Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Jizhen Cai

Bankruptcy Prediction of a New Data Set of Companies in Norway

Master's thesis in Simulation and Visualization
Supervisor: Ibrahim A. Hameed, Hao Wang
June 2019

NTNU
Norwegian University of
Science and Technology

Jizhen Cai

Bankruptcy Prediction of a New Data Set of Companies in Norway

Master's thesis in Simulation and Visualization
Supervisor: Ibrahim A. Hameed, Hao Wang
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Abstract

In recent years, more and more researchers have been focusing on the research of bankruptcy prediction. However, traditional methods based on statistical models may not be able to deal with some new data sets, which are becoming more and more sophisticated than before. At the same time, new methods of data mining have been springing up for the last few decades. Therefore, in this master thesis, we discuss some data mining algorithms and apply those algorithms upon a new data new about the bankruptcy situations of companies in Norway for bankruptcy prediction. Additionally, some related data visualization approaches are also implemented.

Preface

This thesis is my concluding work of the degree of Master of Science in Simulation and visualization, Department of ICT and Engineering, Norwegian University of Science and Technology (NTNU). Firstly, I would thank Ibrahim A. Hameed, my main supervisor, because he has guided me in the writing of master thesis. Secondly, I would like to thank Associate Professor Hao Wang, who has given me useful suggestions on machine learning methods. Thirdly, I'll give my thanks to Wenqiang Cui, who has inspired me in the data visualization. Last but not least, I would give my deepest thanks to my girl friend Mengyao Gu, who always supports me with her love.

Aalesund, Norway
June, 4th, 2019
Jizhen Cai

Table of Contents

List of Tables

List of Figures

Abbreviations

LR	=	Logistic Regression
LDA	=	Linear Discriminant Analysis
SVM	=	Support Vector Machines
DT	=	Decision Trees
kNN	=	k Nearest Neighbor
NB	=	Naive Bayes
GNB	=	Gaussian Naive Bayes
MLP	=	Multi Layer Perceptron
RF	=	Random Forest
GB	=	Gradient Boosting
KMeans	=	k-means clustering

Chapter 1

Introduction

In this chapter, an introduction about this master thesis is given. Section 1.1 presents the motivation and background for this project. Section 1.2 mentions the scope of this master thesis. Section 1.3 describes the research questions that are discussed in this thesis.

1.1 Motivation

Business stress prediction and bankruptcy prediction have been heated-discussed topics for companies and corporations all over the world for the last few decades.

As a matter of fact, in the 1960s, some researchers such as Beaver (1966), Altman (1968) started to apply some methods on the problem of bankruptcy prediction. [2][3] Ever since then, a series of novel approaches have been applied. Relevant researches from Wilcox (1970), Ohlson (1980), Manski (1981), Gilbert et al. (1990), Shumway (2001), Chava et al. (2004) have made some progress and development, which have inspired other researchers on the problem of bankruptcy prediction. [4][5][6][7][8][9]

Traditionally, people heavily rely on some traditional statistical models, the assessment and judgment from relevant experts. However, nowadays, the development of novel financial indexes and the explosive growth in the volume of data have made it much harder to tackle with the problem of bankruptcy prediction using those traditional approaches.

At the same time, the techniques in data mining, machine learning and deep learning have been developed and improved in a very astonishing speed. Therefore, The field where data mining algorithms and the prediction of bankruptcy are combined together has drawn more and more attention from researchers and experts of related areas. In fact, the application of these methods can help us make bankruptcy predictions and find out those companies with possibility for bankruptcy, which can prevent some possible bankruptcy, or at least help both companies and stockholders to reduce their economic

losses in advance.

In view of that, we'll apply various data mining algorithms such as Logistic Regression, Support Vector Machines, Decision Tree, Random Forest, k Nearest Neighbor, Naive Bayes, Linear Discriminant Analysis, Multi Layer Perceptron, Stochastic Gradient Descent, Gradient Boosting into the field of bankruptcy prediction.

1.2 Scope and limitations

There are two reasons why we need to narrow down the scope that we are going to discuss in this project: Firstly, both the bankruptcy prediction and data mining are the fields with endless potential extensions, which make it necessary for us to study only the overlapping area of these two parts. Secondly, we only have a limited time for this project. Because of that, it's unrealistic even impossible to cover all aspects for the problem of bankruptcy prediction. Therefore, the constraints and limitations of this thesis are described as below.

1.2.1 Limitation 1: Only our original data set will be considered

Even though there are some other open data sets on bankruptcy prediction, like Polish companies bankruptcy data Data Set. [10] we choose only to adopt our novel data set about Norwegian companies. This decision is based on the following considerations:

Firstly, our main target is to find out the best model for the prediction of the bankruptcy of companies in Norway. This work is not only very original and unique, and also quite helpful for the government, experts and companies to make decisions, improve their finance situations and reduce potential losses.

Secondly, there can be some noticeable differences concerning the ways of sampling and collecting data among all these various data sets. In other words, it's likely that these different data sets are not comparable. In viewing of that, utilizing different data sets may not be a wise choice, especially when we take reliability and robustness of these data sets into considerations.

Thirdly, as we have mentioned above, due to the constraints of time, it's not practical to try all these data sets. It makes more senses to apply our methods only on our original data set and find out the best solution for the real world case of Norwegian companies, which is promising in bringing profits and benefits to the government and companies.

1.2.2 Limitation 2: Only reliable academic sources will be considered

In our master thesis, only the resources of scientific papers will be considered. This consideration is mainly due to the limitation of time. In view of that, we must define and narrow down scope of our references. Even though it's admitted that some other kinds of materials such as industrial reports, government reports may also be useful, we still

decide to put them aside. Otherwise, the workload can become too overwhelming to be completed. Moreover, compared with other sorts of resources, scientific papers are much more reliable, because most of the papers on those journals have to be assessed via the peer review from experts and examiners in relevant areas.

1.3 Problem Definition

The problem definition of this master thesis can be summarized using the following statement:

What are the conclusions that can be helpful in the bankruptcy prediction of the companies in Norway?

1.3.1 Objectives

The objectives of this master thesis are comprised of these following parts:

First and foremost, we would like to discuss and explore all the existed methods in the field of bankruptcy prediction. To be more precise, we prefer to studying the issue of bankruptcy prediction from the aspects of data mining methods and visualization methods, which is quite different from some traditional relevant researches focusing on the viewpoints of financial terms and theories.

Secondly, we intend to construct different data mining models and apply them on our original data set of Norwegian companies. By adjusting those amounts of hyper-parameters of these algorithms and comparing their results, we can find out the most robust and accurate model to help us predict the situation of bankruptcy of those companies.

Thirdly, the combination of the explorations of visualization methods and data mining algorithms is likely to presenting us some valuable and precious conclusions and ideas on how to predict the bankruptcy. For example, if we know a specific industry has a very high rate of bankruptcy, suggestions may be given to relevant administrators and leaders of such business. In other words, a good model with high precision in bankruptcy prediction may shed light upon the possibility of reducing economic losses and preventing business failure.

1.3.2 Research Questions

Now that we have presented our problem definition and the several objectives of this master thesis, we can now formulate our research questions that are corresponding to our objectives.

1.3.2.1 Research Question 1

Research Question 1: What are the methods that other researchers utilize in the field of bankruptcy prediction?

1.3.2.2 Research Question 2

Research Question 2: How can we use data mining algorithms and visualization methods in the field of bankruptcy prediction?

1.3.2.3 Research Question 3

Research Question 3: What are the conclusions and suggestions that we can get from our research?

1.4 Thesis structure

The structure of this thesis can be described as follows:

Chapter 2: Literature Review

This part will present what other researchers have done concerning the field of bankruptcy prediction for the past few decades. Both the visualization methods and data mining algorithms will be covered.

Chapter 3: Methods

This part gives detailed descriptions about the theories behind the methods that we utilize in the data mining and visualization.

Chapter 4: Experiment

This part mainly deals with the process about how we have conducted our experiments. Besides, the results and their comparisons will also be presented in this chapter.

Chapter 5: Analysis

In this part, we make analysis about the results we get.

Chapter 6: Conclusion

We'll mention the conclusions that we've gotten from this project, the lessons that we've been taught, the responses to the research questions in the objective part. Furthermore, we'll discuss the drawbacks of our project and experiments. These discussions may give guidance to further related work.

Chapter 2

Literature Review

2.1 Visualization

Keim et al. (2002) proposed an algorithm that can efficiently solve the problem of complex optimization in pixel placement. The usefulness of this algorithm was further confirmed using the data set from the real world. As a matter of fact, traditional simple graphics, such as bar charts, pie charts that were utilized in the visualization analysis, tended to being able to show only a very small number of features of data and have a high correlation among its features. The new visualization approach in this paper combined the intuitive feature of bar charts and the feature of screen space of showing much information. [11]

Diansheng Guo (2003) described a human-centered exploration environment. The tasks of uncovering patterns in high-dimensional data were made possible by using computational and visualization methods of this environment. In fact, the feature of high dimensionality in big data can lead to the difficulties of using many data mining algorithms. Therefore, practical and useful methods for dimension reduction and visualization of high dimension data must be formed. Viewing that, this environment solved the problems including the feature selection, automatic clustering for high-dimension data, visualization components. [12]

Melanie Tory and Torsten Mller (2004) presented a novel high-level visualization taxonomy, which was based on visualization algorithms instead of data. The rules of their classification were the features of being discrete or not, spatialization, timing, color and transparency. Even though traditionally visualization was categorized as two different classes as scientific visualization and information visualization, the authors of this paper proposed and maintained a totally different view in visualization taxonomy which was called model-based visualization taxonomy. Unlike traditional design model that heavily relied on the data type, the taxonomy made by authors of this paper emphasized the importance and influence of human in the visualization. [13]

Pick, James B (2004) introduced the basic ideas and conceptions of the Geographic Information System. GIS was widely used in the financial field, including finance, banking, retail, marketing, construction, city planning. In fact, GIS and its related applications and visualizations can successfully help the companies figure out the budget, make comparisons among the proposed plans, make the optimal schedule and decisions. GIS was often connected with other technologies such as Map servers, Hand-held GIS, Mobile wireless communications, IBM modeling and other databases, RFID, GPS. [14]

Wilkinson et al. (2006) proposed an approach in finding out most appropriate ways in the exploration of high dimension data. Visual Analytics was mainly used for fields including checking raw data for determining anomalies, exploring data to discover plausible models, checking model assumptions. This was very useful especially when high-dimensional data was needed to be dealt with. Some related works on projections, which were utilized for dimension reduction in many different circumstances, manifolds and features, made use of clustering and mapping for visualization. Feature measures were comprised of outliers, density, shape and association. Based on these features and methods, researchers were able to visualize using SPLOM or in parallel coordinate plots. [15]

Keim et al. (2008) mentioned the scope and challenges of visual analytics. Because of the explosion of information and knowledge, the speed of the creation of new data was much faster than what researchers can analyze. Therefore, there were more studies on visual analytics, which focused on presenting information more efficiently, explicitly, interactively. Visual Analytics was more than just visualization. Instead, it included Interaction, Scientific Analytics, Statistical Analytics, Information Analytics, Geospatial Analytics, Knowledge Discovery and Data Management. Even though Visual Analytics was widely applied in various fields including Physics and Astronomy, Business Analysis, Environmental Monitoring, Disaster and Emergency Management, Software Analytics, Mobile Graphics and Traffic, there still existed many problems. Challenges such as human information discourse, user acceptability, data quality and uncertainty must be noticed and solved in further studies. [16]

Tatu et al. (2009) proposed an approach that can help the users in exploring and visualizing overwhelming amounts of data, especially when there existed some high-dimensional data. Usually, some unique features of the high-dimension data can lead to great difficulties in visualizing them. For example, the correlations among those countless variables can make it a very difficult decision to select the most proper features for visualizations. In other words, without appropriate guidance, it was likely that some figures that had little relevance with the users would be created. For high-dimension data, mainstream visualization approaches included Scatter Plots, Parallel Coordinates. Based on these two common approaches, the researchers proposed ranking functions that can measure the quality of classified and unclassified data. Then their proposed methods can search and find the best patterns for visualization. [17]

Enrico Bertini and Denis Lalanne (2010) categorized the observed techniques in visualization and data mining. Besides, the authors proposed some extensions and methods that had not been explored in the former works of other researchers. Unlike data mining, which focused on the machine part, visual analytics placed more factors and weights on

the human part. The authors mentioned several categories in this paper. The first category was enhanced visualization, which included projection of Multidimensional Scaling, intelligent data reduction, pattern disclosure. The second category was enhanced mining, which included Nomograms of SVM and patterns exploration and filtering. The last one was integrated visualization and mining, which included white-box integration, bracketing technique of black-box integration. The authors also suggested several methods in improving the data analysis, such as enhancement in the building of visual model, improving the process of verification and refinement, augmenting in the prediction building. [18]

Lisa Meloncon and Emily Warner (2017) made researches on the need, development and practical application of data visualization in all different fields. In the past decade, there existed amounts of data, which led to a huge demand for various approaches for data visualization. However, even though data visualization was widely used, the challenges and difficulties of the communication of researchers from different fields and background were big problems. The authors mentioned that the following types of visuals were most popular: numbers and icons, pictographs, bar graphs, pie charts, bar charts, flow charts, funnel plots. The authors came to the conclusions that pictographs/icon arrays, bar graphs were excellent forms of visualization. Besides, the visualizations should be kept as simple as possible and more attention should be paid to the design features. [19]

2.2 Data Mining Methods

Choong Nyoung Kim Raymond McLeod Jr. (1999) used two inductive learning method. The first method was ID3 method, which consisted of a procedure for generating an efficient discrimination tree for classifying various features and types. The second was neural network, which was able to train various parameters and construct sophisticated models. [20]

Sung et al. (1999) made a comparison among Discriminant Analysis, Decision Tree, Neural Networks, Genetic algorithms concerning the problem of bankruptcy prediction. In the end, the researchers chose the decision tree techniques as the main models they used. The result of multivariate discriminant analysis was compared to show the performance of model. Additionally, even though neural networks harvested good results, they were abandoned because of the problem in interpretability. [21]

Hui Li and Jie Sun (2011) conducted some experiments using case-based reasoning (CBR) on the data set of Chinese companies and proved the usefulness of the algorithm of CBR. Besides, the algorithm of Support Vector Machine was also implemented on the same data set so that the results can be compared and analyzed. In fact, Case-based reasoning had been utilized in the field of business future prediction for a long time. Even though there were some discussions on the disadvantages of CBR, it was still one of the most widely used algorithm in financial prediction. [22]

Gang Wang and Jian Ma (2011) proposed an integrated ensemble approach, called RS-Boosting. This novel method combined two different ensemble methods, which were boosting and random subspace. In this paper, several other credit risk prediction methods, including Logistic Regression Analysis, Decision Tree, Artificial Neural Network, bagging, boosting and random subspace were also implemented. The results showed that RS-Boosting behaved better than all the other algorithms. However, this experiment can be further improved because this paper only used Decision Tree as base classifier. [23]

Lin et al. (2011) proposed a hybrid manifold learning approach method which combined isometric feature mapping algorithm and support vector machines to deal with the problem of business failure prediction. The data was firstly processed using ISOMAP, then after the kernel selection and parameters selection, the data was imported into the SVM classifier. Additionally, in this paper, the Principal Component Analysis was also introduced and implemented as a comparison and benchmark. [24]

Li et al. (2011) proposed a novel multiple criteria CBR method for binary business failure prediction (BFP) with similarities to positive and negative ideal cases (SPNIC). The results from these experiments showed that this novel approach perform much better short-term discriminate capability than comparative methods. In this paper, some methods such as MDA, Logit, Probit, CBR with kNN and CBR with decision tree were implemented as baselines and comparisons for their proposed new method. [25]

Soo Y. Kim (2011) provided an optimal model which can minimize the empirical risk of classification of bankruptcy prediction. Algorithms including Multivariate discriminant Analysis, Logistic Regression, Neural Networks and Support Vector Machines were all

tested and compared. When these algorithms were evaluated from the aspects of type I error, type II error, it was obvious that Artificial Neural Network performed much better than other algorithms. [26]

Arindam Chaudhuria and Kajal De (2011) dealt with the issue of the problem of bankruptcy prediction using a novel Soft Computing tool, which was called Fuzzy Support Vector Machine. This approach combined the popular machine learning algorithm called Support Vector Machine and Fuzzy Sets, which were capable of handling uncertainty. Because of this, FSVM was evidently much better and more robust than a single algorithm. Additionally, The result of clustering effect of Probabilistic neural networks on bankruptcy data sets was also compared with the result of FSVM. The result showed the superiority of FSVM.[27]

Bhattacharyya et al. (2011) evaluated two advanced data mining approaches, support vector machines and random forests, together with the well-known logistic regression. It was seen that Logistic Regression performed competitively with the more advanced techniques on certain measures, especially in comparison with SVM and where the class imbalance in training data was not large. It showed better performance than SVM on sensitivity except where the class imbalance in the training data became large (for DF4, with 2% fraud). The precision, F, G-mean and wtdAcc measures showed a similar comparison between LR and SVM. LR was also seen to exhibit consistent performance on AUC across the different training data sets. Random Forests showed overall better performance than the other techniques on all performance measures. [28]

Ravisankar et al. (2011) used data mining techniques such as Multilayer Feed Forward Neural Network (MLFF), Support Vector Machines (SVM), Genetic Programming (GP), Group Method of Data Handling (GMDH), Logistic Regression (LR), and Probabilistic Neural Network (PNN) to identify companies that resort to financial statement fraud. Results based on AUC indicated that the PNN was the top performer followed by GP which yielded marginally less accuracy in most of the cases. Also, the results obtained in this study were better than those obtained in an earlier study on the same data set. Ten-fold cross-validation was performed throughout the study. Prediction of financial fraud was extremely important as it can save huge amounts of money from being embezzled. [29]

Hardle et al. (2012) combined the Support Vector Machine and genetic algorithm to help the prediction of default. Because SVM had various hyperparameters that needed to be set by the users of the algorithms, the genetic algorithm can help optimize those SVM parameters. Besides, some classical methods such as discriminant analysis, logit and probit models were also introduced and compared with the results of the model of SVM. [30]

Gang Wang and Jian Ma (2012) proposed a new hybrid approach called RSB-SVM to deal with the credit risk assessment, which used Support Vector Machine as a base learner and bagging and random subspace. In additionally, some other models based on machine learning algorithms such as Linear Regression Analysis, Decision Trees, Artificial Neural Network were also implemented as benchmarks. The experiments showed that RSB-SVM outperformed bagging and random subspace in getting more various component SVM classifiers. [31]

Divsalar et al. (2012) implemented some gene expression programming and multi-expression programming to construct models to deal with bankruptcy prediction. GEP was based on genetic programming, which was a very suitable approach in optimization. In other words, regardless of our initial input, this algorithm can always try to find the fittest option by mutation and match. Unlike GEP, MEP uses linear chromosomes. When compared with traditional statistical methods, it was obvious that GEP and MEP can get rid of the difficulty in pre-defining equations. [32]

Myoung-Jong Kim and Dae-Ki Kang (2012) proposed a genetic algorithm-based coverage optimization technique. This technique can avoid the problem of multicollinearity in the bankruptcy prediction. In this paper, single classifiers such as Decision Tree, Neural Network, Support Vector Machine were implemented respectively first. Then ensemble methods like Boosting, Bagging, CO-Boosting, Co-Bagging were implemented. [33]

Hsieh et al. (2012) proposed a variant of SVM by introducing a penalty function. The introducing of Particle Swarm Optimization (PSO) helped in creating the PGSVM. Besides, some other algorithms such as back-propagation neural network and classical SVM optimized by ABC algorithm were also implemented and compared with results of model. The proposed EABC-PGSVM method outperformed other comparable methods. [34]

Andrey et al. (2012) proposed a method that extracted information from sequences of financial ratios and investigated the usefulness of this information for bankruptcy prediction. In this paper, the approach of Markov for Discrimination mapped a time-varying sequence of ratios into one independent variable in prediction models. The results showed that the Markov for Discriminant predictors greatly improved the performance of prediction of bankruptcy. [35]

