learning, especially neural networks, can effectively analyze the sentiment in the text.

Our findings also showed that TF-IDF, although not as sophisticated as features extracted through word embeddings or LTSM (Long-Term Short Memory), represent an efficient basic metric to extract the most descriptive terms of bankruptcy. This study also compared different classifiers, showing that neural networks are better classifier in this context: business risk reports as data and TF-IDF scores as features. However, neural networks remain computationally expensive and not as transparent and straightforward to explain as SVM or Naïve Bayes.

This study presents other limitations, which prompt future investigations. First, the size of the sample is limited, making it less likely to be representative of all Japanese listed companies. Moreover, models such as neural networks and Naïve Bayes usually perform better the more examples it learns. Second, quantitative, or qualitative information only is

not sufficient when evaluating the financial situation of company in practice. Therefore, future work should combine both in the same model, and our study constitutes a useful prior step in building such comprehensive model.

Chapter 5. ENSEMBLE NEURAL NETWORK USING A SMALL DATASET FOR THE PREDICTION OF BANKRUPTCY: COMBINING NUMERICAL AND TEXTUAL DATA

1. Introduction

This research focuses on the development of a bankruptcy prediction model using quantitative and qualitative data simultaneously. Being able to predict bankruptcy accurately will allow managers, investors, and other stakeholders to spot the risk in advance, based not only on quantitative information, but also on qualitative information, and to take the necessary measures. Quantitative data here are represented by numerical values, while qualitative data are extracted from textual sources.

In bankruptcy prediction, although previous studies were primarily focused on quantitative data, namely financial ratios, several studies have also investigated the addition of qualitative data to their model (Boratyńska & Grzegorzewska, 2018; Park & Han, 2002; Slowinski & Zopounidis, 1995; Zakharova, 2013; Zopounidis et al., 1992). They found out that quantitative and qualitative data may be based on different assumptions but are complementary. Therefore, using both in the same model yields better prediction results than using only one of the two types.

However, these models were trained using a huge volume data in the development process. In practice, though, especially when it comes bankruptcy, the data may be limited naturally (for instance, the number of companies which has gone bankrupt cannot be controlled) or expensive in terms of time and/or money to get. Therefore, there is a need to develop a bankruptcy prediction model with mixed data (quantitative and qualitative) using a small dataset. In this study, we are using data from Japanese listed companies, out of which only a few (about 200 companies) have filed for bankruptcy up until 2019, which is relatively small as a dataset. Here, a small dataset means that the number of examples used in the training is far smaller than the number of predictors.

In this study, we develop a neural network model using different types of data: numerical and non-numerical data. Before feeding the data to the model, non-numerical features, such as texts, are transformed beforehand. However, when the data are different

in nature (categorical or continuous, sequential, or non-sequential, etc.), a plain concatenation may cause discrepancies, and hence, hurt the neural network model.

Some studies have investigated the use of neural networks with small datasets, especially in the field of medicine to identify diseases. There is no exact threshold to determine if a dataset is small, but Pasini (2015) has investigated on 451 cases, and D'souza et al. (2020) experimented on the MNIST dataset[19] comparing samples of size 100, 500, and 1,000.

Pasini (2015) has successfully developed a neural network tool for small datasets analysis using the leave-one-out procedure. The leave-one-out procedure is a cross-validation method commonly used in machine learning. This method sets the elements in the validation dataset randomly, to improve the robustness of the model, and the author emphasized the effectiveness of such procedure with neural networks trained with only few samples.

D'souza et al. (2020) found out that optimization task is more important when the data is scarce. The authors developed a convolutional neural network using a two-step method to find the optimal neural network structure. The first step consists of finding and listing all potential network structures, and the second step consists of finding the optimal layer dimension by picking the best performing models and permuting combinations of layer dimension.

However, both studies require a huge computing power. In fact, the leave-out-procedure can be seen as an extreme version of the classic k-fold cross-validation, where the value of k corresponds to the number of examples in the dataset. Even with a small dataset, the computation behind the cross-validation would take a long time. Similarly, the method proposed by D'souza et al. (2020) may end up with a large list of potential neural network structures and training each model may take a long time without additional optimization or with a limited computing power.

---

[19] MINST dataset is an open database of images of handwritten digits, containing 60.000 training examples and 10.000 testing examples.

Therefore, in the present study, we introduce another method using an ensemble neural network with 2 models: one with numerical data, and another with textual data. The purpose is then to investigate how well the ensemble neural network model can perform with a small dataset to predict bankruptcy, when using numerical and textual data. Thus, this paper attempts (1) to compare the performance of a neural network using combined data (numerical and textual data into a single data frame) with an ensemble of two individual networks with different data types, (2) to determine what combination of numerical and textual data gives the best bankruptcy prediction results.

2. Methodology

This section will outline the setup and steps towards the construction of the ensemble neural network. An ensemble model combines several "weak" models with relatively low performance into a single stronger model with a high predictive performance. In the model presented in this manuscript, we combined two independent neural networks, one using numerical data and one using textual data.

2.1. Input data

In this study, we used both numerical and textual data. The numerical data is composed by a set of 22 financial ratios, which are the mainly used variables in literature and in practice, when it comes to analyzing the financial situation of companies or predicting their failure (Altman, 1968; Liang et al., 2016; Ohlson, 1980; Shimerda & Kung, 2012).

The values of the financial ratios were normalized between 0 and 1, because despite being expressed in percentages, the scales are very different (maximum value= 12029.10%, minimum value= -4889.69%). Normalization ensures that the model will not attribute higher weights to high value input, as it may hurt the learning of the model (Bhanja & Das, 2018).

As for the textual data, we extracted segments of text from Japanese listed companies' securities reports and "*kessan tanshin*" reports, that directly or inherently refer to the business profitability and/or competitive advantage of the company, which are the two most common and important qualitative factors of bankruptcy according to literature

(Boratyńska & Grzegorzewska, 2018; M. J. Kim & Han, 2003; Park & Han, 2002; Slowinski & Zopounidis, 1995; Zakharova, 2013).

The sections that refer to the business profitability and/or competitive advantage of the company are the following:

- Risks factors (from securities report)
- Performance overview (from securities report)
- Issues to be addressed (from securities report)
- Performance prediction (from "*kessan tanshin*" report)

Each segment of text was preprocessed so all stop words or meaningless characters are removed, then transformed into numerical values using Term Frequency – Inverse Document Frequency (TF-IDF) scores.

The total sample of 137 companies used in this study comprises 68 cases for the bankrupted class and 69 cases for the non-bankrupted class. The financial ratios and the reports of each company were downloaded from the EOL database[20], ranging from Fiscal Year 2004 to Fiscal Year 2018, and no industry were excluded.

2.2. Neural network development

We developed 3 types of neural networks models: individual model (using a single type of data), combined model (using numerical and text features combined into a single data frame), and ensemble model (2 individual networks using different types of data).

After the data has been preprocessed, it is split into 3 sets: training set, validation set and test set. As a rule of thumb, when a huge volume data is available, the data is split randomly according to a certain distribution. But in the case of a small dataset, the same method of splitting would create a great imbalance amongst the 3 sets, as our dataset is comprised of companies from different subclasses (different industries and data periods). Therefore, it is necessary to preserve a certain level of control when splitting the data by preserving a minimum of balance amongst the sets. Ultimately, the elements in each set

[20] EOL is a database that provides financial and non-financial information on Japanese listed companies.

were selected randomly, however, we made sure that each set had elements from each subclass.

A total of 15 different models were built, each model being a combination of financial ratios and the segments of texts (see Table 16), to find out which texts are the most descriptive of a nearly coming bankruptcy.

Table 16: Input information in each model

Model	Numerical data	Textual data			
	Financial ratios	Risks factors	Overview of the performance	Issues to be addressed	Performance prediction
A	○	○			
B	○	○	○		
C	○	○		○	
D	○	○			○
E	○	○	○	○	
F	○	○	○		○
G	○	○		○	○
H	○	○	○	○	○
I	○		○		
J	○		○	○	
K	○		○		○
L	○		○	○	○
M	○			○	
N	○			○	○
O	○				○

Source: prepared by the author

Each neural network model is then trained and cross-validated. We closely monitored the training loss and the validation loss to ensure that the model was learning appropriately. If the loss does not converge, the settings of the network are iteratively changed until the optimal settings are found (until the loss is minimized). Moreover, to avoid overfitting, we applied some regularization techniques to the models.

For the individual models, the training set is directly fed to the network. For the combined model, after each data has been preprocessed, they are first concatenated into a single data frame before the training. For the ensemble model, instead of concatenating the inputs into a single data frame, two individual neural networks were built. The output from each network is then weighted proportionally to its performance to calculate the output of the ensemble model (See Figure 9).