Olson et al. (2012) made comparisons among these common algorithms that were implemented in the field of bankruptcy prediction, including Neural Networks, Support Vector Machines, Decision Trees. However, these algorithms had their own disadvantages. For instance, information gotten from neural networks was hard to explain and understand, which can be a terrible thing especially in the financial field. Various decision tree algorithms including Quinlans ID3, C4.5, C5 and CART were all implemented. The experiments showed that C5 decision tree was the most suitable one. [36]

Jie Sun and Hui Li (2012) proposed a novel approach based on SVM in dealing with financial stress prediction. In this paper, SVM kernels such as linear, polynomial, RBF and sigmoid, and the filter feature selection/extraction methods of stepwise multi discriminant analysis (MDA), stepwise logistic regression (logit), and principal component analysis (PCA) were all applied. The researchers proposed the criteria for selecting base SVM classifiers and the combination mechanism of the SVM ensemble. From the result, if the user can only utilize one algorithm, then RBF-SVM combined with stepwise MDA would be the fittest choice. [37]

Huang et al. (2012) proposed a kernel local fisher discriminant analysis based manifold-regularized SVM model to solve the issue of the prediction s of financial distress. Usually, the feature of high dimension and nonlinear distributed data can bring about a noticeable

bad effect on the performance of various algorithms on the financial problems. Five different classifiers were implemented in this paper, including decision tree, nearest neighbors with three neighbors, logistic regressions, Bayesian and RBFNetwork. [38]

Lin et al. (2012) studied the various machine learning algorithms in the field of financial crisis prediction. These algorithms included Decision Trees, Support Vector Machines, Neural Networks, Case-based Reasoning, k-nearest Neighbor, self-organizing maps, k-means, Expectation Maximization, Logistic Regression, Nave Bayes, Discriminant Analysis, Data Envelope Analysis, Iostonic Separation, Mahalanobis-Taguchi, Genetic algorithms, Group Method of Data Handling, Rough Sets, Fuzzy Sets. Also, some other ensemble algorithms were also implemented, and their results were compared and analyzed. [39]

Jae Kwon Bae (2012) proposed a financial distress prediction model based on radial basis function support vector machines. The author also compared the classification accuracy performance between this RSVM model and other artificial intelligence techniques so that some useful suggestions could be given to relevant experts. These compared techniques included Multiple Discriminant analysis, Logistic Regression, Multi-layer Perceptron, Classification Tree Algorithms, Bayesian Networks. [40]

Chih-Fong Tsai and Kai-Chun Cheng (2012) applied a simple distance-based clustering outlier detection method upon Australian, German, Japanese and UC competitions datasets. Also, several classification approaches, including artificial neural networks, decision trees, logistic regression and support vector machines were also implemented so that their results can be compared with the result from the proposed method. The results showed that SVM can get the most accurate result and be the relatively most robust one. [41]

Arieshanti et al. (2013) used various methods including k-NN, fuzzy k-NN, SVM, Bagging Nearest Neighbour SVM, Multi Layer Perceptron, hybrid of MLP and multiple linear regression to deal with the problem of bankruptcy prediction. The results were that Fuzzy k-NN got the best accuracy. Next to the Fuzzy k-NN, k-NN had the accuracy of 75.42%, which ranked second. The third place was gotten by the hybrid of MLP and Linear Regression. [42]

Mu-Yen Chen (2013) used particle swarm optimization and subtractive clustering to form a hybrid ANFIS model in handling bankruptcy prediction. ANFIS was a multi-layer adaptive network-based fuzzy inference system. From some aspects, ANFIS was similar to Fuzzy Logic and Artificial Neural Network. This model was tested on the data set about 160 electronics companies and predict their performances. [43]

Lin et al. (2013) proposed a hybrid model that combined locally linear embedding algorithm and support vector machines in tackling the bankruptcy prediction issue. LLE was able to map the high dimension data into low dimension data. In the experiment part, the researchers implemented the LLE and PCA (Principal Component Analysis) and compared their results. The conclusion was this proposed method had better classification accuracy as well as fewer Type I and Type II errors. [44]

Chih-fong Tsai and Yu-Feng Hsu (2013) proposed a meta-learning framework for

bankruptcy prediction. Relevant experiments were conducted on five different data sets, including Australian dataset, German dataset, Japanese dataset, Bankruptcy dataset, UCSD competition data set. From the results, they successfully got the performance of MLP, CART, LR and hybrid methods combing them. [45]

Rao et al. (2013) used Altman Z-score and KMV Merton Distance as tools to form models in handling the issue of bankruptcy prediction. Altman Z-Score Model can utilize various financial indexes to predict the financial stress of a company which can be very useful in bankruptcy prediction. While at the same time, KMV-Merton distance default can get the probability of default for each company at any time, which was more flexible. [46]

Fedorova et al. (2013) applied different kinds of modern learning algorithms including MDA, LR, CRT and ANNs to help determine the most efficient algorithm in the bankruptcy prediction. Both multi-layer perceptron and radial basis function network were implemented in the experiments. With the combinations of financial indexes, the researchers achieved 88.8% of overall accuracy in the end. [47]

Chen et al. (2013) used self-organizing map (SOM) to convert temporal sequence into trajectory vector. Then the trajectory self-organizing map clusters the trajectory vectors to a number of trajectory patterns. In fact, SOM had the advantage of data abstraction and spatialization, which made it a very suitable tool for the data processing and visualization for high dimension data. As a matter of fact, its feature in abstracting the data into 2-D dimension and showing them dynamically can help improve the interpretability especially in the field of bankruptcy prediction. [48]

Carlos Serrano-Cinca and Begoa Gutirrez-Nieto (2013) used Partial Least Square Discriminant Analysis (PLS-DA), which was a PLS regression with a dichotomous dependent variable. In this paper, other 8 common algorithms including LDA, LR, MLP,KNN, NB, SVM, C4.5, BRT were also implemented and compared with the results from the PLS-DA in dealing with the data concerning USA banking crisis. In fact, the PLS-DA had its own unique advantage in dealing with the data that had the problem of multicollinearity. [49]

Ligang Zhou (2013) studied the relationship between the sampling methods and quantitative bankruptcy prediction models. In this paper, seven sampling methods and five quantitative models were tested on the real data sets. Methods including Random oversampling with replication, SMOTE, Random Undersampling, Undersampling Based on Clustering from Nearest Neighbor were formed and tested on the data set of USA Bankruptcy Dataset and Japanese Bankruptcy Dataset. This paper apparently showed the importance of choosing suitable sampling methods in the bankruptcy prediction. [50]

Xiong et al. (2013) utilized the credit card data to help predict the personal bankruptcy prediction. In this paper, the researchers worked to take sequence information, sequence patterns that were extracted from data mining. Those information and features were then combined with those features extracted from Support Vector Machine classifier. [51]

Birsen Eygi Erdogan (2013) applied the algorithm of Support Vector Machine using different variable sets and with all variables separately. In this paper, the author set different values of gamma and cost parameters. The results showed that when gamma was set to

1 and cost was set to 8, the error of the model was 0.10 and the sensitivity was 0.92. [52]

Ahmed et al. (2013) firstly gave some introduction on the outlier detection. In fact, outlier detection had been a significant field of detecting abnormalities in various application domains including clustering-based disease onset identification, gene expression analysis, computer network intrusion, financial fraud detection and human behavior analysis. Existed methods to detect outliers were inadequate due to poor accuracy and lack of any general technique. Most techniques considered either small clusters as outliers or provide a score for being outlier to each data object. These approaches had limitations due to high computational complexity and misidentification of normal data object as outliers. In this paper, they provided a novel unsupervised approach to detect outliers using a modified k-means clustering algorithm. The detected outliers were removed from the dataset to improve clustering accuracy. They validated their approach by comparing against existing techniques and benchmark performance. Experimental results on benchmark datasets showed that their proposed technique outperformed existed methods on several measures. [53]

Wang et al. (2014) proposed a novel method called FS-Boosting in handling corporate bankruptcy. Unlike these traditional statistical methods including LDA, MDA, QDA, LRA and FA, FS-Boosting was a machine learning algorithm based on ensemble algorithm. In this paper, some several methods such as LRA, NB,DT, ANN,SVM, Bagging and boosting were implemented and compared with the result of the method of FS-Boosting. FS-Boosting got a relatively high accuracy because it reduced the type II error. [54]

Chih-Fong Tsai (2014) studied the case of financial distress prediction using cluster analysis with classifier ensembles. In this paper, not only methods including Multilayer-perceptron neural network, Support Vector Machine, Decision Tree, Genetic algorithm, K-nearest neighbor, Case-based reasoning and fuzzy set theory were implemented, the hybrid methods combining several of them were also created and implemented. The author found that the combing SOM with classifier ensembles by the weighted voting approach can get the best prediction result. [55]

Niccol Gordini (2014) conducted some researches using genetic algorithms, logistic regression, support vector machine on data set for small and medium-sized enterprises. Experiments were done by using different conditions in size and geographical area. The result was that the Genetic algorithms beat all the other algorithms in the performance of the prediction of bankrupt and non-bankrupt cases. [56]

Yan Huang and Gang Kou (2014) proposed a kernel entropy manifold learning in handling financial data analysis. In financial data analysis, some machine learning algorithms including Artificial neural network, Support Vector Machine, SOM, Partial least square regression and principal component analysis were usually adopted. Kernel entropy manifold learning algorithm was able to map the high-dimension data into low-dimension data so that the most important features can be extracted. [57]

Junyoung Heo and Jin Yong Yang (2014) pointed out that traditional methods for the bankruptcy forecasting for general companies were not very suitable for construction companies that had big liquidity. In this paper, the researchers proposed that the model of Ad-

aBoost was likely to being the most suitable one. In fact, when compared with algorithms such as ANN, SVM, Decision Tree, Z-score, the AdaBoost apparently got better results, especially when applied to the data set of large-sized construction companies whose capitals were relatively high. [58]

Yu et al. (2014) proposed the method of Leave-One-Out-Incremental Extreme Learning Machine (LOO-IELM). This method was mainly based on the structure of a single Hidden Layer Feedforward Neural Network. In the experiments of the researchers, about 12 features of the data set were selected by financial experts, while around 9 features were selected by this model. Finally, this LOO-IELM model got better results. [59]

Ming-Chang Lee (2014) discussed various existed algorithms including Neural networks, Bayesian classifier, Discriminant analysis, Logistic regression, K-nearest neighbor, decision tree, case base reasoning, support vector machine, software computing, fuzzy rule-based system and hybrid models. However, the author proposed a survival analysis method called cox model. [60]

Yu et al. (2014) proposed an approach called Delta Test-ELM, which operated in an incremental way to create less complex ELM structures and determine the number of hidden nodes automatically. In addition, Bayesian Information Criterion and Delta Test were utilized as well. In the end, it showed that DT-ELM got the best performance in the results. [61]

Yeh et al. (2014) utilized the going-concern prediction using hybrid random forests and rough set. In relevant experiments, the researchers implemented pureRST, RF+DT, RF+NN, RF+SVM. However, there were some things that can be further explored. For example, the combining of NN, DT, SVM and going-concern may bring something different. [62]

Joaqun Abelln and Carlos J. Mantas (2014) used the decision tree and ensemble classifiers like Random Subspace, Bagging. From the results, the best result was gotten from B-CDT method. This paper successfully constructed a new procedure to build decision trees, which was called Credal Decision Trees. The most important point of this paper was that the model it proposed can be applied to many other fields instead of only bankruptcy and credit scoring. [63]

Lin et al. (2014) proposed an integrated approach to feature selection for the financially distressed prediction problem. In recent years, classifiers such as MDA, Logit Regression, Neural Network, Decision Tree, Support Vector Machine, Case-Based Reasons were implemented to solve relevant problems. However, in this paper, the researchers proposed a wrapper algorithm based on the genetic algorithm which was called HARC. And the researchers found that those models built with the HARC feature beat the models built with expert knn model. [64]

Gintautas Garva and Paulius Danenas (2014) proposed particle swarm optimization for linear support vector machines based classifier selection. Even though SVM was widely used in various fields, it still had the problems of inflexibility in modeling. Therefore, the researchers created a novel method combing the PSO and linear SVM. This gave more possibilities to apply this to other various cases. This algorithm performed well both on

the German dataset and the Australia dataset. [65]

Renu and Suman (2015) classified the types of fraud detection. Besides, they also listed the common techniques for fraud detection. Six fraud detection methods were mentioned in their work, including Bayesian networks, hidden markov model, genetic algorithm, decision tree, support vector machine and neural network. The authors maintained that accuracy in fraud detection can be improved by combining various methods in the future. [66]

Emanuel et al. (2015) applied the cluster analysis and artificial neural networks to a real case of credit card fraud detection. As the paper mentioned, the Cluster Analysis (CA) was used to automatically normalize qualitative data. It consisted of a series of techniques and algorithms that were able to separate data in homogeneous clusters, according to a similarity criterion. However, CA can result in information loss, which caused data to lose its ability to explain some fraud behavior. To prevent this, a metric was applied to minimize losses, named Information Gain. [67]

Mahmoudi et al. (2015) investigated a linear discriminant, called Fisher Discriminant Function for the first time in credit card fraud detection. Besides, they also proposed a modified fisher discriminant function to counter the problem of biases in classification methods. [68]

Kim et al. (2015) proposed a novel method called geometric mean based boosting algorithm (GMBoost) to resolve data imbalance problem. Some similar algorithms like AdaBoost and cost-sensitive boosting were also implemented and compared with the result of the proposed model. Besides the researchers found that the Smote was a good solution in dealing with imbalanced data set. [69]

Gergely FEJR-KIRLY (2015) made some summaries on the evolution of the techniques in the bankruptcy prediction. In fact, the evolution of bankruptcy prediction approaches can be divided into three stages. In the first stage, people heavily relied on the ratio analysis on bankruptcy prediction. In the second stage, some multivariate analysis tools including MDA, LA, PA were introduced. In the third stage, more various models combining theories of AI were created, such as neural network analysis, ANN, mixed logit model, hybrid method combing Fuzzy kNN with GA, Bayesian model. [70]

Hafiz et al. (2015) studied the bankruptcy prediction using the methods in the field of big data. In the algorithm part, the researchers discussed approaches including Multi-discriminant Analysis, Logit analysis, Artificial Neural networks, Support Vector Machines, Rough Sets, Case Based Reasoning, Iterative Dichotomiser, Genetic Algorithm. Generally, the framework like Apache offered similar tools and methods to implement these algorithms on big data. [71]

Kalyan Nagaraj and Amulyashree Sridhar (2015) discussed the necessary steps in dealing with bankruptcy prediction including data collection, data pre-processing, development of models, knowledge extraction. In addition, classification algorithms including Logistic regression, Rotation forest, Nave Bayes, Neural Networks, RBF-based support vector machine were studied and implemented. Their respective accuracy and precision were listed and compared. [72]

Iturriaga et al. (2015) utilized the methods of multilayer perceptrons and self-organizing maps to help determine and predict the bankruptcy stress in the three years before the occurrence of bankruptcy. The researchers compared the results of the model combing MLP and SOM with the results from other algorithms including discriminant analysis, LR, SVM and RF. [73]

Philippe du Jardin (2016) studied the problem of bankruptcy prediction by using ensemble techniques such as bagging, boosting, random subspace and different methods including random forest, decision tree, logistic regression and neural networks. [74]

Mansouri et al. (2016) utilized the approaches of artificial neural network model and logistic regression to conduct some reseaches on the issue of bankruptcy prediction. The researchers tried to make bankruptcy predictions about companies in the cases of three years advance, two years advance, one years advance. They found that the ANN model beat the model of linear regression in all cases. [75]

Zieba et al. (2016) proposed a novel method in handling the issue of bankruptcy prediction. This method combined both Extreme Gradient Boosting and Decision Trees. They also formed a novel idea that was called synthetic features. With the help of these features, they were able to accomplish the tasks of bankruptcy prediction before several years. What was more, various methods, including LDA, MLP, JRip, J48, CJ48, AdaBoost, AdaCost, SVM, CSVM were also experimented and compared with the main methods. [10]

Azayitea et al. (2016) proposed a hybrid model of neural networks using discriminant analysis, multiplayer neural network and self-organizing maps. They found that the introduction of a dynamic layer can greatly improve the results. [76]

Kim et al. (2016) discussed the problem of handling imbalanced data set concerning the classification issues, including oversampling like SMOTE and MSMOTE, undersampling like OSS and WE. In this paper, the researchers proposed cluster-based evolutionary understanding method, which combined clustering and GA to deal with the data imbalance. Besides, the model of artificial neural networks was also adopted. [77]

Zhao et al. (2016) proposed a novel model using kernel extreme learning machine to handle the issue of bankruptcy prediction. Some other methods including support vector machines, extreme learning machine, random forest, particle swarm optimization, fuzzy kNN were also implemented and compared with the results of this model. [78]

Barboza et al. (2017) adopted several artificial intelligence algorithms, including support vector machines, bagging, boosting, random forest, artificial neural networks, discriminant analysis and logistic regression to deal with the issues of the prediction of bankruptcy. In their results, they found the models including MDA, linear regression and artificial neural networks were worse than the performance of other machine learning algorithms. However, there was one exception here. The SVM performed not so good because of the feature of high dimension of the data set. [79]

Nanxi Wang (2017) applied three relatively new methods including support vector machine, neural network with dropout and autoencoder in the problem of bankruptcy prediction. In fact, these algorithms harvested very good results especially when compared

with algorithms including logistic regression, genetic algorithm and random forest. But it was also worthwhile noticing the fact that both SVM and autoencoder had their own disadvantages. [80]

Martens et al. (2017) used ant colony optimization to deal with the problem of credit rating prediction. The author avoided some black-box algorithms like artificial neural networks because the results from these algorithms can be very difficult to explain. Instead, the author used AntMiner+, a classification algorithm that was created based on the ideas of ant colony optimization. Now that AntMiner extracted rule sets from data sets, this method can perfectly avoid dilemma in explaining. Whats more, AntMiner got a good result but used less rules than C4.5. [81]

Wang et al. (2017) proposed a new kernel extreme learning machine (KELM) with the help of an algorithm called grey wolf optimization (GWO) to help handle the problem of bankruptcy prediction. KELM was a variant of ELM. In this paper, researchers had done experiments using algorithms including GWO-KELM, PSO-KELM, GA-KELM, GS-KELM. Experiments were both done on the Wieslaw dataset and Japanese bankruptcy dataset. Methods performed well on both datasets. [82]

Chou et al. (2017) proposed a hybrid structure integrating statistical theory and computational intelligence technique which was based on the genetic algorithm with statistical measurements and fuzzy logic based fitness functions for key ratio selection. With the intention of making comparisons, the well-known BPNN classifier was also implemented. The researchers also discussed some well-applied methods in the field of prediction model design, including traditional statistical method, Linear Discriminant Analysis, Logistic Regression, LVQ, Data Envelopment Analysis, Case-Based Reasoning, Decision Tress, Fuzzy Logic, Rough Set, Neural Network, Kohonen Map, Support Vector Machines, Genetic Algorithm, Particle Swarm Optimization and Soft Computing. [83]

Sun et al. (2017) proposed two methods to deal with the problem of dynamic financial distress predication. The first was DEVE-AT, which tried to combine the outputs of Adaboost-SVM and Timeboost-SVM. This method considered the issue of misclassification and issue of time of samples at the same time. The second is ADASVM-TW. This method combined Adboost-SVM and time weighting. [84]

Frank Wagenmans (2017) studied the issue of bankruptcy prediction. The author utilized AUC-curve and ROC-curve as the indicators for performance measures. Some algorithms including decision tress, random forest, logistic regression, neural networks were used in this research. After comparing and analysis, the author found that the approach of random forest beat other algorithms and was the most robust one. After that, the performance of logistic regression with regularization was next to it. [85]

Chapter 3

Methods

3.1 Logistic Regression

Logistic regression is widely used in the areas where researchers need to classify all the data into two classes. Because of this, Logistic Regression is often applied in medical prediction and assessment. For example, in the problem of the prediction of lung cancer. Patients can have multiple different variables, including ages, genders, weights, heights, whether taking cigarettes or not, eating habits, living conditions, working conditions, level of education, but they will be classified into two categories finally. [86]

Similarly, Logistic Regression is also utilized in the financial areas including the credit card fraud detection, financial stress prediction. What's more, in the areas such as email spamming, we can also use logistic regression to help us determine whether an email is a rubbish email or not. The contents in the email such as title, key words can work as the features of input to help us construct models using logistic regression to filter out useless emails. [87]

In Logistic Regression, the output y can range from 0 to 1. In fact, this can also be the probability of being the potential target. Compared with Linear Regression, the advantage of Logistic Regression is that it maps the result into the scope of [0,1]. In real cases of prediction, we usually deem the value of 1 as the positive result, while we also deem the value of 0 as the negative result.

Here we can have the Sigmoid function:

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

3.2 Linear Discriminant Analysis

Linear Discriminant Analysis, which is also called as Fisher Discriminant Analysis is widely utilized in the area of classification and dimension reduction. It's a supervised machine learning algorithm with extraordinary ability in feature extraction and robustness in countering noises.

The core idea behind Linear Discriminant Analysis is that when we have gotten a specific training data set, we need to get the projections of all the data on a defined line. This line must satisfy this kind of requirement: the projections of the data that belong to the same class should be as close as possible, while the projections of those data that belong to different classes should be as far as possible. [88]

For example, supposing we need to deal with a binary classification problem. And there are 6 points in this specific data set. Undoubtedly, there can be countless different methods for choosing the line and thus their projections also be divergent. However, when we make a comparison between Figure 3.1 and Figure 3.2, we can surely notice that the choice of line in Figure 3.2 is much better than the one in Figure 3.1.

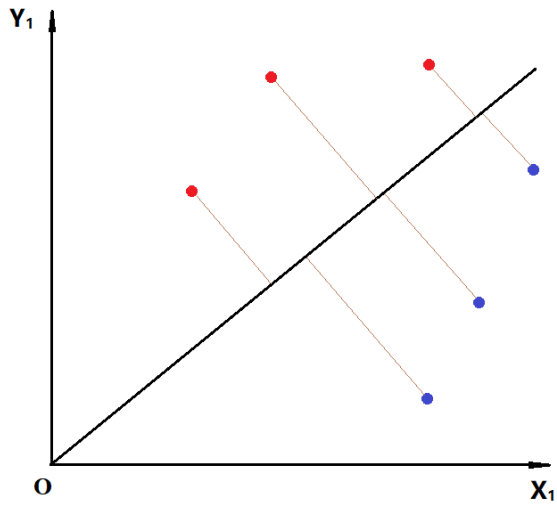


Figure 3.1: LDA-inappropriate choice of line

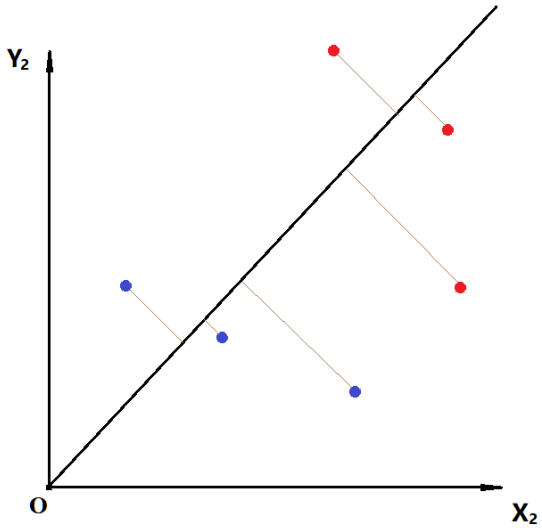


Figure 3.2: LDA-appropriate choice of line

Saying we have a data set D = $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, $y_i \in \{-1, +1\}$

Now let's say X_i, μ_i, Σ_i represent the set, average vector, co-variance of the i-th class. On this case, saying there's line called ω. Then for this binary classification problem, the center of these two projections are respectively $\omega^T\mu_0$ and $\omega^T\mu_1$. Meanwhile, the co-variances of these two classes of data are respectively $\omega^T\Sigma_0\omega$ and $\omega^T\Sigma_1\omega$.

As is mentioned above, it's desired that those data that belong to the same class should make their projections be as close as possible. Because of that, we hope the value of ($\omega^T\Sigma_0\omega$ + $\omega^T\Sigma_1\omega$) be as small as possible. Similarly, the aim of letting the projections of different classes being as far as possible makes it necessary to get the maximum value of $|\omega^T\mu_0$ - $\omega^T\mu_1|$. [89]

To sum what has been discussed above, the task of finding the most appropriate line in LDA equals to the find the maximum value of the following Loss Function.

$$\text{Loss Function} = \frac{|\omega^T\mu_0 - \omega^T\mu_1|}{\omega^T\Sigma_0\omega + \omega^T\Sigma_1\omega}$$

3.3 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm which is able to classify the data into two classes. The most important thing of SVM is to find out the hyperplane that can successfully divide the data into binary states. [90]

Supposing that we have gotten a training data set which can be represented as

D = $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, $y_i \in \{-1, +1\}$

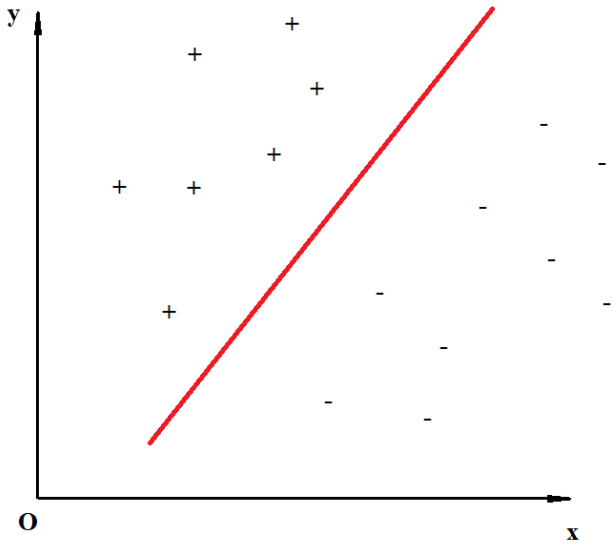


Figure 3.3: SVM Hyperplane

In fact, this hyperplane can be represented as:

$$\omega^T \mathrm{X} + \mathrm{b} = 0$$

where $\omega = (\omega_1, \omega_2, \cdots, \omega_n)^T$, which denotes the slope of the hyperplane.

b denotes the distance between the hyperplane and the origin.

Now it's required and necessary to ensure that all the inputs of these positive points above the hyperplane can give positive results, while all the negative ones below the hyperplane correspond to negative results.

$$\omega^T X_i + b \geq 1, y_i = 1$$

$$\omega^T X_i + b \leq -1, y_i = -1$$

Evidently, the target of SVM is to find out the minimum value of 0.5 $|\omega|^2$ so that

$$y_i(\omega^T X_i + b) \geq 1$$

We can convert the target above by using Lagrange.

$$\text{L} = 0.5|\omega|^2 + \sum_i^n l_i(1 - y_i(\omega^T X_i + b))$$

This can be solved using Quadratic programming theory in the optimization field. [91]

3.4 Decision Tree

Decision Tree is a tree structure. In a decision tree, every non-leave node can represent a test or a divergence using one of the features of this data set, while every leave node represents a result of classification. When we need to utilize decision tree to help us make decisions and classify data, the process is to classify a specific data item based on the conditions on the non-leave nodes, until we get the class that this data item belongs to. [92]

In fact, the structure of decision tree makes it possible for us to form a series of rules that may help us to classify some unknown data in the future. In other words, its feature of Divide and Conquer makes it an algorithm with strong ability in dealing with sophisticated cases. [93]

The procedure of induction of rules can be well shown using the figure below.

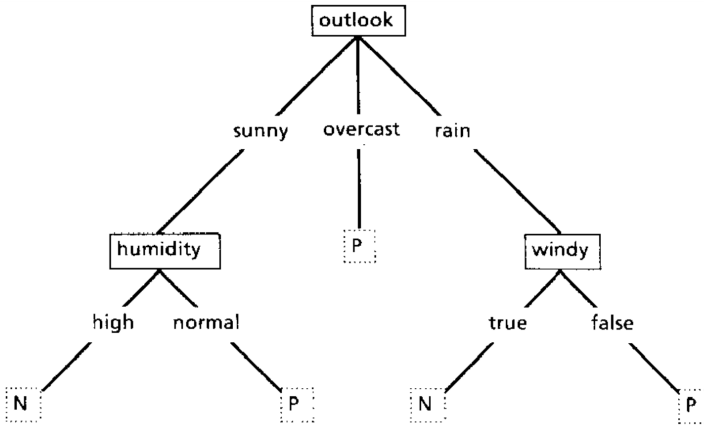


Figure 3.4: Simple Decision Tree Induction [1]

A more complex process of induction of Decision Tree can be shown in the Figure 3.5.

Both the inductions in Figure 3.4 and Figure 3.5 describe the same problem. Saying that the objects are Saturday mornings and their aims are to classify them. Viewing that there are four available features including outlook, temperature, humidity and windy, thus rules are inducted to construct the classification model. [1] In this case, the relatively complex induction model in Figure 3.5 performs much better in the aspect of interpretability of the training data, which is also more likely to improve the robustness and accuracy of the decision tree.

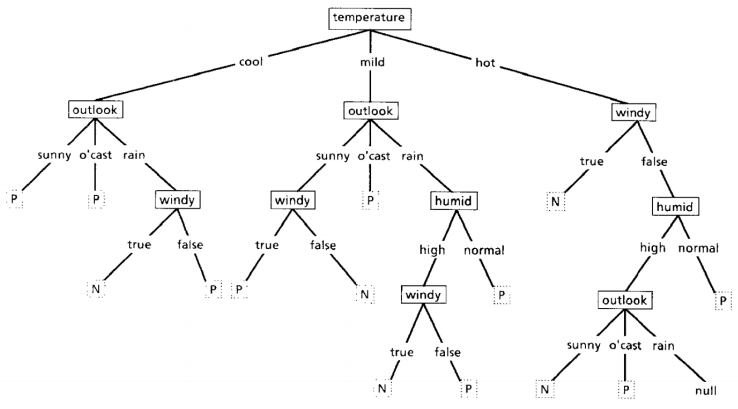


Figure 3.5: Complex Decision Tree Induction [1]

Obviously, the most significant and difficult thing in Decision Tree is how to select the most appropriate features for the tree. Because of that, we'll introduce the concept of Information Entropy.

Saying that in the data set D, there are T different types of data in total.

$$\text{Entropy(D)} = -\sum_{i=1}^{T} p_i(log_2 p_i)$$

where Entropy(D) is the information entropy of the data set D

p_i is the percentage of the i-th type in the T types [94]

When we suppose that the feature f has O different possible options of values. Based on the definition of Information Entropy, we can get the concept of Information Gain, which can be represented as follow.

$$\text{Gain(D,f)} = -\sum_{o=1}^{O} p_i|\frac{D^f}{D}|Entropy(D^f) + \text{Entropy(D)}$$

Gain(D,f) denotes the information gain when we divide the data set D using feature f. Therefore, for all the features that remain to be selected, we should choose the one that has the maximum information gain. [94]

However, evidently this method of choosing the feature can lead to some problems, because this method inclines to choose those features with more options of values. Thus, there's a variant of decision tree. This variant utilizes Gain Ratio as the index for choosing the most suitable features.

$$\text{GR(D,f)} = \frac{-\sum_{o=1}^{O} p_i|\frac{D^f}{D}|Entropy(D^f)+Entropy(D)}{-\sum_{o=1}^{O} p_i|\frac{D^f}{D}|Entropy(D^f)}$$

3.5 k-Nearest Neighbors

k-Nearest Neighbors (kNN) is a widely used supervised machine learning algorithm. Its mechanism is to deal with the data based on the distance between each points in the high-dimension data. In other words, for a specific point in the data, this algorithm will find out k nearest neighbors concerning this point. The core idea of kNN is that if most data in this k nearest neighbors belong to a specific class, then all the data in this kNN can also be represented using this specific class. It's obvious that a closer data can have a bigger weight and thus have a higher influence on the results. [95]

Unlike most machine learning algorithms, kNN doesn't have a very explicit training process. For example, if we want to use SVM algorithm, we must firstly train our model on the training data set, then apply the algorithm upon the test data set. However, in kNN, we store these data until we need to test and then start to calculate these data. [96]

It will not be too difficult to realize that the value of k can be very determinant in the algorithm of kNN. Different values of k may lead to somehow different results. Thus, we should be very cautious in determining a proper k value. For instance, in the figure below, we have a binary classification problem. Apparently, in this example, the values of k can affect the results greatly.

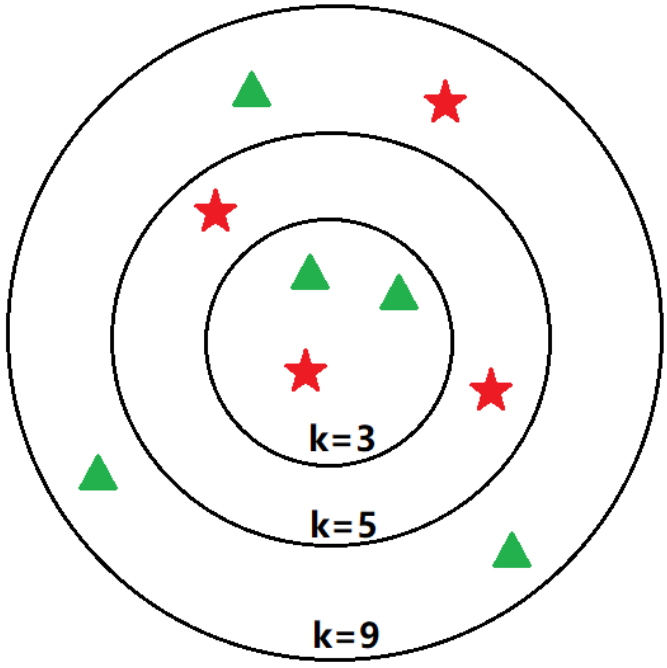


Figure 3.6: kNN with different values of k

Usually, we use two approaches to measure the distances. The first measure metric is Euclidean Distance, which can be represented as below. [97]

$$\text{EuclideanDistance(x,y)} = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The second approach is Manhattan Distance, which can be denoted like this.

$$\text{ManhattanDistance(x,y)} = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sqrt{|x_i - x_j| + |y_i - y_j|}$$

3.6 Naive Bayes

Naive Bayes is based on Bayesian Theory. Thus, we need to talk about Bayesian decision theory first. Briefly, Bayesian theory studies how to label different classes as accurate as possible when all the related possibilities have already been known.

Saying we have N types of different labels. C = $\{c_1, c_2, \cdots, c_N\}$, then we can have:

$$R(c_i|x) = \sum_{j=1}^{N} l_{ij} P(c_j|x)$$

where l_{ij} is the losses when we wrongly label a c_j class as a c_i class

P($c_i|x$) denotes the expected loss when we classify x as the c_j class

In this case, it's safe to come to the Bayesian Decision Rule: we should select the class label that can make the risk on this sample as low as possible so that we can ensure the total risk can have the lowest value. [98]

$$h(x) = \underset{c \in C}{argmin} R(c|x)$$

where h(x) would be referred as Bayes Optimal Classifier.

$R(c|x)$ is referred as Bayes Risk

If we define l_{ij}=0 (i=j), l_{ij}=1 (i≠j). We can now have this equation. [99]

$$h(x) = \underset{c \in C}{argmax} P(c|x)$$

According to the Bayes Theorem, we can get this equation.

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)}$$

When we think twice about the equation above, we can find the tremendous difficulty in figuring out the value of $P(x|c)$, which can be very hard if the volume of data set is relatively limited. Therefore, we need to introduce Naive Bayes, which has the attribute conditional Independence assumption. Based on this assumption, we can much calculate it much more easily.

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)} = \frac{P(c)}{P(x)} \prod_{i=1}^{f} P(x_i|c)$$

where f stands for the number features

In fact, there exists many other variants of Naive Bayes. Among them, one of the most popular classification method is called Gaussian Naive Bayes. Compared with Naive Bayes, Gaussian Naive Bayes makes an assumption that the likelihood of each feature obeys the rule of Normal Distribution, which can be represented as function below. [100]

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp(-\frac{(x_i - \mu_y)^2}{2\pi\sigma_y^2})$$

3.7 Neural Networks

Neural Networks are inspired from the biological structure of neurons. In biological neural networks, each neuron is connected with other neurons. When this neuron is activated, it will send chemical materials to its neighbors' neurons. Because of that, the voltage in those near neurons can be changed. There exists a very significant concept called threshold. Whenever the voltage in a neuron is modified, we need to make a judgment whether the current voltage is above the threshold or not. neurons in the neighbor will be activated if the voltage is over the threshold. Otherwise, it won't be activated.

In 1943, McCulloch and Pitts constructed a novel model, which was called MP model. The core idea of this model is based on the abstraction of biological neural networks mentioned in the last paragraph. For example, saying a neuron will take in the chemical signals from m neurons in the neighbors. [101]

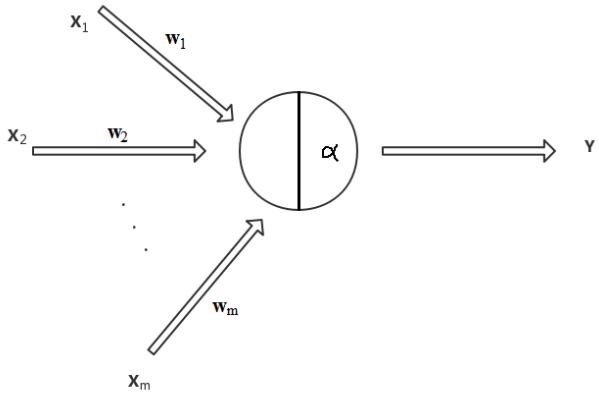


Figure 3.7: MP Neuron Model

In the figure 3.5, we can notice that those signals from other neurons will be imported through the connection with various weights. The total input will be compared with the threshold of this neuron to determine the state of this neuron. We have mentioned Sigmoid function in Logistic Regression algorithm. Similarly, we'll also adopt Sigmoid function here, because it's able to map the output into the scope from 0 to 1.