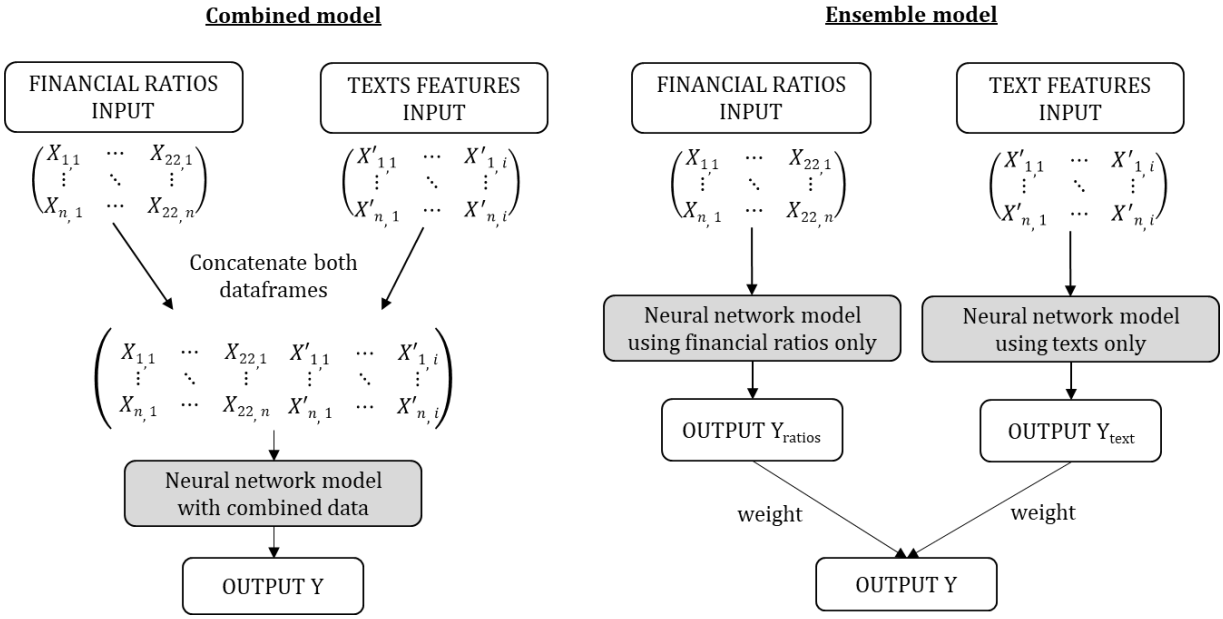

Figure 9: Schematic representation of the Combined model and Ensemble model used in this paper

Source: prepared by the author

The weighted average of the individual inputs is calculated as follows:

$$weighted\ average = \frac{\sum_{i=1}^{n}(x_i \times w_i)}{\sum_{i=1}^{n} w_i}$$

Where

x = financial ratio or text predictions

w = weight factor (between 0 and 1)

3. Results

After training, the performance of the models is evaluated using the test set. The accuracy rates of the individual model, combined model and ensemble model are summarized in Figure 10.

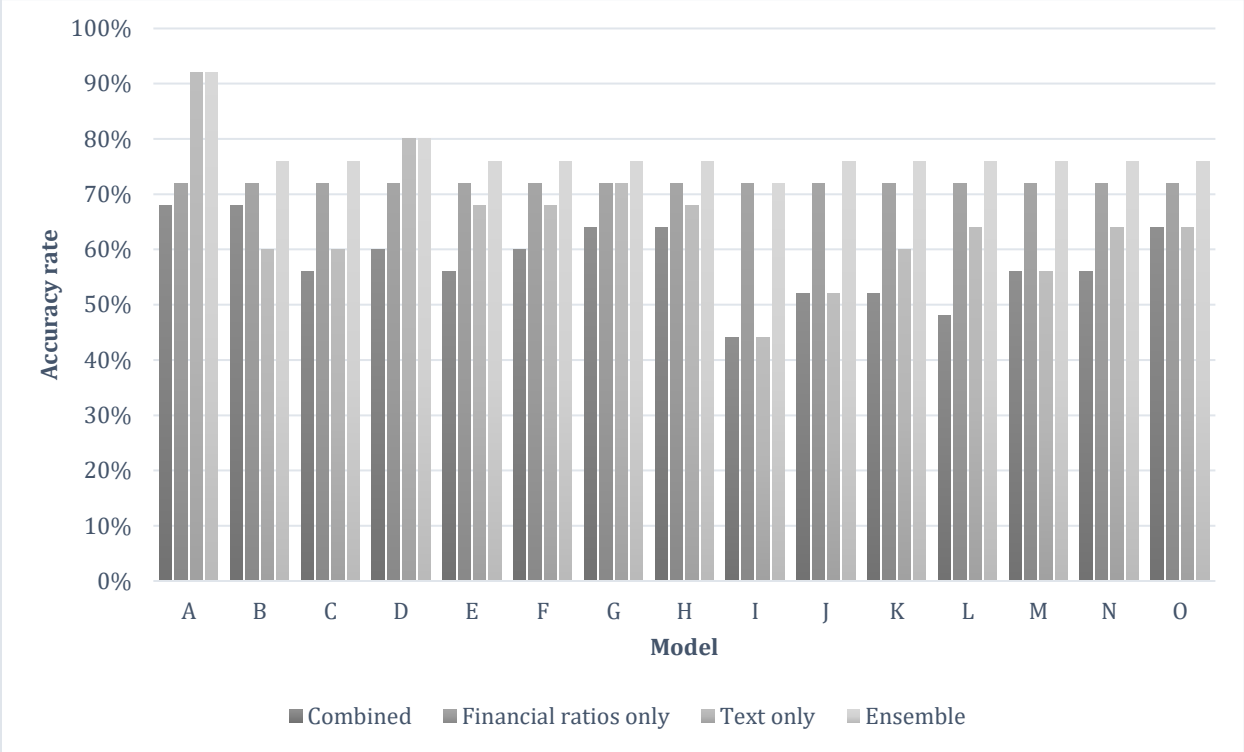

Figure 10: Graph showing the performance of the different models: combined model, individual models and ensemble model

Source: prepared by the author

The ensemble model showed the highest accuracy for all the models (model A to model O). The model A (financial ratios and risk factors) showed the best performance with an accuracy rate of 92%.

We compared the performance of the ensemble with other machine learning models The results are summarized in Table 17.

Table 17: Performance comparison with other machine learning models

Model input	NN_{ens}	RDC	TREE	SVM	KNN	LOG	NB
A	92%	88%	64%	64%	80%	64%	56%
B	76%	84%	72%	64%	80%	64%	52%
C	76%	80%	64%	64%	80%	64%	56%
D	80%	84%	64%	64%	84%	64%	56%
E	76%	80%	72%	64%	80%	64%	52%
F	76%	84%	72%	64%	80%	64%	56%
G	76%	84%	76%	64%	76%	64%	52%
H	76%	84%	72%	64%	80%	64%	52%
I	72%	76%	72%	60%	80%	64%	56%
J	76%	80%	68%	60%	80%	64%	52%
K	76%	80%	68%	60%	80%	64%	56%
L	76%	84%	72%	68%	80%	64%	52%
M	76%	80%	64%	64%	80%	64%	60%
N	76%	84%	68%	64%	80%	64%	52%
O	76%	84%	64%	64%	80%	68%	64%

NN_{ens}: Ensemble Neural network, RDC: Random Forest, TREE: Decision Tree, SVM: Support Vector Machine, KNN: K-Nearest Neighbor, LOG: Logistic Regression, NB: Naive Bayes

Source: prepared by the author

In overall, the Random Forest demonstrated the best performance. And model A showed the highest accuracy rate (88%) here too. K-Nearest Neighbor model also showed about 80% of accuracy amongst all models, however, the results are too stable regardless of the input information, making it less reliable than the Random Forest. The structure of the data indeed has an important impact on the results when using K-Nearest Neighbor, meaning that it is sensible to the dimensionality of the data. Therefore, as the dimension increases, more training examples is required. The remaining machine learning models

are even more sensible to this matter, as they produced poorer results (52 to 72% accuracy).

4. Discussion

Combining quantitative data and qualitative data did not give better prediction results than using only one type of data. However, when creating individual networks for each type of data, then averaging their respective results in an ensemble network, the performance was better than when using only one type of data. These confirm the fact that plain combination of different data creates discrepancies within the data, making the learning difficult for the model. The results varied depending on the input information, and the model with financial ratios and risk factors (model A) showed the best result. Thus, we can deduce that amongst all the texts, the information contained in the section about risk factors is the most valuable when it comes to bankruptcy prediction.

The results also show that the performance of some other machine learning model was better than the ensemble neural network. Amongst all the models, Random Forest model yielded the best accuracy rates. This suggests that, although our ensemble neural network can be used with small datasets, it is still too complex when fed with a small size of samples. What is interesting is that Random Forest is also an ensemble learning method which uses decision trees. This is consistent with Dietterich (2000) stating that when the amount of available training data is too small, ensemble learning can help find a good approximation and improve prediction accuracy by averaging the outputs of the individual models. In fact, those constitute the benefits of using ensemble models. Since the base models are trained separately and do not have to be individually highly performant, this method requires much less computation power and model tuning.

Another possible reason why Random Forest showed high levels of accuracy is the fact that we used only two base networks for the ensemble, while the Random Forest model used dozens of trees.

5. Conclusion

In a context where the available data is scarce, and predictive variables are far more numerous, the parameter tuning becomes too complex, and the neural network ends up with a poor learning. In this paper, we demonstrate the viability of using an ensemble neural network with different data – numerical data in one network and textual data in another – in such context.

This study also confirms that quantitative and qualitative data are indeed complementary, and ensemble learning can bring out that complementarity, by giving a weighted average of the individual models, and yielding a higher level of prediction accuracy.

One limitation to this study is the need of a lot of computation power for Grid Searching all the parameters, so for efficiency reasons, it was done using an iterative process (trial-and-error). Therefore, although the current settings of the model have showed acceptable results, it may not be the optimal ones.

As for further research, we intend to explore other ensemble learning approaches with neural network by using bagging, boosting, stacking methods, and integrating more than 2 networks.

Chapter 6. BAGGING AND BOOSTING ALGORITHMS WITH NEURAL NETWORKS: USE OF SMALL SIZED DATASET WITH MIXED TYPES OF VARIABLES

1. Introduction

In the previous chapter, we have introduced the use of the ensemble concept to improve the performance of the neural network model when using a small sized dataset with mixed types of variables. The ensemble comprised two neural networks with respectively financial ratios and texts as data and yielded satisfying results.

This begs the question: can we improve the performance further using well known ensemble methods? In fact, as described in Chapter 2, bagging ensemble and boosting ensemble are the main types of ensemble learning. Those algorithms utilize multiple base models, generally more than just two. Some studies have investigated the use of bagging or boosting ensemble in bankruptcy prediction. For instance, Nanni & Lumini (2009)investigated Australian credit data (690 examples), German credit data (1000 examples) and Japanese credit data (690 examples) and found out that ensemble method outperforms standalone models in bankruptcy prediction and credit scoring. M. J. Kim & Kang (2010)conducted a series of experiments on 1458 externally audited Korean manufacturing firms and showed that bagging and boosting with neural networks performed better than a standalone neural network. S. Y. Kim & Upneja (2014) applied Adaboost with decision trees, a major method of boosting, to successfully predict financial distress of restaurants. Zięba et al. (2016) introduced eXtreme Gradient Boosting (XGB) with decision trees, another major method of boosting, and showed that boosting performed better than other techniques. They conducted studies on a dataset of Polish companies from 2007 to 2013 (for the bankrupt class) and from 2000 to 2012 for (for the non-bankrupt class) with a total sample of 5910 to 10503.

Apart from the Australian dataset and Japanese dataset in Nanni & Lumini (2009), those studies have used large datasets (more than 1000 samples) and using only numerical data. Therefore, in the present study, we investigate the effectiveness of ensemble methods based on neural networks, with a small sized dataset with mixed types of variables.

Our objectives are to (1) compare the performance of the ensemble when the data is combined into a single data frame and when it is not, and (2) compare the performance of bagging and boosting method on our dataset.

2. Model construction

The data used is the same as in Chapter 5. We will integrate financial ratios and texts related to business risk in our models.

Two bagged models, one with financial ratios and textual data combined before it is fed to the model, will be constructed (see Figure 11). And similarly for the boosted models (see Figure 12), to evaluate the impact of the data structure on ensembles. For the boosting model, we used the Adaboost algorithm consisting of adjusting the weights of the data points that have been misclassified in the previous model iteratively to achieve a high level of accuracy for the overall model.

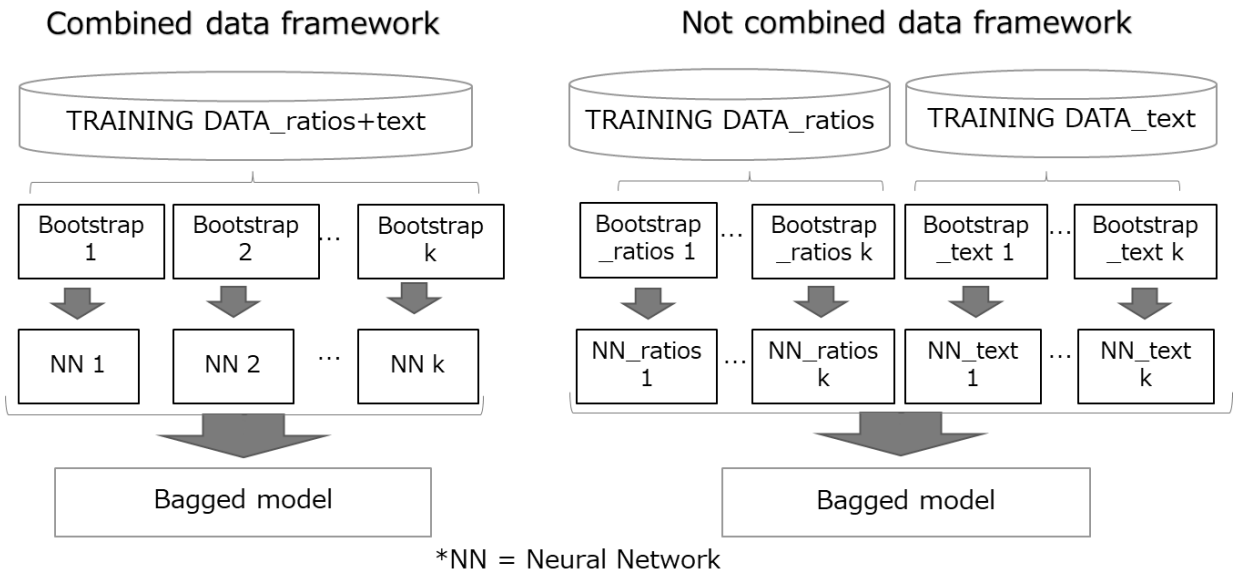

Figure 11: Bagging ensemble models' frameworks
Source: prepared by the author

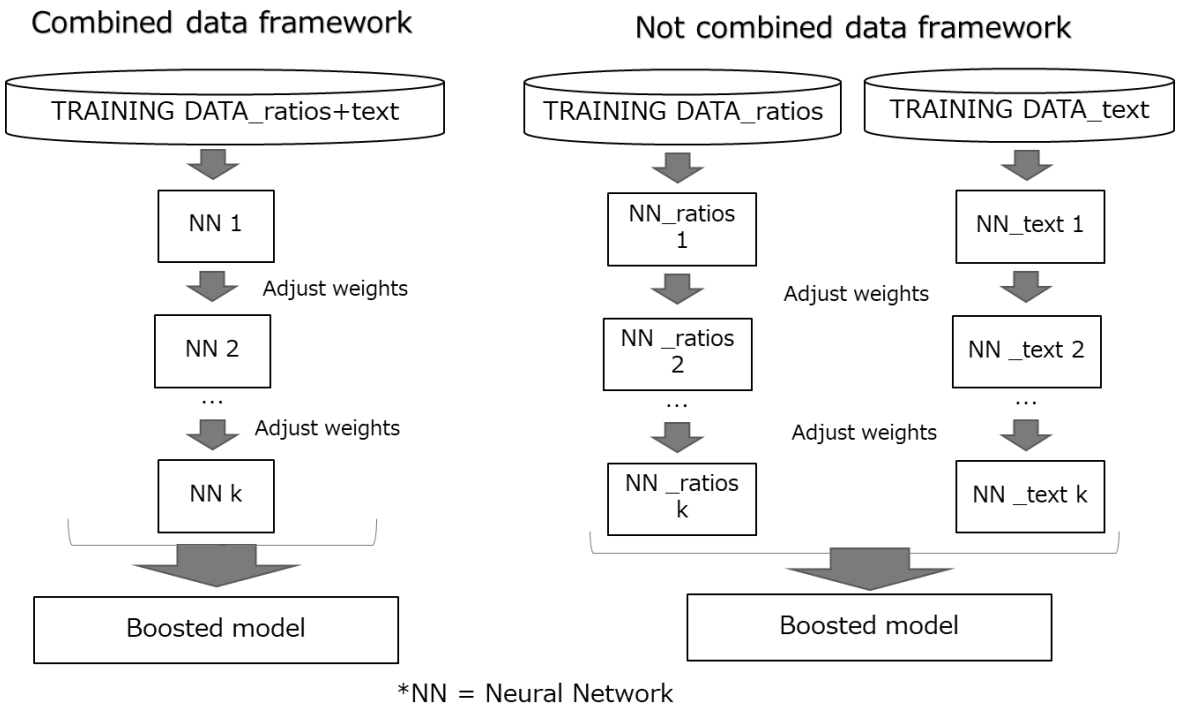

Figure 12: Boosting ensemble models' frameworks

Source: prepared by the author

3. Results and discussion

Each model is evaluated based on different number of classifiers (neural networks). Table 18 and Tables 19 summarize the accuracy of each type of model.

For the bagging method, although the highest accuracy was 85% with 10 neural networks, the model stabilizes at 80% accuracy when the number of classifiers is superior or equal to 20. Therefore, it would be correct to conclude that the best accuracy is 80%.

For the boosting method, in both frameworks, the best accuracy achieved is 100%. This stabilizes after using 20 neural networks when the training data is combined, and 15 neural networks when the data is not combined.

Table 18: Bagging models accuracy

Number of classifiers	Combined data	Not combined data
5	60%	75%
10	64%	85%
15	64%	75%
20	68%	80%
25	52%	80%
50	56%	80%
100	56%	80%

Source: prepared by the author

Table 19: Boosting models accuracy

Number of classifiers	Combined data	Not combined data
5	100%	100%
10	100%	95%
15	95%	100%
20	100%	100%
25	100%	100%
50	100%	100%
100	100%	100%

Source: prepared by the author

The best bagging and boosting ensembles, respectively 80% and 100% of accuracy, are far better than a standalone neural network with only 68% accuracy. The results are summarized in Table 20.

Table 20: Overall comparison of results

Model	Accuracy rate
Standalone model with both data combined	68%
Ensemble with 2 tuned classifiers without combining	92%
Best bagging ensemble with both data combined	68%
Best bagging ensemble without combining	80%
Best boosting ensemble with both data combined	100%
Best boosting ensemble without combining	100%
Random Forest	88%

Source: prepared by the author

4. Conclusion

When using small sized dataset with mixed types of variables, we showed that ensembles are an effective method. Compared to a standalone neural network, it is more robust and more stable.

With bagging method, the performance was higher when each type of data is kept separated, which suggests that the structure of the data frame maybe important in bagging, while with boosting method, the structure of the data frame does not affect the performance.

Another advantage of using ensembles is that we did not have to perform any heavy parameter tuning, as ensembles are based on poor models. This is particularly important when the hardware is limited because finding optimal parameters often requires high computing power.

However, a limitation of the use of ensembles is obviously the fact that the training is longer since there are more models involved. Another limitation is that if neural networks had little interpretability, using more than one makes it worse.

Chapter 7. CONCLUSION

1. Thesis claim restatement

This manuscript evaluates a new perspective for applying machine learning models on bankruptcy prediction tasks and our proposed ensemble neural network models can be seen as the components of effective bankruptcy prediction approaches. Not only the proposed models can overcome the limitation given by the availability of large training data, but it can also improve the framework for the applications of neural network models in bankruptcy prediction when using both numerical and textual data.

2. Significance of this study

The findings in this manuscript suggest that ensemble classifiers can outperform a single tuned classifier. Moreover, this work demonstrated that ensembles can robustly and accurately predict near-future bankruptcy automatically with neural networks.

Based on these findings, we recommend investigating the use of such methods when the training data is scarce and composed of mixed types of variables, which have been overlooked in the literature that has mainly focused on a single type of data when building a model and/or has assumed the availability of massive data in any context.

Although we live in the age of big data, there are still some domains (for instance, bankruptcy prediction or some fields in medicine), where it is difficult to access large datasets or good quality data.

Not only from a conceptual aspect, but also from a practical aspect, reconciling quantitative and qualitative data is important, especially when evaluating the financial situation of a company.

3. Limitations and further research

While the limited computing resources constraints the optimization of the models, these findings provide new insight into the usability of ensemble methods to improve prediction power.