The expression of Sigmoid function is $\sigma(x) = \frac{1}{1+e^{-x}}$. The figure of Sigmoid function can be shown below. [102]

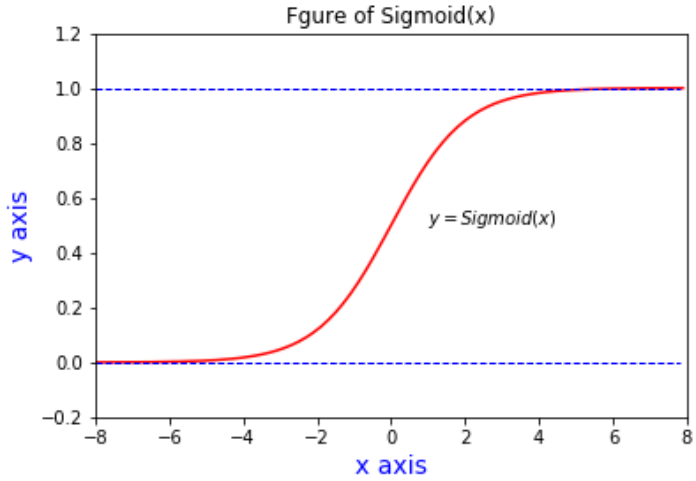


Figure 3.8: figure of Sigmoid function

3.7.1 Multilayer Perceptron (MLP)

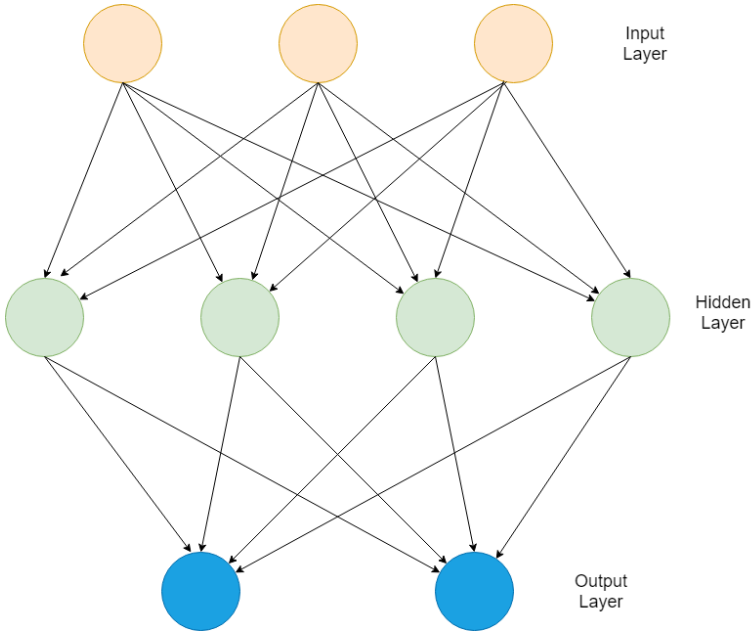


Figure 3.9: figure of MLP

In any MLP, it must have an input layer and an output layer. While it can have multiple hidden layers. [103] In the figure 3.7, we take the simplest form of MLP, which only has only one hidden layer. In MLP, the relationship between layers is fully connected.

In this case, if we assume the input is a vector represented as X. Then the input to the hidden layer is $(\omega_1 X + b_1)$, if the function of hidden layer is a sigmoid function, then the result of hidden layer is $Sigmoid(\omega_1 X + b_1)$, where $Sigmoid(x) = \frac{1}{1+e^{-x}}$. Then the result of the output layer is

$$f(x) = Softmax(\omega_2 \frac{1}{1+e^{-(\omega_1 X + b_1)}} + b_2)$$

MLP is a relatively old version of applied neural network. MLP is quite easy to construct and implement when compared with other neural networks. However, it still has some noticeable disadvantages.

Firstly, since the MLP adopts the full connection, which could lead to an explosion in the calculation of the parameters of the neural networks. Of course, the speed in the training of neural network would be much slower. Secondly, if there are too many hidden layers in the MLP, it's likely the problem of gradient vanishing would be very terrible, which would make it almost impossible for us to train the model. [104]

3.7.2 Convolution Neural Networks (CNN)

When we treat MLP as a traditional and classical model for neural networks, we can also view CNN as an advanced variant of neural networks. Compared with MLP, CNN introduced two new approaches: convolutional layer and pooling layer. [105]

By the introduction of convolutional layer and kernel, we are able to extract some deeper features behind the data. For example, if we use MLP to deal with the problems of image classification, we can find that the precision is not so good. This is because MLP is not good at understanding the distortions and scales of the data. While at the same time, the introduction of kernel makes the model to classify those images more robustly. [106] In other words, CNN is more advantageous in handling spatial data. Besides, the pooling layer can efficiently reduce the dimension of the data, which can not only reduce the work of calculation, but also lessen the possible vanishing gradient problem.

3.8 Ensemble Learning

Ensemble learning is a supervised learning method, which combines different learning approaches to get the optimal results. Those base learners in the ensemble learning can be homogeneous or heterogeneous. [107]

3.8.1 Bagging

Bagging is based on the method of Bootstrap Sampling. Saying that there's a data set containing m items. Bootstrap Sampling is a sampling method that will choose one item as a part of the result sampling volume. This item will be put back into the original data set so that it can be chosen from in this next choosing. When the number of items in the result sampling volume equals m, we stop this time of choosing. After repeating the process for N times, we can get N different result sampling volume that each contains m items. We firstly train N different models on these N different volume, then we combine these N base learners. [108]

3.8.2 Random Forest

Random Forest is a variant of Bagging. Apart from building Bagging ensemble based on the decision tree, random forest introduces some random selection in the training process. Saying there are M features for a specific node. Traditional decision trees will directly choose the feature that could bring in the optimal result at this time. While in random forest, we'll randomly select m ($m < M$) features and choose optimal feature based on this case. After this procedure is repeated for N times, we can get the optimal learner based on these N times. [109]

3.8.3 AdaBoost

AdaBoost is an approach that enables a weak base learner to become a strong base learner. Its core idea is to firstly train a weak base learner. In next round of training, the data that is wrongly classified will be put on higher weights. Based on the base learner from last training round and these weighted data, we can get a new base learner. This process will be repeated until we get N base learners in total. The last step is to get the weighted sum of all these N base learners. [110]

3.9 Clustering

Clustering belongs to unsupervised learning. Unlike supervised machine learning methods, such as SVM and kNN, which need predefined labels to all the training data, clustering doesn't need any classification to the training data. Instead, clustering methods try to classify similar data into distinct parts based on the core rules and features behind those data. [111] The most predominant point that distinguishes Clustering with other methods is that clustering deals with unknown classification. For example, in social network analysis, we have no idea what classes we'll get before we get the results from the clustering. [112]

3.9.1 k-Means

Saying we've gotten a data set D = $\{x_1, x_2, \cdots, x_n\}$. At this time, k-Means forms k different clusters, which can be represented as C = $\{C_1, C_2, \cdots, C_k\}$. Evidently, we can have the equation below:

$$\text{E} = \sum_{x \in C_1} |x - \mu_1|^2 + \cdots + \sum_{x \in C_k} |x - \mu_k|^2$$

$$\text{where } \mu_i = \frac{\sum_{x \in C_1}^{x}}{|C_i|}$$

Generally speaking, we generate those k clusters using the following steps. Firstly, we randomly choose k data from the data set D. Secondly, we assign the remaining data to those k different classes based on their distances to their own nearest clusters. Thirdly, after all data have been assigned to a specific cluster, we figure out the cluster center of each cluster. Now that we have gotten new k different cluster centers, we'll repeat the three steps above until the results satisfy our requirements. [113]

3.10 Geographic information system (GIS)

GIS is shorted for Geographic Information System, which is mainly used for handling various spatial information. GIS has its unique advantages in capturing, storing, manipulating, analyzing, managing, and presenting spatial or geographic data. [114] Compared with other data and traditional statistical approaches of data analysis, the inherent features like latitude and longitude in the GIS data make it difficult to deal with them using traditional statistical methods and common visualization tools.

As for this part of visualization concerning map and geographic information, even though uncounted tools and software are available, we recommend and choose to utilize the Google Map and Google Chart. This decision is based on these considerations as below: Firstly, Google Chart is based on HTML and JavaScript, which gives the Google Chart the features of functioning in cross platforms, scalability. [115] Secondly, in the chart gallery of Google Chart, users can quickly have a glimpse of all different kinds of charts and choose the best template for visualization. Last but not least, Google chart has rich and accurate data about geographic and spatial information. Those information can greatly save the users from overwhelming work in hard-code of spatial information.

Chapter 4

Experiment

In this master thesis, our target is to construct different models for the novel data set of the bankruptcy prediction of Norwegian companies. Undoubtedly, we also need to find out the optimal model for this problem and give some analysis. Thus, we can throw some light upon further researches.

4.1 Experiment Setup

Experiments are mainly conducted on the platform of Anaconda and Jupyter notebook (version 5.5.0). The main language in this project is Python. Besides, we also write some Visual Basic scripts to deal with the batch processing in Microsoft Excel. Additionally, JavaScript is utilized, because the GeoChart of Google Map requires us to use JavaScript scripts to make visualization based on map.

In our process, some Python packages, including Numpy (version 1.14.3), Pandas (version 0.23.0), Matplotlib (version 2.2.2), Seaborn (version 0.8.1), Scikit-learn (version 0.19.1) are imported and utilized.

4.2 Hardware

All the procedure of data pre-processing and experiments are conducted on the ASUS G752VM with the 2.6GHz Intel(R) Core(TM) i7-6700HQ CPU, 8.00GB RAM, 6.00 GB NVIDIA Geforce GTX 1060.

4.3 Data Description and data pre-processing

Our source data is consisted of three files: 1. *Konkurs statistikk 2016.csv*, which includes all the companies that went bankruptcy in 2016. 2. *Konkurs statistikk 2016.csv*, which includes all the companies that went bankruptcy in 2018. 3. *Nyetablerte siste 3ar etc.txt*, which includes all the companies that functioned well from 2016 to 2018.

Because we have only gotten the data of all the companies that went bankruptcy in 2016 and 2018, what we need to do is to filter out the data of the companies that functioned well in 2016 and 2018. Thus, we firstly convert the file of *Nyetablerte siste 3ar etc.txt* into *Nyetablerte siste 3ar etc.csv*. Secondly, we make the file of *Non-bankruptcy companies 2016.csv* and *Non-bankruptcy companies 2018.csv* based on the file *Nyetablerte siste 3ar etc.csv*.

Now it comes to the step of feature selection. However, we must point out that some features are purely categorical and somehow meaningless. For example, in the file of *Non-bankruptcy companies 2016.csv*, its features such as *orgnr*, which stands for the organization number, *DUNS NUMBER*, which is a nine-bit number to distinguish one company from others in the financial database should not be considered as features for further steps. Similarly, some information concerning pure text contents, including *Company name*, *BesøksAdresse* and *BesøksPoststed* should also be thrown away.

In the end, we choose eight parameters as the features of our model: *Registration_Month*, which denotes the month of this company comes into the record. *Bransje*, which denotes the specific industry that this company belongs to. *Fylke*, which denotes the county where this company lies. *Kommune*, which denotes the smaller community this company sits. *Stiftet*, which denotes the establishment year of this company. *Share_Capital*, which denotes the total registration capital when this company is established. *Organization_Form*, which denotes the form of this company. *Ansatte*, which denotes the number of employees in this company.

There are some rows with some features being blank. In this project, we choose to throw away the rows whose information can be partly missing. Besides, in the Organization_Form column, because there are only around 23 different kinds, thus we replace the names of the different types with numbers using Regular Expression. This is implemented using the code below.

```
result = df['Organization_Form'].replace(regex={'ESEK':'1',
'TVAM':'2','ANS':'3','ASA':'4','BBL':'5','BRL':'6',
'ENK':'7','FLI':'8','IKS':'9','KBO':'10','NUF':'11',
'PRE':'12','STI':'13','SAER':'14','AS':'15','BA':'16',
'DA':'17','KF':'18','KS':'19','SA':'20','SE':'21',
'PK':'22','SF':23,
})
```

4.4 Experiment Procedure

4.4.1 Visualization of the data

In the visualization part, our work can be divided into two parts. In the first part, we firstly explore our data using various tools, such as Heat Map, bar chart, bubble charts. This part is put on the first part in the experiment part, because these visualization methods can give us a very explicit and intuitive understanding about the data.

After that, we then seek to visualize the data using tools like GeoChart, which is an API offered by Google Map. The Geochart presents three different possible modes for the visualization based on a map of a country, or a region.