Also, although this research improves our understanding of how to build a machine learning model that can predict bankruptcy with a high-level accuracy, they do not exhaust the other ensemble methods. In this study, we investigated two ensemble approaches: bagging and boosting; other extensions of those algorithms could further improve our models. Moreover, a comparison with a third ensemble method called stacking could also be interesting, as it uses heterogenous classifiers as base models. Combination of different classifiers such as neural network, decision trees, or even ensemble models like random forest can be investigated.

Another interesting extension would be the use of other types of neural networks and investigate how powerful they can be on a small and mixed types of data. Some studies have been conducted regarding the use of convolutional neural networks (CNN) in bankruptcy prediction. In the attempt to reconcile financial ratios and textual data from financial reports, Mai et al. (2019) used a CNN to process the textual data and an artificial neural network after concatenating the processed texts with the financial ratios. Akita, R. (2016) explored the use of recurrent neural networks (RNN) for the concatenated data frame, but using news article as textual data, and stock prices as numerical data. However, both studies employed large datasets to train the neural network: 11,827 U.S. public companies for (Mai et al., 2019a) and time-series data over eight years of ten Nikkei companies in Akita, R. (2016) 's article.

Moreover, to extend furthermore the implications of the results of this thesis, we must also investigate the use of such model with other data that may influence the risk of bankruptcy. Additional quantitative data would be macroeconomic indicators or financial experts' analysis results such as "*kaisha shikiho*[21]" reports and additional qualitative data could include audio recordings. For instance, we could use past interviews of managers or representants of them making public statements about the company's activities and its performance and evaluate if it has an impact on the prediction performance.

[21] "Kaisha Shikiho" is a Japanese quarterly magazine that provides a summary of a company's special features, business performance, financial content, and stock price movements.

Finally, investigating the trend of financial ratios amongst the companies could also reveal some new perspectives about the factors influencing bankruptcy prediction, and from there extract the ratios that show some pattern as variables.

To conclude, this research aimed to help understand how machine learning models can be useful in practice in the context of bankruptcy prediction, so that managers and other shareholders can spot the risk in advance and take necessary measures. Although the present thesis presents clear limitations, we believe it is an essential step towards the conception of a comprehensive bankruptcy prediction model that considers real-world conditions.

REFERENCES

Ahmadi, Z., Martens, P., Koch, C., Gottron, T., & Kramer, S. (2019). Towards bankruptcy prediction: Deep sentiment mining to detect financial distress from business management reports. *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*, 293–302. https://doi.org/10.1109/DSAA.2018.00040

Alaka, H. A., Oyedele, L. O., Owolabi, H. A., Kumar, V., Ajayi, S. O., Akinade, O. O., & Bilal, M. (2018). Systematic review of bankruptcy prediction models: Towards a framework for tool selection. *Expert Systems with Applications*, *94*, 164–184. https://doi.org/10.1016/J.ESWA.2017.10.040

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Source: The Journal of Finance*, *23*(4), 589–609.

Altman, E. I., Haldeman, R. G., & Narayanan, P. (1977). ZETA Tin* ANALYSIS A new model to identify bankruptcy risk of corporations. In *Journal of Banking and Finance* (Vol. 1).

Amani, F. A., & Fadlalla, A. M. (2017). Data mining applications in accounting: A review of the literature and organizing framework. *International Journal of Accounting Information Systems*, *24*, 32–58. https://doi.org/10.1016/j.accinf.2016.12.004

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20–29. https://doi.org/10.1145/1007730.1007735

Beaver, W. H. (1966). Financial Ratios As Predictors of Failure Authors (s): William H . Beaver Source : Journal of Accounting Research , Vol . 4 , Empirical Research in Accounting : Selected Published by : Wiley on behalf of Accounting Research Center , Booth School of Busi. *Journal of Accounting Research*, *4*(1966), 71–111. http://www.jstor.org/stable/2490171

Bellovary, J. L., Giacomino, D. E., & Akers, M. D. (2007a). Financial Education Association A Review of Bankruptcy Prediction Studies: 1930 to Present A Review of Bankruptcy Prediction Studies: 1930 to Present. In *Source: Journal of Financial Education* (Vol. 33).

Bellovary, J. L., Giacomino, D. E., & Akers, M. D. (2007b). Financial Education Association A Review of Bankruptcy Prediction Studies: 1930 to Present A Review of Bankruptcy Prediction Studies: 1930 to Present. In *Source: Journal of Financial Education* (Vol. 33).

Bhanja, S., & Das, A. (2018). *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting*. 5–10.

Blum, M. (2019). *Failing Company Discriminant Analysis Author (s): Marc Blum Published by : Wiley on behalf of Accounting Research Center , Booth School of Business , University of Chicago Stable URL : https://www.jstor.org/stable/2490525*. *12*(1), 1–25.

Boratyńska, K., & Grzegorzewska, E. (2018). Bankruptcy prediction in the agribusiness sector: Lessons from quantitative and qualitative approaches. *Journal of Business Research*, *89*(February), 175–181. https://doi.org/10.1016/j.jbusres.2018.01.028

Calderon, T. G., & Cheh, J. J. (2002). A roadmap for future neural networks research in auditing and risk assessment. *International Journal of Accounting Information Systems*, *3*(4), 203–236. https://doi.org/10.1016/S1467-0895(02)00068-4

Cerchiello, P., Nicola, G., Ronnqvist, S., & Sarlin, P. (2017). *Deep learning bank distress from news and numerical financial data*.

Cerchiello, P., Nicola, G., Rönnqvist, S., & Sarlin, P. (2018). Deep Learning for Assessing Banks' Distress from News and Numerical Financial Data. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3292485

Chitoshi, K. (2014). *Introduction to corporate financial analysis*. Chikura publishing.

Chung, W. (2014). BizPro: Extracting and categorizing business intelligence factors from textual news articles. *International Journal of Information Management*, *34*(2), 272–284. https://doi.org/10.1016/j.ijinfomgt.2014.01.001

Cortes, C., Vapnik, V., & Saitta, L. (1995). Support-Vector Networks Editor. In *Machine Leaming* (Vol. 20). Kluwer Academic Publishers.

Deakin, E. B. (1972). A Discriminant Analysis of Predictors of Business Failure. *Journal of Accounting Research*, *10*(1), 167. https://doi.org/10.2307/2490225

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1857 LNCS*, 1–15. https://doi.org/10.1007/3-540-45014-9_1

D'souza, R. N., Huang, P. Y., & Yeh, F. C. (2020). Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size. *Scientific Reports*, *10*(1), 1–13. https://doi.org/10.1038/s41598-020-57866-2

Edmister, R. O. (1972). *Financial Ratios as Discriminant Predictors of Small Business Failure Author (s): Robert O . Edmister Source : The Journal of Finance , Mar ., 1972 , Vol . 27 , No . 1 (Mar ., 1972), pp . 139-140 Published by : Wiley for the American Finance Associati*. *27*(1), 139–140.

Fan, J., & Li, R. (2006). *Statistical Challenges with High Dimen-sionality: Feature Selection in Knowledge Discovery*.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, *57*, 345–420.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. In *MIT press*. MIT press. {http://www.deeplearningbook.org

Hurbans, R. (2020). *Artificial Intelligence Algorithms*.

Kim, H., Howland, P., & Park, H. (2005). Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research*, *6*, 37–53. https://doi.org/10.1021/bi702018v

Kim, M. J., & Han, I. (2003). The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Systems with Applications*, *25*(4), 637–646. https://doi.org/10.1016/S0957-4174(03)00102-7

Kim, M. J., & Kang, D. K. (2010). Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications*, *37*(4), 3373–3379. https://doi.org/10.1016/J.ESWA.2009.10.012

Kim, S. Y., & Upneja, A. (2014). Predicting restaurant financial distress using decision tree and AdaBoosted decision tree models. *Economic Modelling*, *36*, 354–362. https://doi.org/10.1016/J.ECONMOD.2013.10.005

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, *6*(6), 861–867. https://doi.org/10.1016/S0893-6080(05)80131-5

Liang, D., Lu, C. C., Tsai, C. F., & Shih, G. A. (2016). Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, *252*(2), 561–572. https://doi.org/10.1016/j.ejor.2016.01.012

Mai, F., Tian, S., Lee, C., & Ma, L. (2019a). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, *274*(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Mai, F., Tian, S., Lee, C., & Ma, L. (2019b). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, *274*(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Marsland, S. (2009). *Machine Learning – An Algorithmic Perspective*.

Matin, R., Hansen, C., Hansen, C., & Mølgaard, P. (2019). Predicting distresses using deep learning of text segments in annual reports. *Expert Systems with Applications*, *132*, 199–208. https://doi.org/10.1016/j.eswa.2019.04.071

Min, S. H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Systems with Applications*, *31*(3), 652–660. https://doi.org/10.1016/j.eswa.2005.09.070

Muñoz-Izquierdo, N., Laitinen, E. K., Camacho-Miñano, M. del M., & Pascual-Ezama, D. (2020). Does audit report information improve financial distress prediction over Altman's traditional Z-Score model? *Journal of International Financial Management and Accounting*, *31*(1), 65–97. https://doi.org/10.1111/jifm.12110

Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *36*(2), 3028–3033. https://doi.org/10.1016/J.ESWA.2008.01.018

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, *18*(1), 109. https://doi.org/10.2307/2490395

Ohshima, H., Nakamura, S., & Tanaka, K. (2007). *SlothLib: A Programming Library for Researches on the Web*.

Park, C.-S., & Han, I. (2002). A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction. *Expert Systems with Applications*, *23*(3), 255–264. https://doi.org/10.1016/S0957-4174(02)00045-3

Pasini, A. (2015). Artificial neural networks for small dataset analysis. *Journal of Thoracic Disease*, *7*(5), 953–960. https://doi.org/10.3978/j.issn.2072-1439.2015.04.61

Qu, Y., Quan, P., Lei, M., & Shi, Y. (2019). Review of bankruptcy prediction using machine learning and deep learning techniques. *Procedia Computer Science*, *162*(Itqm 2019), 895–899. https://doi.org/10.1016/j.procs.2019.12.065

Ragini, J. R., Anand, P. M. R., & Bhaskar, V. (2018). Big data analytics for disaster response and recovery through sentiment analysis. *International Journal of Information Management*, *42*, 13–24. https://doi.org/10.1016/j.ijinfomgt.2018.05.004

Rasolomanana, O. (2021). Corporate Bankruptcy Prediction Using Neural Networks: Case of Japanese companies. *Nenpō Zaimu Kanri Kenkyū (Japan Financial Management Association)*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning Representations by Back-Propagating Errors. *MIT Press: Cambridge, MA, USA*, 696–699.