4.4.1.1 Heatmap

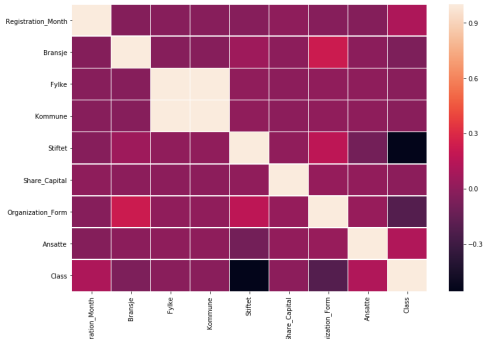


Figure 4.1: heatMap of 2016 data

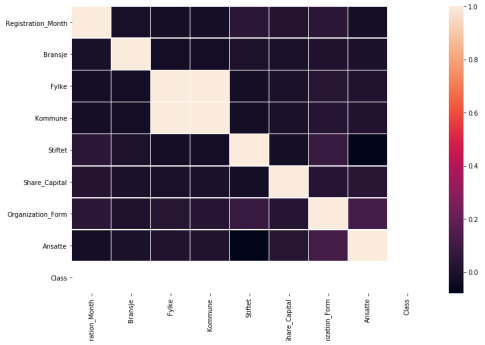


Figure 4.2: heatMap of 2016 bankruptcy data

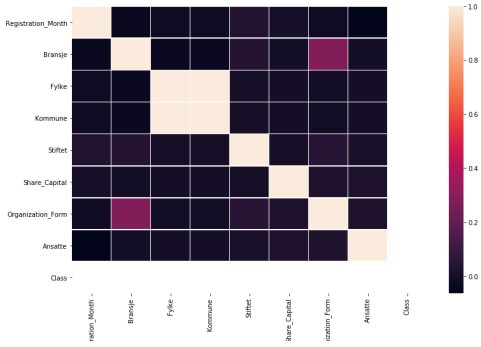


Figure 4.3: heatMap of 2016 non-bankruptcy data

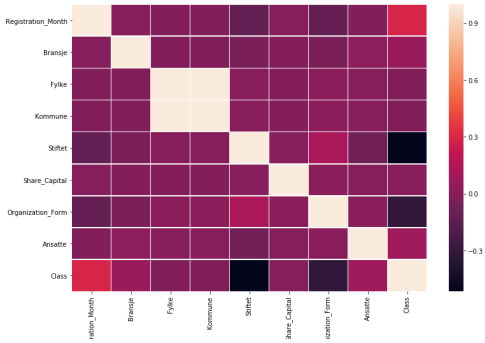


Figure 4.4: heatMap of 2018 data

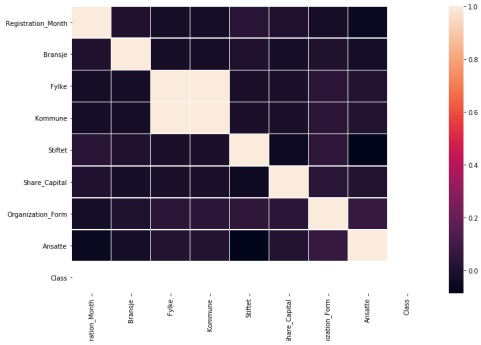


Figure 4.5: heatMap of 2018 bankruptcy data

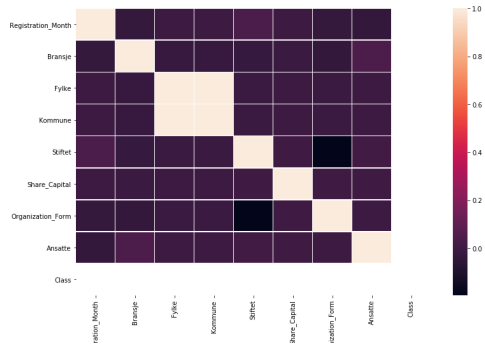


Figure 4.6: heatMap of 2018 non-bankruptcy data

4.4.1.2 Boxplot

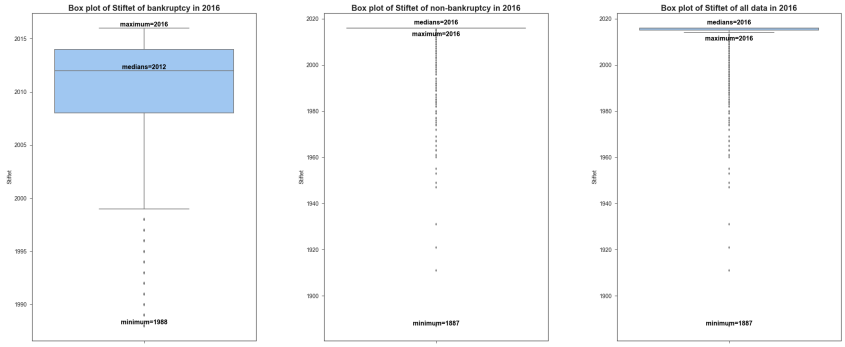


Figure 4.7: Box plots of the *Stiftet* of data in 2016

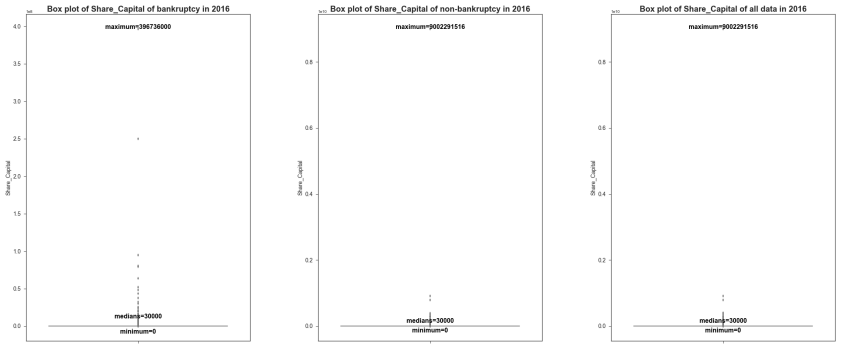


Figure 4.8: Box plots of the *Share_Capital* of data in 2016

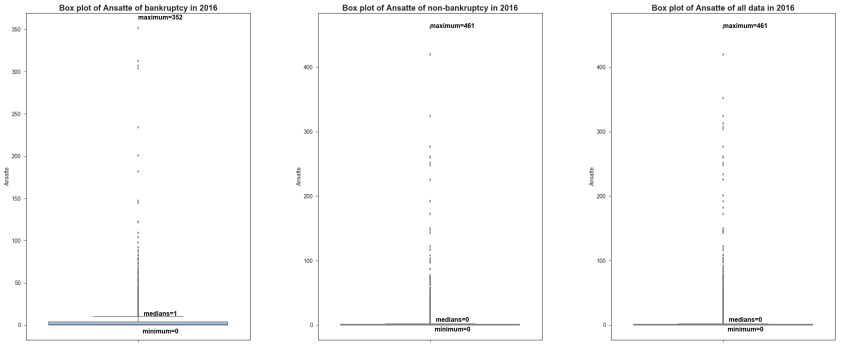


Figure 4.9: Box plots of the *Ansatte* of data in 2016

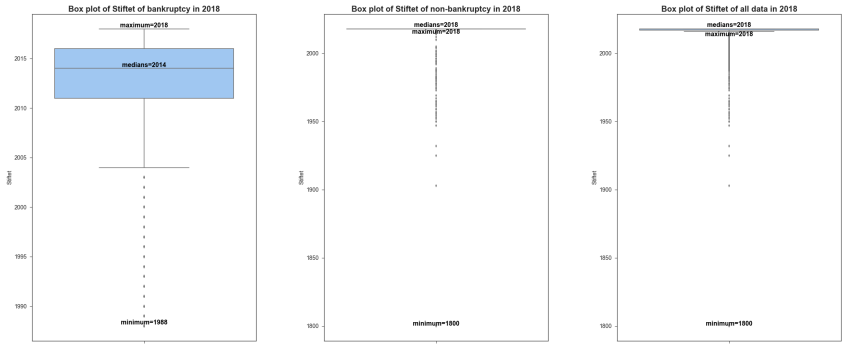


Figure 4.10: Box plots of the *Stiftet* of data in 2018

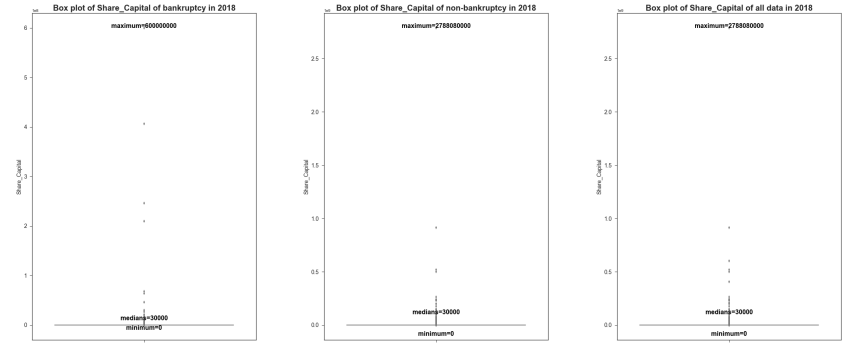


Figure 4.11: Box plots of the *Share_Capital* of data in 2018

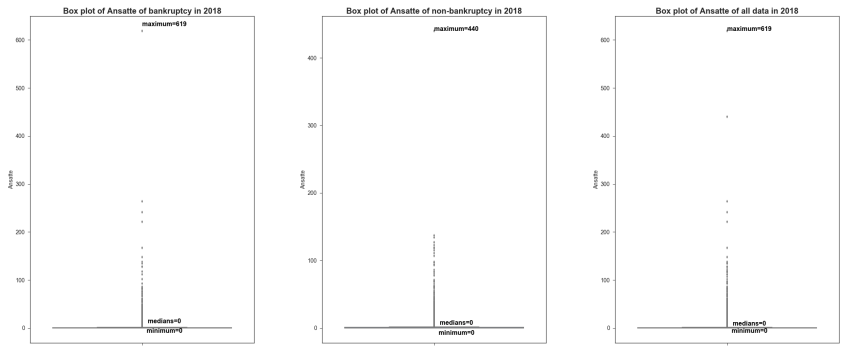


Figure 4.12: Box plots of the *Ansatte* of data in 2018

4.4.1.3 Bubble Chart

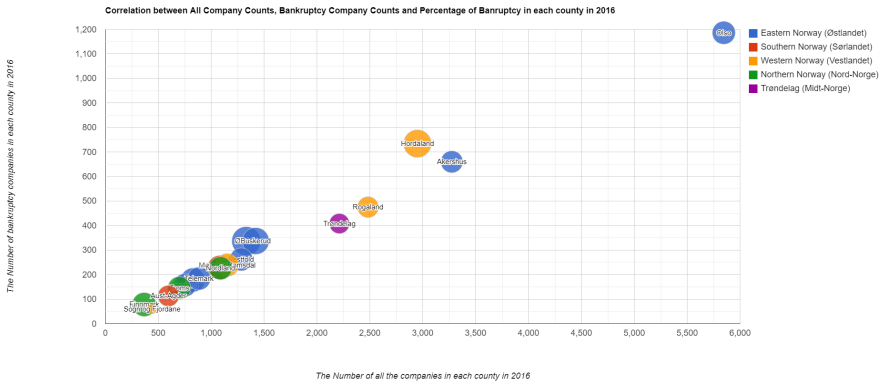


Figure 4.13: Bubble Chart visualization of the data in 2016

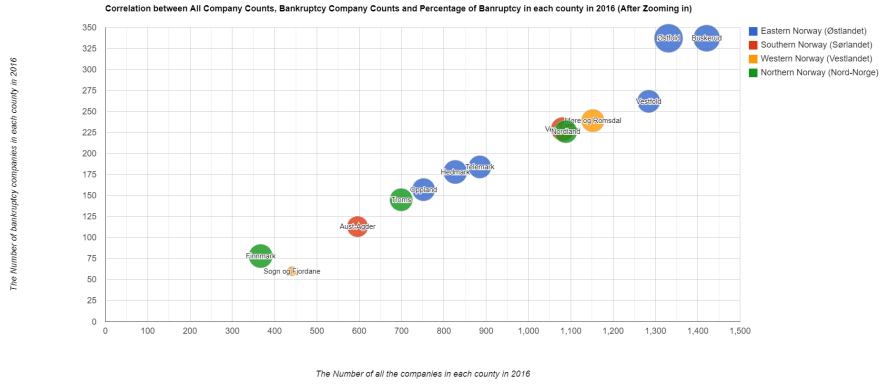


Figure 4.14: Bubble Chart visualization of the data in 2016 (After Zooming In)

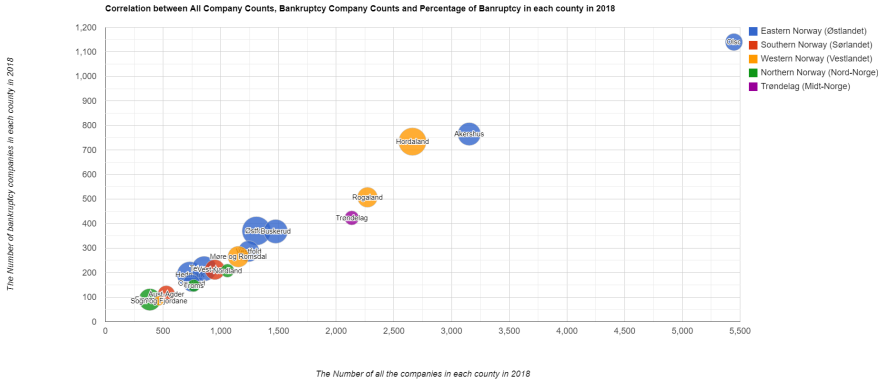


Figure 4.15: Bubble Chart visualization of the data in 2018

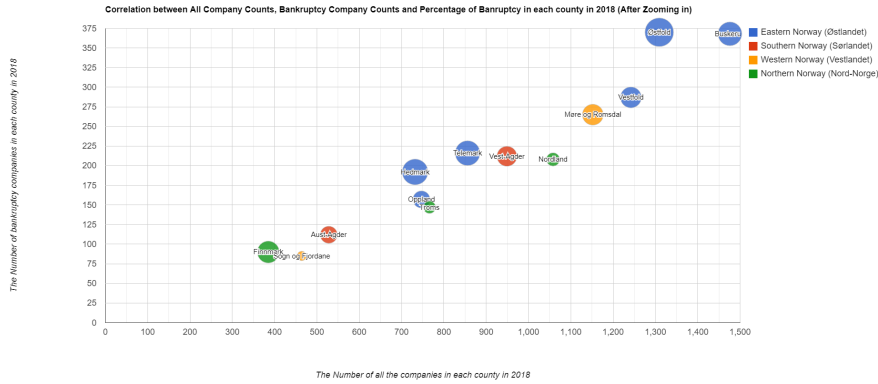


Figure 4.16: Bubble Chart visualization of the data in 2018 (After Zooming In)

4.4.1.4 Bar Chart

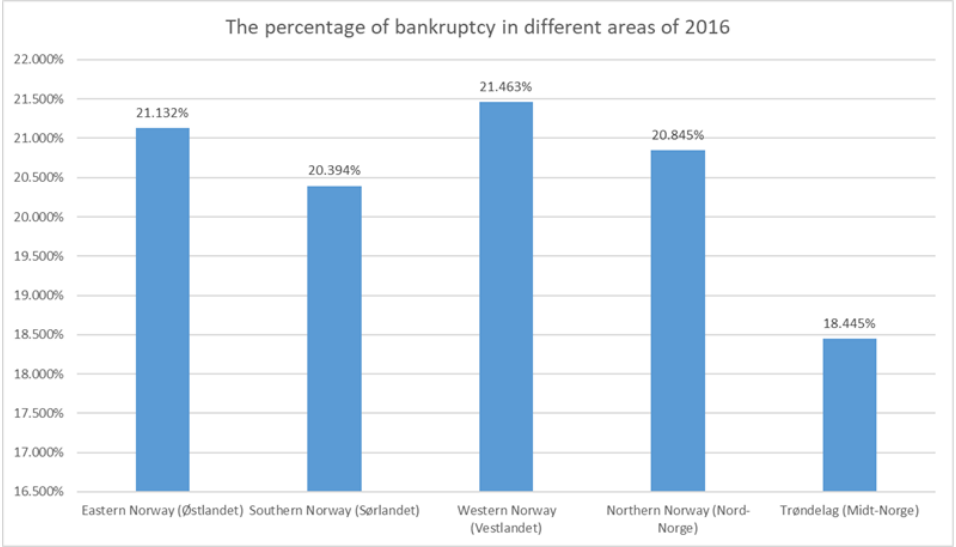


Figure 4.17: Bar Chart visualization of the data in 2016

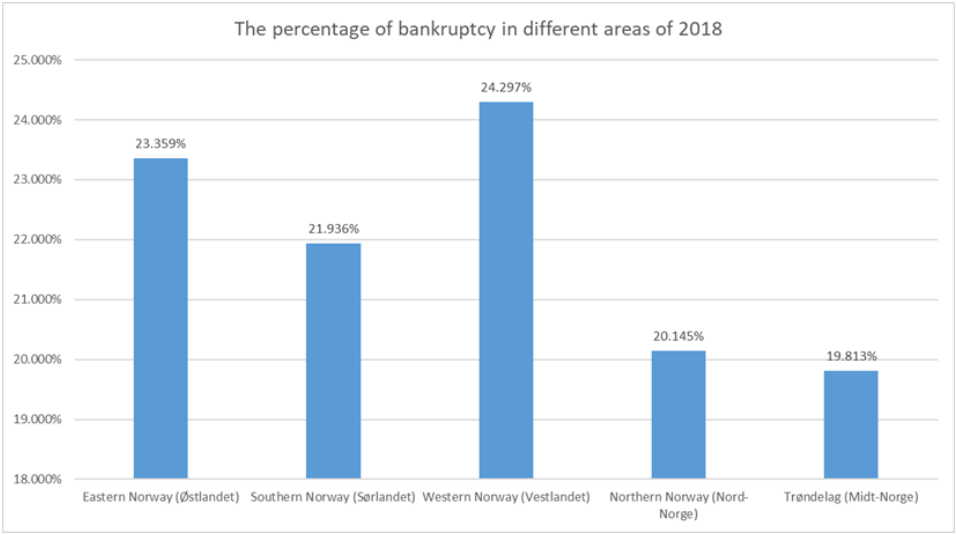


Figure 4.18: Bar Chart visualization of the data in 2018

4.4.1.5 Visualization based on Map

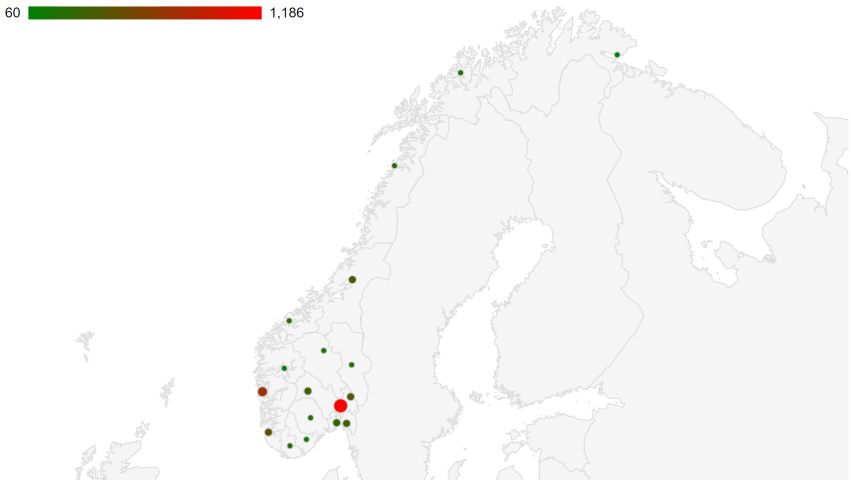


Figure 4.19: Marker map visualization of the number of bankruptcy in 2016

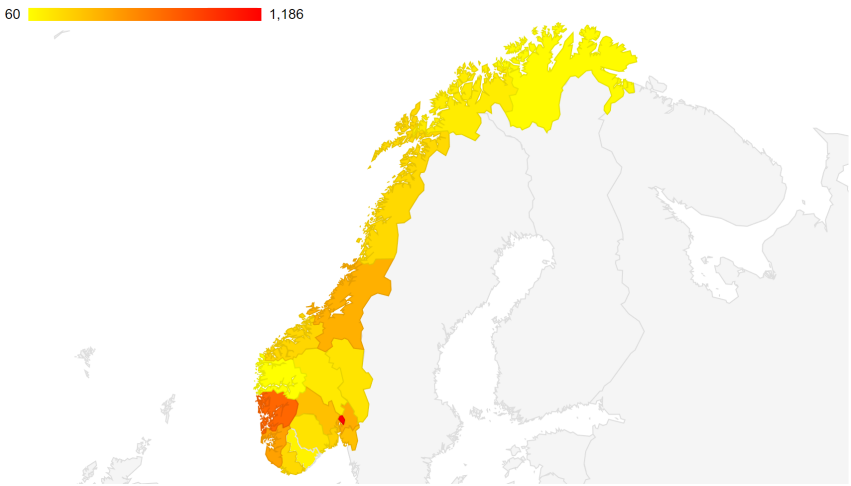


Figure 4.20: Region map visualization of the number of bankruptcy in 2016

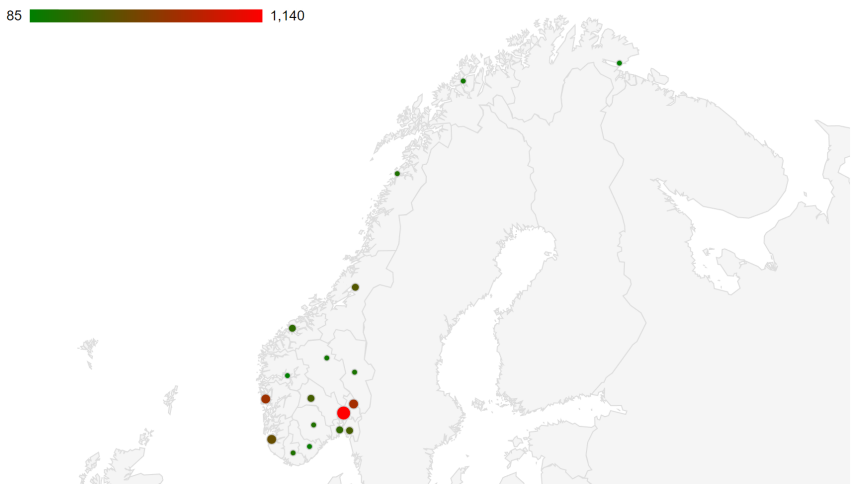


Figure 4.21: Marker map visualization of the number of bankruptcy in 2018

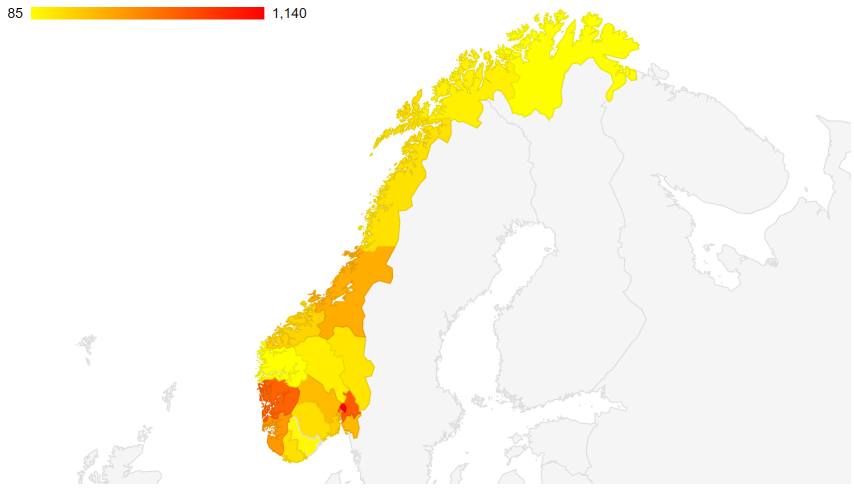


Figure 4.22: Region map visualization of the number of bankruptcy in 2018

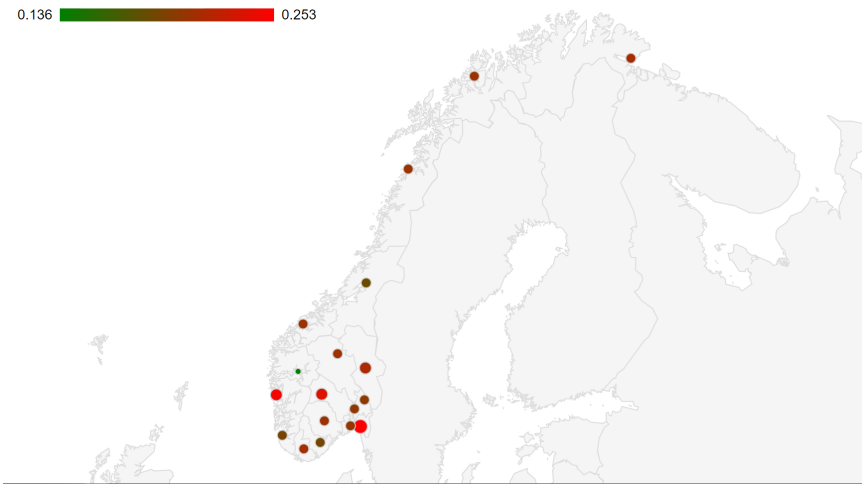


Figure 4.23: Marker map visualization of the percentage of bankruptcy in 2016

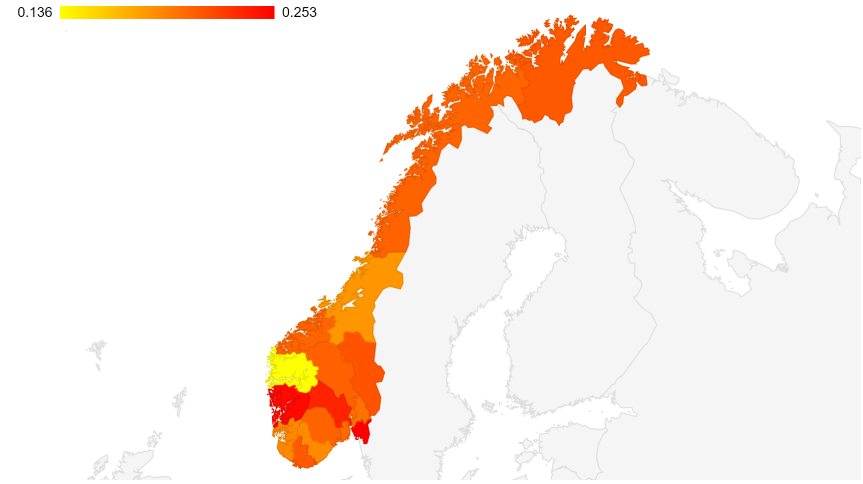


Figure 4.24: Region map visualization of the percentage of bankruptcy in 2016

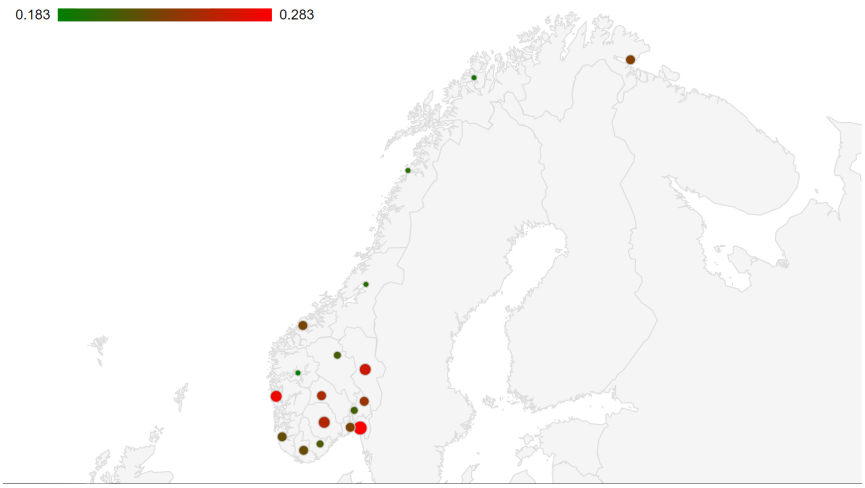


Figure 4.25: Marker map visualization of the percentage of bankruptcy in 2018

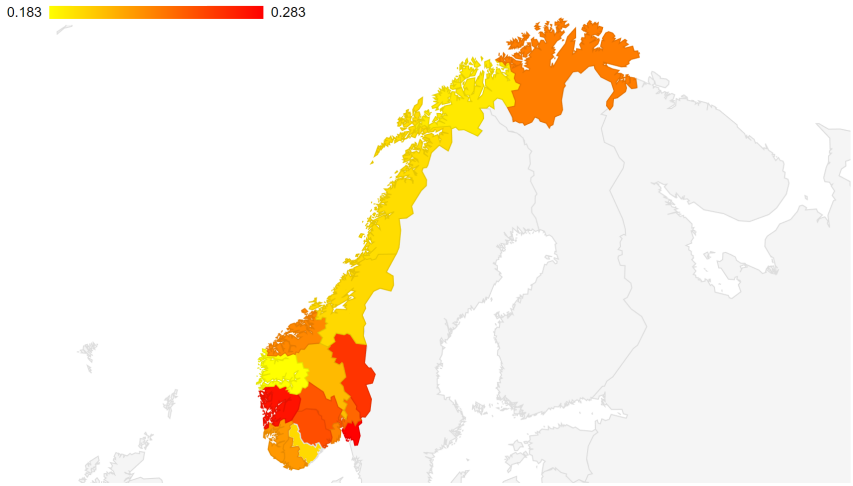


Figure 4.26: Region map visualization of the percentage of bankruptcy in 2018

4.4.2 Single Classifier

4.4.2.1 Logistic Regression

As we have mentioned in Section 3.1, Logistic Regression is a very popular algorithm in the binary classification. Here we test the performance of Logistic Regression on the classification of bankruptcy companies in the data set of 2016 and 2018 respectively. Its results can be shown in Figure .