Sarlin, P. (2013). *On policymakers ' loss functions and the evaluation of early warning systems* (Issue 15).

Shimerda, K. H. C. and T. A., & Kung. (2012). An of Empirical Analysis Useful Financial Ratios. *Financial Management Association İnternational*, *10*(1), 51–60.

Shin, K. S., Lee, T. S., & Kim, H. J. (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, *28*(1), 127–135. https://doi.org/10.1016/j.eswa.2004.08.009

Shirata, C. Y. (2003). *Predictors of Bankruptcy after Bubble Economy in Japan: What can you learn from Japan case?* https://www.researchgate.net/publication/281090087

Shirata, C. Y. (2019). *Bankruptcy prediction model and Corporate rating using AI technology*. Zeimukeiri Kyokai CO., LTD.

Slowinski, R., & Zopounidis, C. (1995). Application of the Rough Set Approach to Evaluation of Bankruptcy Risk. *Intelligent Systems in Accounting, Finance and Management*, *4*(1), 27–41. https://doi.org/10.1002/j.1099-1174.1995.tb00078.x

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

Sulo, $, María, J., & Hidalgo, G. (n.d.). *Text Representation for Automatic Text Categorization*.

Tam, K., & Kiang, M. (1992). Managerial applications of neural networks: the case of bank failure prediction. *Management Science*, *38*(7), 926–947.

Tian, S., & Yu, Y. (2017). Financial ratios and bankruptcy predictions: An international evidence. *International Review of Economics and Finance*, *51*(June), 510–526. https://doi.org/10.1016/j.iref.2017.07.025

Uchida, T. (2020). *Janome version 0.4.1 documentation*. https://mocobeta.github.io/janome/en/

Zakharova, A. A. (2013). Fuzzy swot analysis for selection of bankruptcy risk factors. *Applied Mechanics and Materials*, *379*, 207–213. https://doi.org/10.4028/www.scientific.net/AMM.379.207

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), 2758–2765. https://doi.org/10.1016/j.eswa.2010.08.066

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, *58*, 93–101. https://doi.org/10.1016/J.ESWA.2016.04.001

Zopounidis, C., Pouliezos, A., & Yannacopoulos, D. (1992). Designing a DSS for the assessment of company performance and viability. *Computer Science in Economics and Management*, *5*(1), 41–56. https://doi.org/10.1007/BF00435281

APPENDICES

1. List of companies (bankrupted group)

No	Securities code	Company name	Industry	Bankruptcy date
1	1351	Hoko Co.	Fisheries and Agriculture	25/4/2002
2	1804	Sato Kogyo Co., Ltd.	Construction	3/3/2002
3	1818	Rinkai Nissan Construction Co., Ltd.	Construction	30/3/2002
4	1817	KATSUMURA CONSTRUCTION CO.LTD.	Construction	29/9/2005
5	1889	ＡＯＭＩ　ＣＯＮＳＴＲＵＣＴＩＯＮ　ＣＯ．、ＬＴＤ.	Construction	19/2/2009
6	1839	MAGARA CONSTRUCTION CO.,	Construction	5/7/2008
7	1838	Ｋｏｋｕｎｅ　Ｃｏｒｐ.	Construction	15/11/2002
8	1825	ＥＣＯ－ＴＥＣＨ　ＣＯＮＳＴＲＵＣＴＩＯＮ　ＣＯ．，ＬＴＤ.	Construction	14/4/2004
9	1829	Ｄａｉｗａ　Ｃｏｎｓｔｒｕｃｔｉｏｎ	Construction	2/5/2003
10	1845	MORIMOTO CORPORATION	Construction	1/10/2003
11	1851	OHKI CORPORATION	Construction	30/3/2004
12	1874	SATOHIDE CORPORATION	Construction	10/6/2004
13	1857	MATSUMURA-GUMI CORPORATION	Construction	5/5/2005
14	1858	InoueKogyoCo.,Ltd.	Construction	16/10/2008
15	8839	NICHIMO CORP.	Real estate	13/2/2009
16	1917	Ｎｉｓｓｅｋｉ　Ｈｏｕｓｅ　Ｉｎｄ.	Real estate	30/10/2002
17	1872	AZELCORPORATION	Real estate	30/3/2009
18	1913	ＡＳＡＨＩ　ＨＯＭＥＳ　ＣＯ．、ＬＴＤ.	Real estate	24/4/2009
19	1880	SURUGA	Construction	24/6/2008
20	1908	SAMPEI CONSTRUCTION CO.,LTD.	Construction	24/7/2008
21	1902	ＹＡＭＡＺＡＫＩ　ＣＯＮＳＴＲＵＣＴＩＯＮ　ＣＯ．、ＬＴＤ.	Construction	30/10/2008
22	1830	Ｋｙｏｅｉ　Ｃｏ．，Ｌｔｄ.	Machinery	19/5/2003
23	1800	Tone Geo Tech Co., Ltd.	Construction	19/5/2005
24	1797	Fujiki Komuten	Construction	4/6/2002
25	1790	HEIWA OKUDA CO.,LTD.	Construction	30/1/2009
26	1779	MATSUMOTO KENKO CO.,	Construction	15/12/2008
27	1772	Tohoku Enterprise Co.,Ltd.	Machinery	31/5/2004
28	1754	TOSHIN HOUSING CO.,	Real estate	9/1/2009
29	1744	KYOEI SANGYO CO., LTD.	Real estate	18/7/2008
30	1725	Fujita Corporation	Construction	29/9/2005

31	2205	ＳＵＲＵＧＡＹＡ　ＣＯＭＰＡＮＹ　ＬＩＭＩＴＥＤ	Food	17/1/2014
32	2219	Ｔａｋａｒａｂｕｎｅ　Ｃｏｒｐ.	Food	24/1/2003
33	2228	CYBELE Co.,	Food	17/01/2019
34	2808	SANBISHI　CO.LTD.	Food	28/10/2005
35	2920	KB	Food	17/1/2002
36	3007	Ｋｏｂｅ　Ｋｉｉｔｏ	Textile	20/2/2003
37	3115	ＴＷＲ　Ｈｏｌｄｉｎｇｓ　Ｃｏ.，ｌｔｄ	Textile	10/7/2002
38	3207	CHUO	Real estate	24/4/2009
39	3206	Ｎａｎｋａｉ　Ｗｏｒｓｔｅｄ　Ｓｐｉｎｎｉｎｇ	Textile	28/3/2003
40	3304	TOSCO CO., LTD.	Textile	30/5/2008
41	3584	ＦＵＫＵＳＵＫＥ ＣＯＲＰＯＲＡＴＩＯＮ	Retail	21/6/2003
42	7910	Ｄａｎｔａｎｉ　Ｃｏｒｐ.	Construction	10/4/2002
43	3870	NIPPON KAKOH SEISHI CO.,LTD	Others (Manufacturing)	29/5/2002
44	4079	Ｈａｋｕｓｕｉ　Ｔｅｃｈ	Pharmaceutics	21/6/2002
45	5143	Ｓｅｃａｉｃｈｏ　Ｃｏｒｐ.	Others (Manufacturing)	31/7/2003
46	1786	Oriental Shiraishi	Construction	26/11/2008
47	5925	Ｓａｋａｉ　Ｉｒｏｎ　Ｗｏｒｋｓ	Others (Manufacturing)	9/9/2003
48	5917	ＳＡＫＵＲＡＤＡ　ＣＯ.、ＬＴＤ.	Others (Financial institutions)	27/11/2012
49	5926	AG AJIKAWA CORPORATION	Others (Manufacturing)	21/3/2004
50	6106	Ｈｉｔａｃｈｉ　Ｓｅｉｋｉ	Machinery	19/8/2002
51	6114	SUMIKURA INDUSTRIAL CO., LTD	Machinery	10/5/2002
52	6216	KOTOBUKI INDUSTRY CO.LTD.	Construction	30/10/2002
53	6394	Ｏｙｅ　Ｋｏｇｙｏ	Others (Manufacturing)	29/5/2003
54	6359	Awamura Manufacturing Co.Ltd.	Machinery	14/8/2004
55	6453	SILVER SEIKO	Machinery	27/12/2011
56	6275	Ｉｓｅｋｉ　Ｐｏｌｙ-ｔｅｃｈ	Machinery	27/3/2002
57	6290	S.E.S.CO.,LTD.	Others (Manufacturing)	16/1/2009
58	6660	ＩＮＮＥＸＴ　ＣＯ.、Ｌｔｄ	Machinery	9/9/2011
59	6813	Ｎａｋａｍｉｃｈｉ　Ｃｏｒｐ.	Others (Manufacturing)	19/2/2002
60	6909	Hokubu Communication & Industrial Co., Ltd.	Services	29/4/2002
61	6868	TOKYO CATHODE LABORATORY CO.,	Machinery	14/03/2013
62	6887	LSI Card Corporation	Wholesale	11/7/2005
63	6671	ARM ELECTRONICS CO.,	Machinery	23/8/2010
64	6665	Ｅｌｐｉｄａ　Ｍｅｍｏｒｙ、Ｉｎｃ.	Others (Manufacturing)	27/2/2012
65	6263	Ｐｒｏｄｕｃｅ　Ｃｏ.、Ｌｔｄ.	Machinery	26/9/2008
66	2473	Genesis Technology Inc.	Services	25/9/2008

67	7286	Ｉｚｕｍｉ　Ｉｎｄｕｓｔｒｉｅｓ	Machinery	28/2/2002
68	7312	Ｔａｋａｔａ　Ｃｏｒｐｏｒａｔｉｏｎ	Machinery	1/7/2016
69	6872	COLIN CORPORATION	Pharmaceutics	14/7/2003
70	6667	ＳＨＩＣＯＨ　Ｃｏ.、ＬＴＤ.	Machinery	10/8/2012
71	7884	HONMA GOLF CO., LTD.	Retail	20/6/2005
72	7881	NISSO INDUSTRY CO. LTD.	Others (Manufacturing)	26/7/2004
73	8024	ＳＩＬＶＥＲ　ＯＸ　Ｉｎｃ.	Textile	1/9/2009
74	8094	Ｎａｋａｍｉｃｈｉ　Ｍａｃｈｉｎｅｒｙ　Ｃｏ.、Ｌｔｄ.	Machinery	5/2/2009
75	9963	ＥＭＯＲＩ　ＧＲＯＵＰ　ＨＯＬＤＩＮＧＳ　ＣＯ.，ＬＴＤ.	Wholesale	1/5/2015
76	7449	TAIYOKOGYOCO.,LTD.	Wholesale	8/12/2008
77	7589	Ｄｉｇｉｃｕｂｅ　Ｃｏ.，Ｌｔｄ.	Services	26/11/2003
78	8172	Dai-Ichi Katei Denki Co., Ltd.	Retail	16/4/2002
79	8177	Sogo Denki	Retail	12/2/2002
80	8189	Ｍａｔｓｕｙａｄｅｎｋｉ　Ｃｏ.	Retail	25/9/2003
81	8169	Ｔａｋａｒａｂｕｎｅ　Ｃｏ.	Retail	14/1/2003
82	9917	GEO REACLE CO.,LTD.	Retail	9/7/2003
83	9925	Kitanokazoku Co. Ltd.	Services	17/1/2002
84	9958	Ｓａｒｉ　Ｃｏｒｐ.	Retail	30/9/2003
85	7424	Ｎｉｋｏ　Ｎｉｋｏ　Ｄｏ	Retail	9/4/2002
86	7529	Ｈａｒｕｙａｍａ　Ｃｈａｉｎ	Textile	27/9/2002
87	7580	Ｆｏｏｄｓｎｅｔ　Ｃｏｒｐ.	Services	31/10/2002
88	2758	y-Arriba CORPORATION	Services	15/6/2005
89	8564	ＴＡＫＥＦＵＪＩ　ＣＯＲＰＯＲＡＴＩＯＮ	Others (Financial institutions)	28/9/2010
90	8597	ＳＦＣＧ　ＣＯ.、ＬＴＤ.	Others (Financial institutions)	23/2/2009
91	8580	First Credit Corporation	Others (Financial institutions)	7/3/2002
92	8577	LOPRO	Others (Financial institutions)	2/11/2009
93	8571	NIS GROUP CO.,	Others (Financial institutions)	9/5/2012
94	8489	CREDIT ORGANIZATION OF SMALL AND MEDIUM-SIZED ENTERPRISES	Others (Financial institutions)	25/2/2011
95	1920	SHOKUSAN JUTAKU CO., LTD.	Real estate	13/1/2002
96	8837	Ｈｏｋｋａｉｄｏ　Ｓｈｉｎｋｏ　Ｃｏ.，Ｌｔｄ.	Real estate	9/4/2003
97	8845	Ｃｅｓａｒ　Ｃｏ.	Real estate	24/3/2003
98	8858	DIA KENSETSU Co., LTD.	Construction	19/12/2008

99	8866	Commercial-re.co.,Ltd	Real estate	6/5/2010
100	8868	UrbanCorporation	Real estate	13/8/2008
101	8874	ＪＯＩＮＴ　ＣＯＲＰＯＲＡＴＩＯＮ	Real estate	29/5/2009
102	8878	THEJAPANGENERALESTATECO.,LTD.	Real estate	5/2/2009
103	8882	ZEPHYR CO., LTD.	Real estate	18/7/2008
104	8884	ＤＩＸ　ＫＵＲＯＫＩ　ＣＯ．、ＬＴＤ.	Real estate	14/11/2008
105	8900	ＳＥＩ　ＣＲＥＳＴ　ＣＯ．、ＬＴＤ.	Real estate	2/5/2011
106	8901	ＤＹＮＡＣＩＴＹ　Ｃｏｒｐｏｒａｔｉｏｎ	Real estate	31/10/2008
107	8910	SUNCITY CO.,LTD	Real estate	26/9/2011
108	8911	SOHKEN HOMES Co.,Ltd.	Real estate	26/8/2008
109	8921	C' s Create Co.,Ltd	Real estate	26/9/2008
110	8937	Human21 Corp.	Real estate	19/9/2008
111	8939	ＤＡＩＷＡＳＹＳＴＥＭ　ＣＯ.,ＬＴＤ.	Real estate	1/10/2010
112	8943	Ｓ－ＧＲＡＮＴ. ＣＯ. 、ＬＴＤ.	Real estate	12/3/2009
113	8947	Noel Co.,Ltd.	Real estate	30/10/2008
114	8991	LIFE STAGE CO.,	Real estate	30/4/2009
115	3236	ＰＲＯＰＥＲＳＴ　ＣＯ. 、ＬＴＤ.	Real estate	14/5/2010
116	3247	Ｌ－ＣＲＥＡＴＥ　Ｃｏ. 、Ｌｔｄ.	Real estate	2/10/2008
117	9132	ＤＡＩＩＣＨＩ　ＣＨＵＯ　ＫＩＳＥＮ　ＫＡＩＳＨＡ	Shipping	29/09/2015
118	9204	Skymark Airlines Inc.	Airline	28/01/2015
119	9205	Japan Airlines	Airline	19/1/2010
120	9309	Keishin Warehouse Co., Ltd.	Warehouse / Transportation related business	11/4/2002
121	9374	Tiger Kanri,Inc.	Shipping	12/4/2011
122	9378	ＷＯＲＬＤ・ＬＯＧＩ　Ｃｏ.、　Ｌｔｄ.	Warehouse / Transportation related business	30/08/2013
123	9703	GENERAS CORPORATION	Real estate	26/4/2004
124	9673	Koshien Tochi Kigyo Co., Ltd.	Real estate	15/3/2002
125	2328	ARISAKA. CO. LTD.	Services	27/5/2008
126	9712	TransDigitalCo.,LTD.	Services	1/9/2008
127	9609	Ｃ＆Ｉ Ｈｏｌｄｉｎｇｓ Ｃｏ.,Ｌｔｄ.	Services	27/2/2012
128	8569	UNICO CORPORATION	Others (Financial institutions)	25/10/2006
129	9786	CATSINC.	Services	23/2/2004
130	4655	ＮＯＶＡ　ＣＯＲＰＯＲＡＴＩＯＮ	Services	26/10/2007
131	4702	55 Station Inc.	Retail	11/4/2005
132	7633	NESTAGE	Wholesale	12/8/2010
133	8888	ＣＲＥＥＤ　ＣＯＲＰＯＲＡＴＩＯＮ	Real estate	9/1/2009

134	4835	Index	Information and Telecommunications	27/06/2013
135	4328	MOVIE TELEVISION INC.	Information and Telecommunications	1/3/2004
136	8902	Ｐａｃｉｆｉｃ Ｈｏｌｄｉｎｇｓ、 Ｉｎｃ.	Real estate	10/3/2009
137	2731	NIWS Co. HQ Ltd.	Services	30/4/2008
138	4357	La Parler Co.,	Services	5/10/2010
139	2318	Crest Investments	Services	31/7/2012
140	2356	ＴＣＢ Ｈｏｌｄｉｎｇｓ Ｃｏｒｐｏｒａｔｉｏｎ	Services	20/10/2010
141	2403	Link One	Services	28/4/2011
142	8936	ｒｅ－ｐｌｕｓ ｉｎｃ.	Real estate	24/9/2008
143	3379	Ｆｕｊｉ Ｂｉｏｍｅｄｉｘ Ｃｏ.、 Ｌｔｄ.	Retail	14/10/2008
144	8941	REICOF CO.,LTD.	Real estate	20/3/2008
145	2460	ＡＰＲＥＣＩＯ ＣＯ.、ＬＴＤ.	Services	5/6/2009
146	1606	Ｊａｐａｎ Ｄｒｉｌｌｉｎｇ Ｃｏ.、Ｌｔｄ.	Steel	22/6/2018

2. List of the initial 46 financial ratios

INDIVIDUAL		CONSOLIDATED	
X 6	Dividend payout ratio	X 6	Dividend payout ratio
X 7	Dividend Yield	X 7	Dividend Yield
X 17	ROA	X 17	ROA
X 18	operating profit to operating capital	X 18	operating profit to operating capital
X 19	ROE	X 19	ROE
X 20	Return on capital employed	X 20	Return on capital employed
X 22	Gross profit margin	X 22	Gross profit margin
X 23	Operating income to net sales	X 23	Operating income to net sales
X 24	Operating profit margin	X 24	Operating profit margin
X 25	Ordinary income to net sales	X 25	Ordinary income to net sales
X 26	Net income to net sales	X 26	Net income to net sales
X 27	Cost of sales ratio	X 27	Cost of sales ratio
X 28	Selling and administrative expenses to sales ratio	X 28	Selling and administrative expenses to sales ratio
X 29	Finance cost to sales ratio	X 29	Finance cost to sales ratio
X 47	Value Added Ratio	X 47	Value Added Ratio
X 50	Labor sharing ratio	X 50	Labor sharing ratio
X 51	Facilities Distribution Ratio	X 51	Facilities Distribution Ratio
X 52	Actual capital distribution ratio	X 52	Actual capital distribution ratio
X 53	Public Distribution Rate	X 53	Public Distribution Rate
X 54	Share of other capital	X 54	Share of other capital
X 55	Shareholder Distribution Ratio	X 55	Shareholder Distribution Ratio