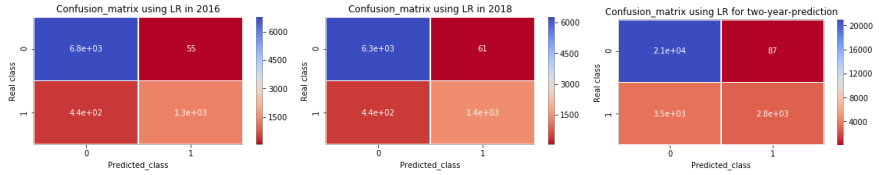


Figure 4.27: Confusion matrices of the data using *LR (Logistic Regression)*

4.4.2.2 Linear Discriminant Analysis

Linear Discriminant Analysis can be used for binary classification. However, it's not so suitable for unbalanced data set.

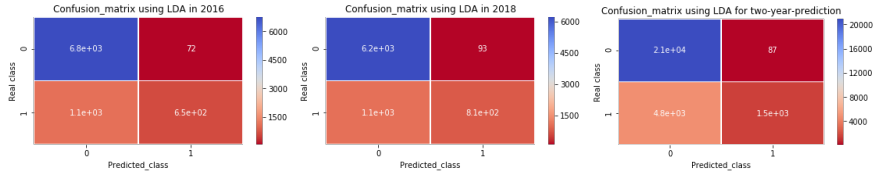


Figure 4.28: Confusion matrices of the data using *LDA (Linear Discriminant Analysis)*

4.4.2.3 Support Vector Machine

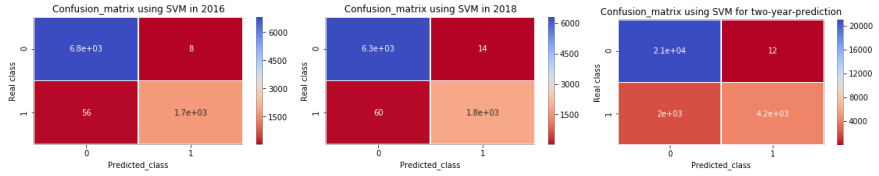


Figure 4.29: Confusion matrices of the data using *SVM (Support Vector Machine)*

4.4.2.4 SGD

This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule. [116]

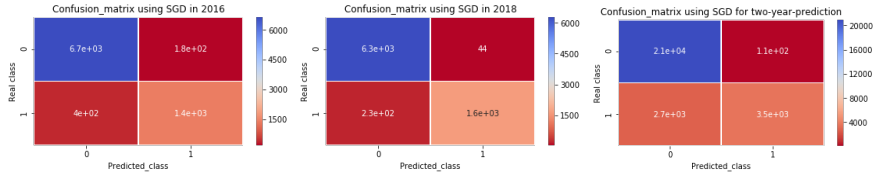


Figure 4.30: Confusion matrices of the data using *SGD*

4.4.2.5 Decision Tree

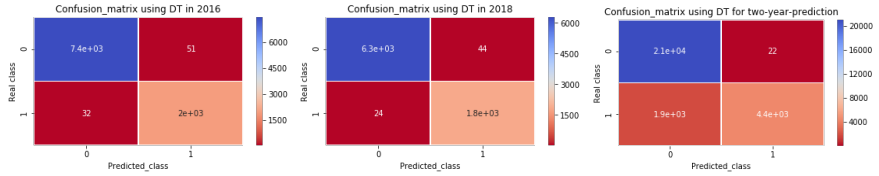


Figure 4.31: Confusion matrices of the data using *DT (Decision Tree)*

4.4.2.6 k-Nearest Neighbors

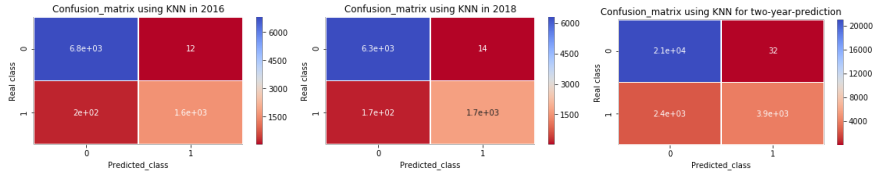


Figure 4.32: Confusion matrices of the data using *KNN (k-Nearest Neighbors)*

4.4.2.7 k-Means

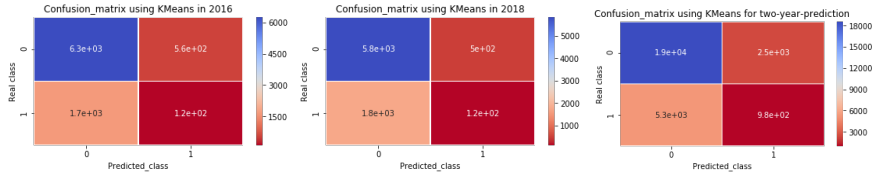


Figure 4.33: Confusion matrices of the data using *KMeans (k-Means)*

4.4.2.8 Naive Bayes

Here we use Gaussian Naive Bayes.

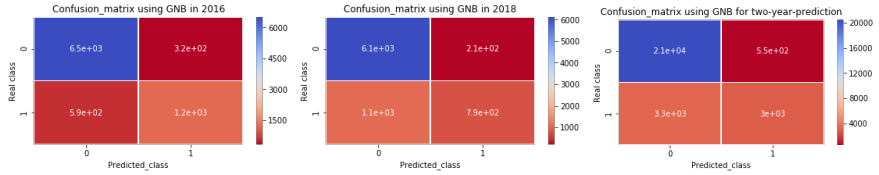


Figure 4.34: Confusion matrices of the data using *GNB (Gaussian Naive Bayes)*

4.4.2.9 MLP

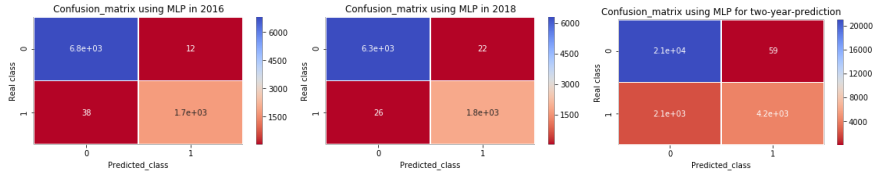


Figure 4.35: Confusion matrices of the data using *MLP (Multilayer Perceptron)*

4.4.3 Multiple Classifiers

Instead of simply applying different single classifiers to deal with the prediction of bankruptcy, Here we also intend to predict the case of the bankruptcy prediction of Norwegian companies of 2018, using models we construct from the data of 2016 and the method of ensemble learning.

4.4.3.1 Ensemble learning using Gradient Boosting & Random Forest

Because both Gradient Boosting and Random Forest are based on the method of the Decision Tree. Therefore, we compare the results of two-year-prediction using these two methods.

Figure 4.36: Confusion matrices of two-year-prediction using *Gradient Boosting* and *Random Forest*

4.4.3.2 Ensemble learning using Bagging

In this section, we have done several experiments by combining Bagging and other single classifiers.

The following combinations will be tested in this part: Decision Tree + Bagging, Gaussian Naive Bayes + Bagging, K Nearest Neighbor + Bagging, Linear Discriminant Analysis + Bagging, Logistic Regression + Bagging, Multi Layer Perceptron + Bagging, Stochastic Gradient Descent + Bagging, Support Vector Machine + Bagging, Random Forest + Bagging, Gradient Boosting + Bagging.

4.4.3.3 Ensemble learning using Majority Voting

Considering the fact that Majority Voting will utilize several classifiers at the same time and form the optimal model, thus we'll put several classifiers each time and try to find out the best prediction model.

4.5 Experiment Results

4.5.1 Single Classifier

2016-Single-Classifier Table					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
KMeans	6.439%	17.062%	91.816%	78.960%	65.646%
LDA	36.338%	90.014%	98.946%	85.598%	76.511%
GNB	66.853%	79.073%	95.373%	91.669%	86.685%
LR	75.252%	96.069%	99.195%	93.876%	90.181%
SGD	77.436%	88.654%	97.408%	94.289%	90.884%
KNN	88.858%	99.250%	99.824%	97.164%	95.479%
SVM	96.865%	99.540%	99.883%	99.186%	98.707%
DT	98.388%	97.455%	99.319%	99.572%	99.323%
MLP	97.872%	99.318%	99.824%	99.446%	99.121%

Table 4.1: 2016-single-classifier

2018-Single-Classifier Table					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
KMeans	6.247%	19.086%	92.164%	76.864%	62.887%
LDA	43.193%	89.690%	98.531%	85.427%	76.804%
GNB	42.338%	79.063%	96.682%	85.000%	76.193%
LR	76.615%	95.922%	99.036%	93.470%	89.790%
SGD	87.827%	97.395%	99.305%	96.500%	94.563%
KNN	90.710%	99.183%	99.779%	97.319%	95.839%
SVM	96.797%	99.234%	99.779%	99.059%	98.545%
DT	98.719%	97.676%	99.305%	99.620%	99.413%
MLP	98.612%	98.823%	99.652%	99.652%	99.366%

Table 4.2: 2018-single-classifier

Two-Year-Prediction-Single-Classifier Table					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
KMeans	15.674%	28.423%	88.236%	77.831%	64.892%
LDA	23.989%	94.539%	99.587%	81.467%	70.200%
GNB	46.989%	84.262%	97.384%	86.041%	77.901%
LR	44.043%	96.950%	99.587%	85.655%	77.165%
SGD	56.260%	96.953%	99.473%	88.413%	81.679%
KNN	62.297%	99.188%	99.848%	89.884%	84.048%
SVM	67.474%	99.718%	99.943%	91.158%	86.096%
DT	70.277%	99.504%	99.896%	91.854%	87.213%
MLP	67.076%	98.618%	99.720%	91.041%	85.915%

Table 4.3: Two-Year-Prediction-single-classifier

4.5.2 Ensemble Learning & Multiple Classifiers

4.5.2.1 Gradient Boosting & Random Forest

Two-year-prediction using Gradient Boosting and Random Forest					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
Gradient Boosting	67.203%	99.953%	99.991%	91.095%	85.993%
Random Forest	71.886%	99.911%	99.981%	92.267%	87.871%

Table 4.4: two-year-prediction using Gradient Boosting and Random Forest

4.5.2.2 Bagging Ensemble

Two-year-prediction using Bagging Ensemble and other methods					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
LDA+Bagging	24.036%	94.490%	99.582%	81.476%	70.215%
GNB+Bagging	44.744%	84.380%	97.531%	85.554%	77.096%
LR+Bagging	44.090%	96.953%	99.587%	85.666%	77.182%
SGD+Bagging	71.567%	99.933%	99.986%	92.187%	87.742%
KNN+Bagging	62.313%	99.189%	99.848%	89.888%	84.054%
SVM+Bagging	67.522%	99.694%	99.938%	91.169%	86.114%
DT+Bagging	71.376%	99.933%	99.986%	92.138%	87.665%
MLP+Bagging	67.283%	98.923%	99.782%	91.097%	86.003%
Gradient Boost+Bagging	70.086%	99.977%	99.995%	91.814%	87.145%
Random Forest+Bagging	71.567%	99.956%	99.991%	92.187%	87.742%

Table 4.5: two-year-prediction using Gradient Boosting and Random Forest

4.5.2.3 Majority Voting Ensemble

Two-year-prediction using Majority Voting and other methods					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
DT+GNB+Majority Voting	70.277%	99.571%	99.910%	91.855%	87.214%
DT+KNN+Majority Voting	65.833%	99.831%	99.967%	90.755%	85.448%
DT+LDA+Majority Voting	70.389%	99.505%	99.896%	91.882%	87.258%
DT+LR+Majority Voting	70.341%	99.459%	99.886%	91.870%	87.238%
DT+MLP+Majority Voting	70.277%	99.549%	99.905%	91.855%	87.214%
DT+SGD+Majority Voting	55.527%	99.971%	99.995%	88.296%	81.469%
DT+SVM+Majority Voting	65.419%	99.976%	99.995%	90.656%	85.287%
DT+GB+Majority Voting	70.421%	99.550%	99.905%	91.891%	87.272%
DT+RF+Majority Voting	70.452%	99.460%	99.886%	91.898%	87.283%
GNB+KNN+Majority Voting	61.341%	97.915%	99.611%	89.632%	83.649%
GNB+LDA+Majority Voting	38.292%	87.323%	98.343%	84.245%	74.899%
GNB+LR+Majority Voting	46.161%	91.854%	98.780%	98.780%	77.811%
GNB+MLP+Majority Voting	68.334%	96.971%	99.364%	91.326%	86.382%
GNB+SGD+Majority Voting	40.968%	96.294%	99.530%	84.978%	76.051%
GNB+SVM+Majority Voting	41.430%	99.541%	99.943%	85.131%	76.283%
GNB+GB+Majority Voting	70.548%	96.724%	99.288%	91.877%	87.266%
GNB+RF+Majority Voting	68.637%	95.734%	99.088%	91.380%	86.476%

Table 4.6: two-year-prediction using Majority Voting (part 1)

Two-year-prediction using Majority Voting and other methods					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
KNN+LDA+Majority Voting	56.865%	99.139%	99.853%	88.594%	81.958%
KNN+LR+Majority Voting	58.538%	99.164%	99.853%	88.987%	82.597%
KNN+MLP+Majority Voting	66.900%	99.291%	99.858%	91.009%	85.859%
KNN+SGD+Majority Voting	53.807%	99.441%	99.910%	87.889%	80.809%
KNN+SVM+Majority Voting	58.474%	99.783%	99.962%	88.983%	82.586%
KNN+GB+Majority Voting	67.474%	99.835%	99.967%	91.160%	86.098%
KNN+RF+Majority Voting	66.900%	99.810%	99.962%	91.018%	85.870%
LDA+LR+Majority Voting	34.724%	96.162%	99.587%	83.657%	73.859%
LDA+MLP+Majority Voting	62.807%	97.817%	99.582%	89.983%	84.217%
LDA+SGD+Majority Voting	23.766%	94.610%	99.596%	81.424%	70.128%
LDA+SVM+Majority Voting	23.750%	99.400%	99.957%	81.476%	70.193%
LDA+GB+Majority Voting	62.488%	97.953%	99.611%	89.909%	84.096%
LDA+RF+Majority Voting	59.430%	98.055%	99.649%	89.179%	82.915%
LR+MLP+Majority Voting	64.511%	97.897%	97.897%	90.399%	84.886%
LR+SGD+Majority Voting	44.043%	96.984%	99.592%	85.656%	77.165%
LR+SVM+Majority Voting	43.772%	99.637%	99.953%	85.641%	77.124%
LR+GB+Majority Voting	62.504%	97.929%	99.606%	89.912%	84.102%
LR+RF+Majority Voting	61.883%	98.007%	99.625%	89.764%	83.862%

Table 4.7: two-year-prediction using Majority Voting (part 2)

Two-year-prediction using Majority Voting and other methods					
Method	Bankruptcy Recall	Bankruptcy Precision	Non-Bankruptcy Recall	Non-Bankruptcy Precision	Model overall accuracy
MLP+SGD+Majority Voting	51.736%	98.246%	99.725%	87.394%	80.009%
MLP+SVM+Majority Voting	64.798%	99.730%	99.948%	90.500%	85.038%
MLP+GB+Majority Voting	68.222%	99.281%	99.853%	91.337%	86.385%
MLP+RF+Majority Voting	68.461%	99.124%	99.820%	91.394%	86.477%
SGD+SVM+Majority Voting	53.807%	99.705%	99.953%	87.893%	80.815%
SGD+GB+Majority Voting	56.515%	100.000%	100.000%	88.527%	81.844%
SGD+RF+Majority Voting	54.603%	100.000%	100.000%	88.082%	81.121%
SVM+GB+Majority Voting	66.773%	100.000%	100.000%	90.989%	85.823%
SVM+RF+Majority Voting	66.199%	100.000%	100.000%	90.848%	85.596%
GB+RF+Majority Voting	72.523%	99.978%	99.995%	92.430%	88.130%

Table 4.8: two-year-prediction using Majority Voting (part 3)

Chapter 5

Analysis

Viewing the fact that our experiments can be divided into separate parts, it's suitable to make analysis of these two parts individually.

5.1 Visualization

From the heat map of Figure 4.1, the feature of *Bransje* had a relatively noticeable positive correlation with the feature of *Organization_Form*. In a similar case, it can also be found that the feature of *Stiftet* also had a positive correlation with the feature of *Organization_Form*. Besides, the feature of *Registration_Month* had a comparatively lower positive correlation with the feature of *Registration_Month*. From Figure 4.2, *Organization_Form* had a weak positive correlation with *Ansatte*. Meanwhile, the feature of *Stiftet* had a correlation with the feature of *Organization_Form*. In Figure 4.3, the feature of *Organization_Form* had a strong positive correlation with *Bransje*. The Figure 4.4 revealed that the strong positive correlation between *Registration_Month* and *Class*, weak positive correlation between *Organization_Form* and *Stiftet*, strong negative correlation between *Class* and *Stiftet*, negative correlation between *Class* and *Stiftet*, weak negative correlation between *Registration_Month* and *Stiftet*, weak negative correlation between *Organization_Form* and *Stiftet*. Figure 4.5 showed the weak correlation between *Ansatte* and *Organization_Form*. Figure 4.6 suggested that there existed weak positive correlation between *Bransje* and *Ansatte*, weak positive relationship between *Stiftet* and *Registration_Month*, strong negative correlation between *Stiftet* and *Organization_Form* .

From the Figure 4.13 in section of Bubble Chart, we can notice the fact that among all the counties, Oslo ranked first both in the number of bankruptcy and the number of all companies. The county of Akerhus ranked second in the number of companies but ranked third in the number of bankruptcy companies. Next to it, the Hordaland ranked third in the number of companies while it ranked second in the number of bankruptcy among

all these counties. Then we noticed that Rogaland ranked fourth and Trøndelag ranked fifth in the number of bankruptcy and the number of companies. In fact, the number of companies and bankruptcy companies were both much higher than the others. When we zoomed in and checked the Figure 4.14, we can see the relevant data of the remaining counties. Usually, the higher number of the companies a county had, the higher number of bankruptcy companies a county had. Besides, since the diameter of each county denoted its own percentage of bankruptcy, we can have an intuitive and explicit understanding of the bankruptcy situation in each county from this aspect. The situation of data of 2018 was quite similar to the data of 2016. From the Figure 4.15, The only noticeable difference was that Arkerhus beat Hordaland in both the number of bankruptcy and the number of companies.