X 58	Current ratio	X 58	Current ratio
X 59	Quick ratio	X 59	Quick ratio
X 61	Fixed ratio	X 61	Fixed ratio
X 62	Fixed long-term compliance rate	X 62	Fixed long-term compliance rate
X 63	Debt ratio	X 63	Debt ratio
X 64	Capital adequacy ratio	X 64	Capital adequacy ratio
X 66	Interest-bearing debt ratio	X 66	Interest-bearing debt ratio
X 67	Dependence on interest-bearing debt	X 67	Dependence on interest-bearing debt
X 69	Sales growth rate	X 69	Sales growth rate
X 70	Sales per employee growth rate	X 70	Sales per employee growth rate
X 71	Gross profit growth rate	X 71	Gross profit growth rate
X 72	Operating income growth rate	X 72	Operating income growth rate
X 73	Ordinary income growth rate	X 73	Ordinary income growth rate
X 74	Net income growth rate	X 74	Net income growth rate
X 75	Earnings per share growth rate	X 75	Earnings per share growth rate
X 76	Total capital (total assets) growth rate	X 76	Total capital (total assets) growth rate
X 77	Capital adequacy ratio	X 77	Capital adequacy ratio
X 78	Growth in net assets per share.	X 78	Growth in net assets per share.
X 79	Percentage increase in the number of employees	X 79	Percentage increase in the number of employees
X 80	Value-added growth rate	X 80	Value-added growth rate
X 81	Sustainable Growth Rate	X 81	Sustainable Growth Rate
X 82	Dividend per share growth rate	X 82	Dividend per share growth rate
X 83	Operating cash flow ratio (cash flow margin)	X 83	Operating cash flow ratio (cash flow margin)
X 84	Operating cash flow to current liabilities ratio	X 84	Operating cash flow to current liabilities ratio
X 100	EBITDA margin	X 98	EBITDA margin

3. Source code

3.1. Chapter 3

```
# IMPORT LIBRARIES AND PACKAGES
import pandas as pd
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
import matplotlib.pyplot as plt

# IMPORT DATASETS
traindata = pd.read_csv('…')
X_train = traindata.drop('Y', axis=1)
y_train = traindata['Y']

testdata = pd.read_csv('…')
# print('TEST DATASET \n:',
testdata.describe().transpose())
# print('TEST DATASET \n:', testdata)
X_test = testdata.drop('Y', axis=1)
y_test = testdata['Y']

# DATA PREPROCESSING
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# IMPLEMENT NEURAL NETWORK
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# BINARY CLASSIFICATION
n_inputs = X_train.shape[1]

model = Sequential([
    Dense(n_inputs, input_shape=(n_inputs, ),
activation='relu'),
    Dense(31, activation='relu'),
    Dense(1, activation='sigmoid')
])


# COMPILE THE NN
model.compile(optimizer='adam',
loss='binary_crossentropy')
model.summary()

from tensorflow.keras.callbacks import
EarlyStopping
early_stop = EarlyStopping(monitor='val_loss',
mode='min', verbose=1, patience=25)

# FIT THE NN TO THE TRAINING SET
model.fit(X_train, y_train, batch_size=40,
validation_size=0.2,
          epochs=1000, verbose=2,
callbacks=[early_stop])

# PLOT THE LOSS
losses = pd.DataFrame(model.history.history)
losses.plot()

# PREDICTION (O if bankrupt and 1 if healthy)
import numpy as np
y_pred = model.predict(X_test)

pred = []
for predictions in y_pred:
    if predictions < 0.50:
        predictions = 0
    else:
        predictions=1
    pred.append(predictions)

y_pred = np.array(pred)
y_test = np.array(y_test)

# SAVE WEIGHTS
weights = model.get_weights()
# for layer in model.layers:
#     weights = layer.get_weights()

import sys
sys.stdout = open('indiv_weights.txt', 'w')
print("\n\nweights:\n", weights)
```

3.2. Chapter 4

```
from janome.tokenizer import Tokenizer
from janome.analyzer import Analyzer
from janome.charfilter import *
from janome.tokenfilter import *
import nltk
import re

#%%
#Bankrupt_files
bankrupted_risk_reports = os.listdir('…')
bankrupted_risk_reports = [file for file in
bankrupted_risk_reports]

bankrupted_file = []
for report in range(0,
len(bankrupted_risk_reports)):
    with open(f"…", encoding="utf8") as file:
        filesB = file.read()
        bankrupted_file.append(filesB)

df_bankrupted = pd.DataFrame(bankrupted_file)
df_bankrupted.columns = ['text_file']
df_bankrupted['Y'] = 0 #Y=0 means bankrupted

#%%
#Nonbankrupt_files
nonbankrupted_risk_reports = os.listdir(…')
nonbankrupted_risk_reports = [file for file in
nonbankrupted_risk_reports]

nonbankrupted_file = []
for report in range(0,
len(nonbankrupted_risk_reports)):
    with open(f"…") as file:
        filesNB = file.read()
        nonbankrupted_file.append(filesNB)

df_nonbankrupted =
pd.DataFrame(nonbankrupted_file)
df_nonbankrupted.columns = ['text_file']
df_nonbankrupted['Y'] = 1 #Y=1 means non-
bankrupted
```

```
#Bankrupt_files
bankrupted_risk_reports_val = os.listdir('…)
bankrupted_risk_reports_val = [file for file in
bankrupted_risk_reports_val]

bankrupted_file_val = []
for report in range(0,
len(bankrupted_risk_reports_val)):
    with open(f"…", encoding="utf8") as file:
        filesB_val = file.read()
        bankrupted_file_val.append(filesB_val)

df_bankrupted_val =
pd.DataFrame(bankrupted_file_val)
df_bankrupted_val.columns = ['text_file']
df_bankrupted_val['Y'] = 0 #Y=0 means bankrupted

#Nonbankrupt_files
nonbankrupted_risk_reports_val = os.listdir('…')
nonbankrupted_risk_reports_val = [file for file
in nonbankrupted_risk_reports_val]

nonbankrupted_file_val = []
for report in range(0,
len(nonbankrupted_risk_reports_val)):
    with open(f"…", encoding="utf8") as file:
        filesNB_val = file.read()

nonbankrupted_file_val.append(filesNB_val)

df_nonbankrupted_val =
pd.DataFrame(nonbankrupted_file_val)
df_nonbankrupted_val.columns = ['text_file']
df_nonbankrupted_val['Y'] = 1 #Y=1 means non-
bankrupted

concatenated_val = [df_bankrupted_val,
df_nonbankrupted_val]
df_val = pd.concat(concatenated_val)
# print(df_val)

#Bankrupt_files (test data)
testbankrupted_risk_reports = os.listdir('…')

testbankrupted_file = []
for report in range(0,
len(testbankrupted_risk_reports)):
    with open(f"…", encoding="utf8") as file:
        testfilesB = file.read()
        testbankrupted_file.append(testfilesB)

df_bankrupted_test =
pd.DataFrame(testbankrupted_file)
df_bankrupted_test.columns = ['text_file']
df_bankrupted_test['Y'] = 0

#Nonbankrupt_files (test data)
testnonbankrupted_risk_reports = os.listdir('…')

testnonbankrupted_file = []
for report in range(0,
len(testnonbankrupted_risk_reports)):
    with open(f"…", encoding="utf8") as file:
        testfilesNB = file.read()

testnonbankrupted_file.append(testfilesNB)

df_nonbankrupted_test =
pd.DataFrame(testnonbankrupted_file)
df_nonbankrupted_test.columns = ['text_file']
df_nonbankrupted_test['Y'] = 1

concatenated_test = [df_bankrupted_test,
df_nonbankrupted_test]
df_test = pd.concat(concatenated_test)
# df_test

from sklearn.utils import shuffle
df_test = shuffle(df_test)
concatenated = [df_bankrupted, df_nonbankrupted,
df_val, df_test]
df = pd.concat(concatenated, ignore_index=True)

def clean_data(risk_report_test):
    """Returns cleaned text data."""

    with open(f"C:/… stop_words.txt", mode="r",
encoding="utf8") as file:
        stop_words = file.read()

    with open(f"C:/…stop_words_accounting.txt",
mode="r", encoding="utf8") as file:
        stop_words_acc = file.read()

    with open(f"C:/…onaji_vocab.txt", mode="r",
encoding="utf8") as file:
        onaji_vocab = file.read()


    char_filters = [UnicodeNormalizeCharFilter(),
                    RegexReplaceCharFilter(u'キャ
ッシュ・フロー', u'キャッシュフロー')]
    tokenizer = Tokenizer()
    token_filters = [CompoundNounFilter(),
                     POSStopFilter(['記号','助詞
', '数', '数接続', '接続詞','連体詞', '接頭詞', '名
詞,数','非自立', '代名詞', '自動詞', '他動詞']),
                     LowerCaseFilter()]


    a = Analyzer(char_filters=char_filters,
tokenizer=tokenizer, token_filters=token_filters)

    def filter(text):
        """
        :param text: str
        :rtype : str
        """
       # アルファベットと半角英数と記号と改行とタブを
排除
        text = re.sub(r'[a-zA-Z0-9¥"¥.¥,¥@]+',
'', text)
        text = re.sub(r'[!"“#$%&()\*\+\-
\.,\/:;<=>?@\[\\\]^_`{|}~]', '', text)
        text = re.sub(r'[\n|\r|\t]', '', text)

        # 日本語以外の文字を排除
        jp_chartype_tokenizer =
nltk.RegexpTokenizer(u'([ぁ-んー]+|[ァ-ン
```

```
ー]+|[\u4e00-\u9FFF]+|[ぁ-んァ-ンー\u4e00-
\u9FFF]+)')
        text =
''.join(jp_chartype_tokenizer.tokenize(text))

        return text

    nosymbol_test = filter(risk_report_test)

    doc = []
    for token in a.analyze(nosymbol_test):
        doc.append(token.surface)
    doc = [x for x in doc if x not in stop_words]
    doc = [x for x in doc if x not in
stop_words_acc]
    doc = [x for x in doc if x not in
onaji_vocab]
    doc = np.array(doc)

    return doc

from sklearn.feature_extraction.text import
CountVectorizer, TfidfTransformer
clean_file_transformer =
CountVectorizer(analyzer=clean_data).fit(df['text
_file'])


clean_file_bow =
clean_file_transformer.transform(df['text_file'])
transformer_tfidf =
TfidfTransformer().fit(clean_file_bow)
clean_file_tfidf =
transformer_tfidf.transform(clean_file_bow)

X = clean_file_tfidf
X = pd.DataFrame.sparse.from_spmatrix(X)
X_train = X[:92]
X_val = X[92:112]
X_test = X[112:]

y = np.array(df['Y'])
y_train = y[:92]
y_val = y[92:112]
y_test = y[112:]

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,
Dropout

model = Sequential([
    Dense(units=256, activation='relu',
input_dim=clean_file_tfidf.shape[1]),
    Dropout(0.5),
    Dense(units=1, activation='sigmoid')
])

model.compile(optimizer='rmsprop',
loss='binary_crossentropy', metrics=['accuracy'])
model.summary()

# TRAIN MODEL
from tensorflow.keras.callbacks import
EarlyStopping
early_stop = EarlyStopping(monitor='val_loss',
mode='min', verbose=1, patience=50)

model.fit(X_train, y_train,
validation_data=(X_val, y_val) , epochs=5000,
verbose=2, callbacks=[early_stop])

# PLOT THE LOSS
losses = pd.DataFrame(model.history.history)
losses.plot()
plt.show()

# NN model
pred_NN = []

for predictions in model.predict(X_test):
    if predictions < 0.5:
        predictions = 0
    else:
        predictions = 1
    pred_NN.append(predictions)

pred = np.array(pred_NN)
true = np.array(y_test)


# Classifier (Naive Bayes)
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
X_train_nb = np.array(X[:112])
y_train_nb = np.array(y[:112])
nb.fit(X_train_nb,y_train_nb)

pred_naive = nb.predict(X_test)

# Classifier (SVM)
from sklearn.svm import SVC

svm_model = SVC(C=1.0, kernel='linear', degree=3,
gamma='auto')
X_train_svm = np.array(X[:112])
y_train_svm = np.array(y[:112])

svm_model.fit(X_train_svm, y_train_svm)

pred_svm = svm_model.predict(np.array(X_test))
```

3.3. Chapter 5

```
# load json and create model
json_file = open('modelA.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)

# load weights into new model
loaded_model.load_weights("model_A_ens_T92.h5")
print("Loaded model from disk")

# evaluate loaded model on test data
loaded_model.compile(loss='binary_crossentropy',
optimizer='rmsprop', metrics=['accuracy'])
score = loaded_model.evaluate(X_test, y_test,
verbose=0)

print('TEXT:', "%s: %.2f%%" %
(loaded_model.metrics_names[1], score[1]*100))
```

```
# EVALUATE MODELS ON TEST DATA
predict_text = []

for predictions in loaded_model.predict(X_test):
    if predictions < 0.5:
        predictions = 0
    else:
        predictions = 1
    predict_text.append(predictions)

pred = np.array(predict_text)
true = np.array(y_test)


# load json and create model
json_fileR = open('modelR.json', 'r')
loaded_modelR_json = json_fileR.read()
json_fileR.close()
loaded_modelR =
model_from_json(loaded_modelR_json)

# load weights into new model
loaded_modelR.load_weights("model_ratio72.h5")
print("Loaded model from disk")

# evaluate loaded model on test data
loaded_modelR.compile(loss='binary_crossentropy',
optimizer='adam', metrics=['accuracy'])
scoreR = loaded_modelR.evaluate(Xtest_ratio,
y_test_ratio, verbose=0)

predict_ratio = []

for predictions in
loaded_modelR.predict(Xtest_ratio):
    if predictions < 0.5:
        predictions = 0
    else:
        predictions = 1
    predict_ratio.append(predictions)

predr = np.array(predict_ratio)
truer = np.array(y_test_ratio)

# ENSEMBLE MODEL
# Average probabilistic predictions of both
models

pred_NN_ratio = []
pred_NN_text = []

for predictions in
loaded_modelR.predict(Xtest_ratio):
    pred_NN_ratio.append(predictions)

for predictionss in loaded_model.predict(X_test):
    pred_NN_text.append(predictionss)

pred_ratio = np.array(pred_NN_ratio)
pred_ratio = pred_ratio.flatten()
pred_text = np.array(pred_NN_text)
pred_text = pred_text.flatten()

average = [statistics.mean(k) for k in
zip(pred_ratio, pred_text)]

# PERFORMANCE EVALUATION - AVERAGE

pred_overall = []

for predictions in average:
    if predictions < 0.50:
        predictions = 0
    else:
        predictions=1
    pred_overall.append(predictions)

predict = np.array(pred_overall)
actual = np.array(y_test)

# check overlapping predictions

df_b = pd.DataFrame(testbankrupted_risk_reports)
df_b.columns = ['file_name']
df_b['sec_code'] = df_b['file_name'].apply(lambda
x:x.split('_')[0])
df_b['filing_date'] =
df_b['file_name'].apply(lambda
x:x.split('_')[3].split('.')[0])
df_b['filing_date'] =
pd.to_datetime(df_b['filing_date'],
format='%Y%m%d')
df_b = df_b.drop(['file_name', 'filing_date'],
axis=1)
df_b['true class'] = 0

df_nb =
pd.DataFrame(testnonbankrupted_risk_reports)
df_nb.columns = ['file_name']
df_nb['sec_code'] =
df_nb['file_name'].apply(lambda
x:x.split('_')[0])
df_nb['filing_date'] =
df_nb['file_name'].apply(lambda
x:x.split('_')[3].split('.')[0])
df_nb['filing_date'] =
pd.to_datetime(df_nb['filing_date'],
format='%Y%m%d')
df_nb = df_nb.drop(['file_name', 'filing_date'],
axis=1)
df_nb['true class'] = 1

df_both = [df_b, df_nb]
concatenated_A = pd.concat(df_both,
ignore_index=True)
concatenated_A['ratio'] = pd.DataFrame(predr)
concatenated_A['text'] = pd.DataFrame(pred)
concatenated_A['ens_same'] =
pd.DataFrame(predict)
concatenated_A['ens_weighted'] =
pd.DataFrame(pred_best_weight)
```

3.4. Chapter 6

```
# BAGGING ENSEMBLE - COMBINED DATA
# EVALUATION ON TRAIN DATASET
bag_NN =
BaggingClassifier(base_estimator=MLPClassifier(),
                          n_estimators=20,
                          max_samples=0.8,
                          oob_score=True,
```

```
                            random_state=0)
bag_NN.fit(X_train, y_train_ratio)
bag_NN_ratios =
BaggingClassifier(base_estimator=MLPClassifier(),

n_estimators=12,

max_samples=0.8, oob_score=True,
                                    random_state=0)
bag_NN_ratios.fit(X_train_ratio, y_train_ratio)

bag_NN_text =
BaggingClassifier(base_estimator=MLPClassifier(),
                                  n_estimators=13,
                                  max_samples=0.8,
oob_score=True,
                                  random_state=0)
bag_NN_text.fit(X_train_text, y_train_ratio)

predictions_ratios = []
for predictions in
bag_NN_ratios.predict_proba(X_val_ratio):
    predictions_ratios.append(predictions[1])
# print(predictions_ratios)
predictions_text = []
for predictions in
bag_NN_text.predict_proba(X_val_text):
    predictions_text.append(predictions[1])
# print(predictions_text)

average = [statistics.mean(k) for k in
zip(predictions_ratios, predictions_text)]
print('average:', average)

pred_overall = []
for predictions in average:
    if predictions < 0.50:
        predictions = 0
    else:
        predictions=1
    pred_overall.append(predictions)

predict = np.array(pred_overall)
actual = np.array(y_val)

# BOOSTING ENSEMBLE
class customMLPClassifer(MLPClassifier):
    def resample_with_replacement(self, X_train,
y_train, sample_weight):

        # normalize sample_weights if not already
        sample_weight = sample_weight /
sample_weight.sum(dtype=np.float64)

        X_train_resampled =
np.zeros((len(X_train), len(X_train[0])),
dtype=np.float32)
        y_train_resampled =
np.zeros((len(y_train)), dtype=np.int)
        for i in range(len(X_train)):
            # draw a number from 0 to
len(X_train)-1
            draw =
np.random.choice(np.arange(len(X_train)),
p=sample_weight)

            # place the X and y at the drawn
number into the resampled X and y
            X_train_resampled[i] = X_train[draw]
            y_train_resampled[i] = y_train[draw]

        return X_train_resampled,
y_train_resampled

    def fit(self, X, y, sample_weight=None):
        if sample_weight is not None:
            X, y =
self.resample_with_replacement(X, y,
sample_weight)

        return self._fit(X, y,
incremental=(self.warm_start and

hasattr(self, "classes_")))

# Turn text dataframes into numpy arrays
X_train_text = X_train_text.to_numpy()
X_test_text = X_test_text.to_numpy()
X_val_text = X_val_text.to_numpy()

# BOOSTING ENSEMBLE - COMBINED DATA
# ADABOOST
num_estimators = 25

AdaBoost = AdaBoostClassifier(base_estimator=
customMLPClassifer(),

n_estimators=num_estimators,
                                    learning_rate=1)
AdaBoost.fit(X_train,y_train)

boostprediction = AdaBoost.score(X_train,y_train)

#Predict the response for test dataset
y_pred_test = AdaBoost.predict(X_test)
y_pred_val = AdaBoost.predict(X_val)

# BOOSTING ENSEMBLE - NOT COMBINED DATA

boost_NN_ratios =
AdaBoostClassifier(base_estimator=
customMLPClassifer(),
n_estimators=num_estimators,
learning_rate=1)
boost_NN_ratios.fit(X_train_ratio, y_train_ratio)

boost_NN_text =
AdaBoostClassifier(base_estimator=
customMLPClassifer(),
n_estimators=num_estimators,
learning_rate=1)
boost_NN_text.fit(X_train_text, y_train_ratio)
```

The codes and datasets can be accessed in GitHub from the following link: https://github.com/kaedeyo/bankruptcy_prediction