Besides, we should also notice that different color denotes that this county belonged to different areas. As for all the counties of Norway, they can be divided into five different parts, including Eastern Norway (Østlandet), Southern Norway (Sørlandet), Western Norway (Vestlandet), Northern Norway (Nord-Norge) and Trøndelag (Midt-Norge). From the Figure 4.17, we can have a very intuitive idea about the percentage of bankruptcy indifferent areas. Western areas had the highest percentage of bankruptcy of 21.463%. Eastern Norway was the second in the percentage of bankruptcy. Northern Norway ranked the third, Southern Norway ranked fourth and Trøndelag ranked fifth in the bankruptcy percentage. This rule was also the same with the data of 2018. In fact, in Figure 4.18, Western Norway also has the highest percentage of bankruptcy. While at the same time, Trøndelag had the lowest percentage of bankruptcy. In view of that, we can have a holistic idea about the bankruptcy situation in different areas of Norway.

When we examine the Figure 4.23 and Figure 4.24, we can find that the county of Østfold had the highest percentage of bankruptcy. While at the same time, the county of Hordaland and the county of Buskerud also had relatively high percentage of bankruptcy in 2016, because their colors were rather red and the radius of their corresponding circles were larger than others. Meanwhile, it was noticeable that the county of Sogn og Fjordane had the lowest percentage of bankruptcy in the same year. Similarly, when the Figure 4.25 and Figure 4.26 were studied, we can quickly notice that the county of Østfold, Hordaland and Hedmark had the highest percentage of bankruptcy in 2018. Like the case of 2016, the county of Sogn og Fjordane also had the lowest percentage of bankruptcy. Therefore, it was safe to say that those companies in Sogn og Fjordane are less likely to become bankrupt, while companies in Østfold and Hordaland have a high rate of bankruptcy.

5.2 Single classifiers & Ensemble Learning

In this part, we firstly came to these nine single classifiers, including Logistic Regression, Linear Discriminant Analysis, Support Vector Machine, Stochastic Gradient Descent, Decision Tree, k-Nearest Neighbors, k-Means, Naive Bayes, Multi Layer Perceptron. The results of using only one individual classifier can be shown in the form of Confusion Matrices. Let's take the Figure 4.27 as an instance. The first confusion matrix stood for the result of bankruptcy when we divided the data of 2016 into training set and testing set, constructed the prediction model on the part of the training set and applied them on the part of testing set. The second confusion matrix of Figure 4.27 denoted the result of building the data mining model on the training set of the 2018 data and applying this model for bankruptcy prediction for the testing part of 2018 data. The third confusion matrix presented the result when we constructed a data mining model based on the data set of 2016 to apply this model upon the data set of 2018 for the problem of bankruptcy prediction.

As was known to us, confusion matrix was often used in classification problem. It was a common but helpful tool in assessing and describing the performance of a model for classification. Usually, for a binary problem, there existed four parts for the result of a classification problem: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Among all these four parts, TP meant that the model had classified those elements as the positive ones when those classified elements were positive in reality. TN meant that some items had been correctly classified into the negative classification, which they were in reality. FP suggested that some items had been wrongly put into the classification of positive, while they actually belonged to the classification of negative. FN showed that some items has been classified as the negative ones. However, those wrongly classified items should been classified as the positive ones. Evidently, the existence of confusion matrix can aid us in quickly getting the picture of the results of binary classification.

In fact, in the following eight figures from Figure 4.28 to Figure 4.35, we presented the confusion matrices of the results of the three cases of constructing prediction model and applying it on the data of 2016, building model and making prediction based on the data of 2018, constructing model on the data of 2016 and applying it upon the data of 2018.

Besides these confusion matrices, we also presented their own corresponding indexes including Bankruptcy Recall, Bankruptcy Precision, Non-Bankruptcy Recall, Non-Bankruptcy Precision, Model Overall Accuracy respectively. Bankruptcy Recall equaled the division between TP and the sum of TP and NP, which in fact denoted the ability of this model to find out all the real bankruptcy items. Bankruptcy Precision can be figured out by getting the division of TP and (TP+FP), which suggested the ability of this model to ensure that relevance between what had been classified as bankruptcy and what were bankruptcy ones in reality. Similarly, we can have a quick understanding of the terms of Non-Bankruptcy Recall and Non-Bankruptcy Precision. Non-Bankruptcy Recall showed the ability to find out all the relevant non-bankruptcy items. And Non-Bankruptcy Precision was a metric measure to show the model's ability in only putting the non-bankruptcy ones. Apart from those four terms, we also had the Model Overall Accuracy, which was defined as the (TP+TN) / (TP+FP+TN+FN). This term of Model Overall Accuracy can bring an intuitive

and holistic impression about the overall performance of a specific model, because it had taken both the positive and negative items into considerations at the same time.

Even though we had done three parts including *Table 4.1: 2016-single-classifier*, *Table 4.2: 2018-single-classifier*, *Table 4.3: Two-Year-Prediction-single-classifier* on the section of single classifier, those three parts had different meanings. Table 4.1 describes the results, when we built a model on the data of 2016 and applied it upon the testing data of 2016. Table 4.2 dealt with the data of 2018. Model was trained on the training data of 2018 and applied on the testing data. And Table 4.3 built prediction model based on the data of 2016 and applied it on the data of 2018. Here we needed to point out we refer this kind of prediction as two-year-prediction throughout the whole project. Apparently, we can notice that those three approaches including Decision Tree, Support Vector Machine, Multilayer Perceptron ranked as best three methods in bankruptcy prediction. In fact, their Bankruptcy Recall and Model Overall Accuracy were far higher than the performances of other algorithms. Moreover, in the remain parts, we only focused on the two-year-prediction. This was mainly because of the limitations in the time that we can spend on the thesis.

After the part of single classifier, we came to the part of multiple classifiers. Here we utilized two approaches including Gradient Boosting and Random Forest firstly. Both of these two methods belonged to Ensemble Learning. After that, we conducted several experiments by combining the Bagging Ensemble Learning with other single classifier methods that we had mentioned above. Then we considered introducing the ensemble learning of Majority Voting and utilize several methods each time to form better models. In Table 4.5, it suggested that when we combined SGD and Bagging Ensemble Learning, we can have a prediction model with good performance. Also the model formed using Random Forest and Bagging was as good as the former model in performance. Last but not least, the model constructed using Gradient Boosting and Bagging also had gotten some good experiment results. As for the part using Majority Voting, we combined several methods to build new models in bankruptcy prediction. Since there were amounts of different combinations, results of relevant experiments were presented in Table 4.6, Table 4.7 and Table 4.8.

Chapter 6

Conclusion

In our thesis, we utilized various data mining algorithms including machine learning algorithms and deep learning algorithms and different visualization approaches in dealing with the problem of bankruptcy prediction. Data set included the information about bankruptcy companies of Norway in 2016, non-bankruptcy companies of Norway in 2016, bankruptcy companies of Norway in 2018, non-bankruptcy companies of Norway in 2018. In this thesis, visualization approaches including Heat Map, Box Plot, Bubble Chart, Bar Chart, Visualization based on GIS and map, Confusion Matrix were implemented. Besides, in data mining parts, we adopted methods including Decision Tree, Gaussian Naive Bayes, K Means Clustering, K Nearest Neighbor, Linear Discriminant Analysis, Logistic Regression, Multi Layer Perceptron, Stochastic Gradient Descent, Support Vector Machine, Random Forest Ensemble Learning, Gradient Boosting Ensemble Learning, Bagging Ensemble Learning, Majority Voting Ensemble Learning.

The results in the visualization part gave us a holistic and intuitive understanding about the bankruptcy situations of 2016 and 2018. For instance, from Figure 4.13 to Figure 4.16, we can know the number of bankruptcy companies, the number of all the companies, the percentage of bankruptcy in 2016 and 2018 respectively. Similarly, the Bar Chart of Figure 4.17 and Figure 4.18, showed the bankruptcy situation of companies in different regions of Norway.

From what we have discussed in our thesis, we can now come to solving those research questions that we have proposed in the chapter 1.

Research Question 1: What are the methods that other researchers utilize in the field of bankruptcy prediction?

Traditionally, some relevant researchers rely on some financial terms and statistical models to solve the issue of bankruptcy prediction. For example, Z-Score, Discriminant Analysis, Case-Based Reasoning (CBR), Logistic Regression Analysis, Principal Component Analysis, Support Vector Machines are often used as common methods in bankruptcy

prediction.

Research Question 2: How can we use data mining algorithms and visualization methods in the field of bankruptcy prediction?

In our paper, visualization methods showed us the features of data from various aspects. For instance, we can use Heat Map to intuitively present the correlations among all features. And Visualization based on GIS and maps can help us have an understanding of the bankruptcy percentage in different counties, which was important in giving guidance on the bankruptcy prediction of different counties. As for the part of the data mining algorithms, we firstly built our prediction models based on the data of 2016, then we applied those models upon the data of 2018 to test the precision and accuracy of our bankruptcy prediction models.

Research Question 3: What are the conclusions and suggestions that we can get from our research?

Firstly, in our thesis, the best prediction model successfully achieved a Model Overall Accuracy of 88.130% by combining the methods of Gradient Boosting, Random Forest and Majority Voting. With the help of this model, we can give a classification of any company in Norway. For example, when another new Norwegian company pops up, we can get and input its relevant data into our model. Saying this company is classified into the class of *bankruptcy companies*, then we can give prompt advice and warning to this company. Secondly, visualization can give us intuitive understandings and ideas about the bankruptcy situation of Norway. For instance, From Figure 4.19 to Figure 4.26, the visualization based on GIS and map can show the number of bankruptcy and the percentage of bankruptcy in different counties.

6.1 Future Work

As we have mentioned in the Chapter 1, there are some limitations and scopes for our thesis. Therefore, More explorations can be done beyond these limitations if we or other relevant researchers have more time.

In fact, we only have the data of bankruptcy companies and non-bankruptcy companies in 2016 and 2018. And this is also the reason why most of our experiments are *two-year-prediction*, which build the prediction models based on the data set of 2016 and test them on the data set of 2018. Thus, as long as future researchers have more detailed data, it's likely that they can form more accurate and robust models. For instance, if the data about the bankruptcy companies in 2017 can be fully known, we can improve the process of forming model by treating data of 2016 as the training set, the data of 2017 as the validation set, the data of 2018 as the testing data set. Moreover, now we can only build a model to predict the possibility of bankruptcy of a specific company. However, even if we have classified an undetermined company as a potential bankruptcy company, it's not possible to give an approximate estimation about when it will go bankruptcy in the future. This point can also be improved as long as we have much larger data sets that

contain the bankruptcy records of all companies in Norway for a series of years. In that case, we can consider introducing some relevant measures in time prediction like RNN and LSTM (Long Short-Term Memory Recurrent Neural Network), which has a good ability in memory and time prediction. [117]

Bibliography

[1] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[2] William H Beaver. Financial ratios as predictors of failure. *Journal of accounting research*, pages 71–111, 1966.

[3] Edward I Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance*, 23(4):589–609, 1968.

[4] Jarrod W Wilcox. A simple theory of financial ratios as predictors of failure. 1970.

[5] James A Ohlson. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research*, pages 109–131, 1980.

[6] Charles F Manski, Daniel McFadden, et al. *Structural analysis of discrete data with econometric applications*. MIT press Cambridge, MA, 1981.

[7] Lisa R Gilbert, Krishnagopal Menon, and Kenneth B Schwartz. Predicting bankruptcy for firms in financial distress. *Journal of Business Finance & Accounting*, 17(1):161–171, 1990.

[8] Tyler Shumway. Forecasting bankruptcy more accurately: A simple hazard model. *The journal of business*, 74(1):101–124, 2001.

[9] Sudheer Chava and Robert A Jarrow. Bankruptcy prediction with industry effects. *Review of Finance*, 8(4):537–569, 2004.

[10] Maciej Zikeba, Sebastian K Tomczak, and Jakub M Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, 2016.

[11] Daniel A Keim, Ming C Hao, Umesh Dayal, and Meichun Hsu. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization*, 1(1):20–34, 2002.

[12] Diansheng Guo. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization*, 2(4):232–246, 2003.

[13] Melanie Tory and Torsten Moller. Rethinking visualization: A high-level taxonomy. In *IEEE Symposium on Information Visualization*, pages 151–158. IEEE, 2004.

[14] James B Pick. Geographic information systems: A tutorial and introduction. *The Communications of the Association for Information Systems*, 14(1):50, 2004.

[15] Leland Wilkinson, Anushka Anand, and Robert Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.

[16] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual analytics: Scope and challenges. In *Visual data mining*, pages 76–90. Springer, 2008.

[17] Andrada Tatu, Georgia Albuquerque, Martin Eisemann, Jorn Schneidewind, Holger Theisel, Marcus Magnork, and Daniel Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66. IEEE, 2009.

[18] Enrico Bertini and Denis Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *Acm Sigkdd Explorations Newsletter*, 11(2):9–18, 2010.

[19] Lisa Meloncon and Emily Warner. Data visualizations: A literature review and opportunities for technical and professional communication. In *2017 IEEE International Professional Communication Conference (ProComm)*, pages 1–9. IEEE, 2017.

[20] Choong Nyoung Kim and Raymond McLeod Jr. Expert, linear models, and nonlinear models of expert decision making in bankruptcy prediction: a lens model analysis. *Journal of Management Information Systems*, 16(1):189–206, 1999.

[21] Tae Kyung Sung, Namsik Chang, and Gunhee Lee. Dynamics of modeling in data mining: interpretive approach to bankruptcy prediction. *Journal of Management Information Systems*, 16(1):63–85, 1999.

[22] Hui Li and Jie Sun. On performance of case-based reasoning in chinese business failure prediction from sensitivity, specificity, positive and negative values. *Applied Soft Computing*, 11(1):460–467, 2011.

[23] Gang Wang and Jian Ma. Study of corporate credit risk prediction based on integrating boosting and random subspace. *Expert Systems with Applications*, 38(11):13871–13878, 2011.

[24] Fengyi Lin, Ching-Chiang Yeh, and Meng-Yuan Lee. The use of hybrid manifold learning and support vector machines in the prediction of business failure. *Knowledge-Based Systems*, 24(1):95–101, 2011.

[25] Hui Li, Hojjat Adeli, Jie Sun, and Jian-Guang Han. Hybridizing principles of topsis with case-based reasoning for business failure prediction. *Computers & Operations Research*, 38(2):409–419, 2011.

[26] Soo Y Kim. Prediction of hotel bankruptcy using support vector machine, artificial neural network, logistic regression, and multivariate discriminant analysis. *The Service Industries Journal*, 31(3):441–468, 2011.

[27] Arindam Chaudhuri and Kajal De. Fuzzy support vector machine for bankruptcy prediction. *Applied Soft Computing*, 11(2):2472–2486, 2011.

[28] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

[29] Pediredla Ravisankar, Vadlamani Ravi, G Raghava Rao, and Indranil Bose. Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, 50(2):491–500, 2011.

[30] Wolfgang K Härdle, Dedy Prastyo, and Christian Hafner. Support vector machines with evolutionary feature selection for default prediction. 2012.

[31] Gang Wang and Jian Ma. A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. *Expert Systems with Applications*, 39(5):5325–5331, 2012.

[32] Mehdi Divsalar, Habib Roodsaz, Farshad Vahdatinia, Ghassem Norouzzadeh, and Amir Hossein Behrooz. A robust data-mining approach to bankruptcy prediction. *Journal of Forecasting*, 31(6):504–523, 2012.

[33] Myoung-Jong Kim and Dae-Ki Kang. Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction. *Expert Systems with applications*, 39(10):9308–9314, 2012.

[34] Tsung-Jung Hsieh, Hsiao-Fen Hsiao, and Wei-Chang Yeh. Mining financial distress trend data using penalty guided support vector machines based on hybrid of particle swarm optimization and artificial bee colony algorithm. *Neurocomputing*, 82:196–206, 2012.

[35] Andrey Volkov and Dirk Van den Poel. Extracting information from sequences of financial ratios with markov for discrimination: an application to bankruptcy prediction. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 340–343. IEEE, 2012.

[36] David L Olson, Dursun Delen, and Yanyan Meng. Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, 52(2):464–473, 2012.

[37] Jie Sun and Hui Li. Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing*, 12(8):2254–2265, 2012.

[38] Shian-Chang Huang, Yu-Cheng Tang, Chih-Wei Lee, and Ming-Jen Chang. Kernel local fisher discriminant analysis based manifold-regularized svm model for financial distress predictions. *Expert Systems with Applications*, 39(3):3855–3861, 2012.

[39] Wei-Yang Lin, Ya-Han Hu, and Chih-Fong Tsai. Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):421–436, 2012.

[40] Jae Kwon Bae. Predicting financial distress of the south korean manufacturing industries. *Expert Systems with Applications*, 39(10):9159–9165, 2012.

[41] Chih-Fong Tsai and Kai-Chun Cheng. Simple instance selection for bankruptcy prediction. *Knowledge-Based Systems*, 27:333–342, 2012.

[42] Isye Arieshanti, Yudhi Purwananto, Ariestia Ramadhani, Mohamat Ulin Nuha, and Nurissaidah Ulinnuha. Comparative study of bankruptcy prediction models. *Telkomnika*, 11(3):591, 2013.

[43] Mu-Yen Chen. A hybrid anfis model for business failure prediction utilizing particle swarm optimization and subtractive clustering. *Information Sciences*, 220:180–195, 2013.

[44] Fengyi Lin, Ching Chiang Yeh, Meng Yuan Lee, et al. A hybrid business failure prediction model using locally linear embedding and support vector machines. *Romanian Journal of Economic Forecasting*, 16(1):82–97, 2013.

[45] Chih-Fong Tsai and Yu-Feng Hsu. A meta-learning framework for bankruptcy prediction. *Journal of Forecasting*, 32(2):167–179, 2013.

[46] Narendar V Rao, Gokhul Atmanathan, Manu Shankar, and Srivatsan Ramesh. Analysis of bankruptcy prediction models and their effectiveness: An indian perspective. *Gt. Lakes Her*, 7(2), 2013.

[47] Elena Fedorova, Evgenii Gilenko, and Sergey Dovzhenko. Bankruptcy prediction for russian companies: Application of combined classifiers. *Expert Systems with Applications*, 40(18):7285–7293, 2013.

[48] Ning Chen, Bernardete Ribeiro, Armando Vieira, and An Chen. Clustering and visualization of bankruptcy trajectory using self-organizing map. *Expert Systems with Applications*, 40(1):385–393, 2013.

[49] Carlos Serrano-Cinca and BegoñA GutiéRrez-Nieto. Partial least square discriminant analysis for bankruptcy prediction. *Decision support systems*, 54(3):1245–1255, 2013.

[50] Ligang Zhou. Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems*, 41:16–25, 2013.

[51] Tengke Xiong, Shengrui Wang, André Mayers, and Ernest Monga. Personal bankruptcy prediction by mining credit card data. *Expert systems with applications*, 40(2):665–676, 2013.

[52] Birsen Eygi Erdogan. Prediction of bankruptcy using support vector machines: an application to bank bankruptcy. *Journal of Statistical Computation and Simulation*, 83(8):1543–1555, 2013.

[53] Mohiuddin Ahmed and Abdun Naser Mahmood. A novel approach for outlier detection and clustering improvement. In *Industrial electronics and applications (ICIEA), 2013 8th IEEE conference on*, pages 577–582. IEEE, 2013.

[54] Gang Wang, Jian Ma, and Shanlin Yang. An improved boosting based on feature selection for corporate bankruptcy prediction. *Expert Systems with Applications*, 41(5):2353–2361, 2014.

[55] Chih-Fong Tsai. Combining cluster analysis with classifier ensembles to predict financial distress. *Information Fusion*, 16:46–58, 2014.

[56] Niccolo Gordini. A genetic algorithm approach for smes bankruptcy prediction: Empirical evidence from italy. *Expert systems with applications*, 41(14):6433–6445, 2014.

[57] Yan Huang and Gang Kou. A kernel entropy manifold learning approach for financial data analysis. *Decision Support Systems*, 64:31–42, 2014.

[58] Junyoung Heo and Jin Yong Yang. Adaboost based bankruptcy forecasting of korean construction companies. *Applied soft computing*, 24:494–499, 2014.

[59] Qi Yu, Yoan Miche, Eric Séverin, and Amaury Lendasse. Bankruptcy prediction using extreme learning machine and financial expertise. *Neurocomputing*, 128:296–302, 2014.

[60] Ming-Chang Lee. Business bankruptcy prediction based on survival analysis approach. *International Journal of Computer Science & Information Technology*, 6(2):103, 2014.

[61] Qi Yu, Mark Van Heeswijk, Yoan Miche, Rui Nian, Bo He, Eric Séverin, and Amaury Lendasse. Ensemble delta test-extreme learning machine (dt-elm) for regression. *Neurocomputing*, 129:153–158, 2014.

[62] Ching-Chiang Yeh, Der-Jang Chi, and Yi-Rong Lin. Going-concern prediction using hybrid random forests and rough set approach. *Information Sciences*, 254:98–110, 2014.

[63] Joaquín Abellán and Carlos J Mantas. Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, 41(8):3825–3830, 2014.

[64] Fengyi Lin, Deron Liang, Ching-Chiang Yeh, and Jui-Chieh Huang. Novel feature selection methods to financial distress prediction. *Expert Systems with Applications*, 41(5):2472–2483, 2014.

[65] Gintautas Garšva and Paulius Danenas. Particle swarm optimization for linear support vector machines based classifier selection. *Nonlinear Analysis: Modelling and Control*, 19(1):26–42, 2014.

[66] Suman Renu. Analysis on credit card fraud detection methods. *International Journal of Computer Trends and Technology (IJCTT)–volume*, 8, 2014.

[67] Emanuel Mineda Carneiro, Luiz Alberto Vieira Dias, Adilson Marques da Cunha, and Lineu Fernando Stege Mialaret. Cluster analysis and artificial neural networks: A case study in credit card fraud detection. In *2015 12th International Conference on Information Technology-New Generations (ITNG)*, pages 122–126. IEEE, 2015.

[68] Nader Mahmoudi and Ekrem Duman. Detecting credit card fraud by modified fisher discriminant analysis. *Expert Systems with Applications*, 42(5):2510–2516, 2015.

[69] Myoung-Jong Kim, Dae-Ki Kang, and Hong Bae Kim. Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Systems with Applications*, 42(3):1074–1082, 2015.

[70] Gergely Fejér-Király. Bankruptcy prediction: A survey on evolution, critiques, and solutions. *Acta Universitatis Sapientiae, Economics and Business*, 3(1):93–108, 2015.

[71] Alaka Hafiz, Oyedele Lukumon, Bilal Muhammad, Akinade Olugbenga, Owolabi Hakeem, and Ajayi Saheed. Bankruptcy prediction of construction businesses: towards a big data analytics approach. In *2015 IEEE First International Conference on Big Data Computing Service and Applications*, pages 347–352. IEEE, 2015.

[72] Kalyan Nagaraj and Amulyashree Sridhar. A predictive system for detection of bankruptcy using machine learning techniques. *arXiv preprint arXiv:1502.03601*, 2015.

[73] Félix J López Iturriaga and Iván Pastor Sanz. Bankruptcy visualization and prediction using neural networks: A study of us commercial banks. *Expert Systems with applications*, 42(6):2857–2869, 2015.

[74] Philippe du Jardin. A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research*, 254(1):236–252, 2016.

[75] Ali Mansouri, Arezoo Nazari, and Morteza Ramazani. A comparison of artificial neural network model and logistics regression in prediction of companies bankruptcy (a case study of tehran stock exchange). *International Journal of Advanced Computer Research*, 6(24), 2016.

[76] Fatima Zahra Azayite and Said Achchab. Hybrid discriminant neural networks for bankruptcy prediction and risk scoring. *Procedia Computer Science*, 83:670–674, 2016.

[77] Hyun-Jung Kim, Nam-Ok Jo, and Kyung-Shik Shin. Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Systems with Applications*, 59:226–234, 2016.

[78] Dong Zhao, Chunyu Huang, Yan Wei, Fanhua Yu, Mingjing Wang, and Huiling Chen. An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. *Computational Economics*, 49(2):325–341, 2017.

[79] Flavio Barboza, Herbert Kimura, and Edward Altman. Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, 83:405–417, 2017.

[80] Nanxi Wang. Bankruptcy prediction using machine learning. *Journal of Mathematical Finance*, 7(04):908, 2017.

[81] David Martens, Tony Van Gestel, Manu De Backer, Raf Haesen, Jan Vanthienen, and Bart Baesens. Credit rating prediction using ant colony optimization. *Journal of the Operational Research Society*, 61(4):561–573, 2010.

[82] Mingjing Wang, Huiling Chen, Huaizhong Li, Zhennao Cai, Xuehua Zhao, Changfei Tong, Jun Li, and Xin Xu. Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction. *Engineering Applications of Artificial Intelligence*, 63:54–68, 2017.

[83] Chih-Hsun Chou, Su-Chen Hsieh, and Chui-Jie Qiu. Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction. *Applied Soft Computing*, 56:298–316, 2017.

[84] Jie Sun, Hamido Fujita, Peng Chen, and Hui Li. Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble. *Knowledge-Based Systems*, 120:4–14, 2017.

[85] Frank Wagenmans. Machine learning in bankruptcy prediction. Master's thesis, 2017.

[86] David O Wilson, Joel L Weissfeld, Arzu Balkan, Jeffrey G Schragin, Carl R Fuhrman, Stephen N Fisher, Jonathan Wilson, Joseph K Leader, Jill M Siegfried, Steven D Shapiro, et al. Association of radiographic emphysema and airflow obstruction with lung cancer. *American journal of respiratory and critical care medicine*, 178(7):738–744, 2008.

[87] Zhongsheng Hua, Yu Wang, Xiaoyan Xu, Bin Zhang, and Liang Liang. Predicting corporate financial distress based on integration of support vector machine and logistic regression. *Expert Systems with Applications*, 33(2):434–440, 2007.

[88] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pages 41–48. Ieee, 1999.

[89] Alan Julian Izenman. Linear discriminant analysis. In *Modern multivariate statistical techniques*, pages 237–280. Springer, 2013.

[90] Lipo Wang. *Support vector machines: theory and applications*, volume 177. Springer Science & Business Media, 2005.

[91] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[92] Mark A Friedl and Carla E Brodley. Decision tree classification of land cover from remotely sensed data. *Remote sensing of environment*, 61(3):399–409, 1997.

[93] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[94] Wei-Yin Loh. Classification and regression trees. *Institute for Mathematical Sciences*, 10:19, 2014.

[95] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[96] Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & security*, 21(5):439–448, 2002.

[97] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[98] Min-Ling Zhang, José M Peña, and Victor Robles. Feature selection for multi-label naive bayes classification. *Information Sciences*, 179(19):3218–3229, 2009.

[99] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[100] Wangchao Lou, Xiaoqing Wang, Fan Chen, Yixiao Chen, Bo Jiang, and Hua Zhang. Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes. *PLoS One*, 9(1):e86703, 2014.

[101] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[102] Robert Stengel. Introduction to neural networks! 2017.

[103] Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on neural networks*, 3(5):683–697, 1992.

[104] Rudolf Kruse, Christian Borgelt, Christian Braune, Sanaz Mostaghim, and Matthias Steinbrecher. Multilayer perceptrons. In *Computational Intelligence*, pages 47–92. Springer, 2016.

[105] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[107] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002.

[108] Cen Li. Classifying imbalanced data using a bagging ensemble variation (bev). In *Proceedings of the 45th annual southeast regional conference*, pages 203–208. ACM, 2007.

[109] Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[110] Xuchun Li, Lei Wang, and Eric Sung. Adaboost with svm-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5):785–795, 2008.

[111] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[112] Ronald L Breiger, Scott A Boorman, and Phipps Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of mathematical psychology*, 12(3):328–383, 1975.

[113] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[114] David M Theobald. *GIS concepts and ArcGIS methods*, volume 2. Conservation Planning Technologies Fort Collins, CO, 2003.

[115] Ying Zhu. Introducing google chart tools and google maps api in data visualization courses. *IEEE computer graphics and applications*, 32(6):6–9, 2012.

[116] Andreas Mueller Peter Prettenhofer. Classification and regression using stochastic gradient descent (sgd). *Github/scikit-learn*, apr 2019.

[117] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

Appendix A: Source Code

A.1 Data Preprocessing

```
' There are some Visual Basic Scripts
' We use the data of 2016 as an example
' Here we only show the part of the bankruptcy data of 2016,
' Even though there are some variations and differences
' When compared with the part of non-bankruptcy of 2016

' Script for the feature of Registration_Month from
' the file of Konkurs statistikk 2016.xlsx
Sub registrationMonthCode()

    For rowNumber = 2 To 28720


    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("D" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("A" & rowNumber).Value = Int((Int(num Mod 10000)) / 100)
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("A" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Bransje from
' the file of Konkurs statistikk 2016.xlsx
Sub bransjeCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("G" & rowNumber).Value
    If (num <> 0) Then
```

```
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("B" & rowNumber).Value = num
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("B" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Fylke from
' the file of Konkurs statistikk 2016.xlsx
Sub fylkeCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("H" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("C" & rowNumber).Value = num
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("C" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Kommune from
' the file of Konkurs statistikk 2016.xlsx
Sub kommuneCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("I" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("D" & rowNumber).Value = num
    Else
```

```
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("D" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Stiftet from
' the file of Konkurs statistikk 2016.xlsx
Sub stiftetCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("Q" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016")
        .Range("E" & rowNumber).Value = Int(num / 10000)
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("E" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Share_Capital from
' the file of Konkurs statistikk 2016.xlsx
Sub shareCapitalCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("R" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("F" & rowNumber).Value = num
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("F" & rowNumber).Value = 0
    End If
```

```
    Next

End Sub


' Script for the feature of Organization_Form from
' the file of Konkurs statistikk 2016.xlsx
Sub organizationFormCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("S" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("G" & rowNumber).Value = num
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("G" & rowNumber).Value = 0
    End If

    Next

End Sub


' Script for the feature of Ansatte from
' the file of Konkurs statistikk 2016.xlsx
Sub ansatteCode()

    For rowNumber = 2 To 28720

    num = ThisWorkbook.Sheets("Bankruptcy 2016").
    Range("T" & rowNumber).Value
    If (num <> 0) Then
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("H" & rowNumber).Value = num
    Else
        ThisWorkbook.Sheets("Bankruptcy 2016").
        Range("H" & rowNumber).Value = 0
    End If

    Next
```

```
End Sub


# Now we come to the Python part.
# After we have gotten the bankruptcy data of 2016
# and non-bankruptcy data of 2016,
# We need to firstly replace those names in
# Orignization_form with categorical numbers,
# Then we should combine these two files together.

# coding: utf-8
# This code is run on Jupyter of Anaconda

import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

bank_pre = pd.read_csv("./2016 Bankruptcy.csv")

bank_pre.head()

bank_pre.shape

bank = bank_pre.drop(columns = ['TVANGS-AVVIKLING',
'BALANSESUM 2014','Alder'])
bank.head()

pre_non_bank = pd.read_csv("./2016 Non-Bankruptcy.csv")
pre_non_bank.tail(20)

pre_non_bank.shape
73


non_bank = pre_non_bank.dropna()
non_bank.tail(20)

non_bank.shape

#add class
bank['Class'] = 1
bank
```

```
#test
bank['Organization_Form'].unique()


non_bank['Class'] = 0
non_bank

#test
non_bank['Organization_Form'].unique()


#merge data
df_concated = pd.concat([bank,non_bank],sort=False)
df_concated

df_concated['Organization_Form'].unique()

type_length = len(df_concated['Organization_Form'].unique())
print(type_length)

res = df_concated['Organization_Form'].replace(
regex={'ESEK':'1','TVAM':'2','ANS':'3','ASA':'4','BBL':'5',
'BRL':'6','ENK':'7','FLI':'8','IKS':'9','KBO':'10',
'NUF':'11','PRE':'12','STI':'13','SAER':'14','AS':'15',
'BA':'16','DA':'17','KF':'18','KS':'19','SA':'20',
'SE':'21','PK':'22','SF':'23'})
res

df_concated["Organization_Form"]=res
df_concated

df_concated['Organization_Form'].unique()

df_concated.to_csv("Real 2016.csv")


#Similarly, we can get the Python code in dealing with
#the bankruptcy of 2018 and non-bankruptcy of 2018
# coding: utf-8

import numpy as np # linear algebra
import pandas as pd # data processing,
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns
```

```
bank_pre = pd.read_csv("./2018 Bankruptcy.csv")
bank_pre.head()

bank_pre.shape

bank = bank_pre.drop(columns = ['TVANGS-AVVIKLING',
'BALANSESUM 2016','Alder'])
bank.head()

pre_non_bank = pd.read_csv("./2018 Non-Bankruptcy.csv")
pre_non_bank.tail(20)

pre_non_bank.shape


non_bank = pre_non_bank.dropna()
non_bank.tail(20)

non_bank.shape

#add class
bank['Class'] = 1
bank

#test
bank['Organization_Form'].unique()

non_bank['Class'] = 0
non_bank

#test
non_bank['Organization_Form'].unique()

#merge data
df_concated = pd.concat([bank,non_bank])
df_concated

df_concated['Organization_Form'].unique()

type_length = len(df_concated['Organization_Form'].unique())
print(type_length)

res = df_concated['Organization_Form'].replace(
regex={'ESEK':'1','TVAM':'2','ANS':'3','ASA':'4','BBL':'5',
```

```
'BRL':'6','ENK':'7','FLI':'8','IKS':'9','KBO':'10',
'NUF':'11','PRE':'12','STI':'13','SAER':'14','AS':'15',
'BA':'16','DA':'17','KF':'18','KS':'19','SA':'20',
'SE':'21','PK':'22','SF':'23'})
res

df_concated["Organization_Form"]=res
df_concated

df_concated['Organization_Form '].unique()

df_concated.to_csv("Real 2018.csv")
```

A.2 Visualization Scripts

```
# Heat Map for data of 2016
# This is Python code
# This is run on Jupyter of Anaconda

# coding: utf-8

import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

data = pd.read_csv("../Real Data/Real 2016.csv")

data.shape

data.head()

data_bankruptcy = data[data["Class"]==1]
data_nonbankruptcy = data[data["Class"]==0]

data_bankruptcy.shape

data_nonbankruptcy.shape

data_bank = data_bankruptcy.drop(columns="Class")
data_bank.head()

data_nonbank = data_nonbankruptcy.drop(columns="Class")
data_nonbank.head()

#2016 bankruptcy

a4_dims = (12.5, 8.3)
fig, ax = plt.subplots(figsize=a4_dims)
sns.heatmap(data_bankruptcy.corr(),linewidths=.5,ax = ax)
plt.savefig("2016_heatMap_bankruptcy")

data_bankruptcy.corr()

a4_dims = (12.5, 8.3)
fig, ax = plt.subplots(figsize=a4_dims)
sns.heatmap(data_nonbankruptcy.corr(),linewidths=.5,ax = ax)
```

```
plt.savefig("2016_heatMap_nonbankruptcy")

data_nonbankruptcy.corr()

a4_dims = (12.5, 8.3)
fig, ax = plt.subplots(figsize=a4_dims)
sns.heatmap(data.corr(),linewidths=.5,ax = ax)
plt.savefig("2016_heatMap_all")

data.corr()


# Box Plot for data of 2016
# This is Python code
# This is run on Jupyter of Anaconda
# coding: utf-8

import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

data = pd.read_csv("../Real Data/Real 2016.csv")
data.shape

data.head()

data_bankruptcy = data[data["Class"]==1]
data_nonbankruptcy = data[data["Class"]==0]

data_bankruptcy.shape

data_nonbankruptcy.shape

data_bank = data_bankruptcy.drop(columns="Class")
data_bank.head()

data_nonbank = data_nonbankruptcy.drop(columns="Class")
data_nonbank.head()


# Stiftet Part

ansatte_bank = pd.Series(data_bankruptcy["Ansatte"],
```

```
name="bankruptcy")
ansatte_nonbank = pd.Series(data_nonbankruptcy["Ansatte"],
name="nonbankruptcy")

d = {"bankruptcy": ansatte_bank,
"nonbankruptcy": ansatte_nonbank}
ansatteForm = pd.DataFrame(data=d)

ansatteForm

d = data_bankruptcy["Ansatte"]

medians = d.median()
maximum = d.max()
minimum = d.min()

a4_dims = (10, 15)
sns.set(style="ticks", palette="pastel")
fig, ax = plt.subplots(figsize=a4_dims)

ax.text(0.1, medians+10, "medians="+str(int(medians)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, maximum+10, "maximum="+str(int(maximum)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, minimum-10, "minimum="+str(int(minimum)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')

plt.title("Box plot of Ansatte of bankruptcy in
2016",fontsize='20',weight='semibold')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
#ax.set_xlabel(..., fontsize=14)
ax.set_ylabel(..., fontsize=14)

sns.boxplot(y=d,ax = ax)
plt.savefig("2016_boxPlot_Ansatte_bankruptcy")


d = data_nonbankruptcy["Ansatte"]

medians = d.median()
maximum = d.max()
```

```
minimum = d.min()

a4_dims = (10, 15)
sns.set(style="ticks", palette="pastel")
fig, ax = plt.subplots(figsize=a4_dims)

ax.text(0.1, medians+5     , "medians="+str(int(medians)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, maximum       , "maximum="+str(int(maximum)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, minimum - 10 , "minimum="+str(int(minimum)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')

plt.title("Box plot of Ansatte of non-bankruptcy in 2016",
fontsize='20',weight='semibold')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
#ax.set_xlabel(..., fontsize=14)
ax.set_ylabel(..., fontsize=14)


sns.boxplot(y=d,ax = ax)
plt.savefig("2016_boxPlot_Ansatte_nonbankruptcy")

d = data["Ansatte"]

medians = d.median()
maximum = d.max()
minimum = d.min()

a4_dims = (10, 15)
sns.set(style="ticks", palette="pastel")
fig, ax = plt.subplots(figsize=a4_dims)

ax.text(0.1, medians +5      , "medians="+str(int(medians)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, maximum       , "maximum="+str(int(maximum)),
fontsize = '16',horizontalalignment='center',
color='black', weight='semibold')
ax.text(0.1, minimum -10       , "minimum="+str(int(minimum)),
fontsize = '16',horizontalalignment='center',
```

```
color='black', weight='semibold')

plt.title("Box plot of Ansatte of all data in 2016",
fontsize='20',weight='semibold')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
#ax.set_xlabel(..., fontsize=14)
ax.set_ylabel(..., fontsize=14)

sns.boxplot(y=d,ax = ax)

plt.savefig("2016_boxPlot_Ansatte_all")


//Here's the code of Bubble Chart of 2016
//It is written using HTML and JavaScript

<html>
   <head>
      <title>Google Charts Tutorial</title>
      <script type = "text/javascript"
      src = "https://www.gstatic.com/charts/loader.js">
      </script>
      <script type = "text/javascript">
         google.charts.load('current', {packages: ['corechart']});
      </script>
   </head>

   <body>
      <div id = "container" style = "width: 550px;
      height: 400px; margin: 0 auto">
      </div>
      <script language = "JavaScript">
```

function drawChart()

{ var data = google.visualization.arrayToDataTable([['ID', 'Total Count','Bankruptcy Count','Region', 'Bankruptcy Percentage'], ['Østfold', 1331,337, 'Eastern Norway (Østlandet)', 25.32,], ['Akershus', 3275, 660, 'Eastern Norway (Østlandet)',20.15,], ['Olso', 5846, 1186, 'Eastern Norway (Østlandet)', 20.29,], ['Hedmark', 827, 178, 'Eastern Norway (Østlandet)', 21.52,], ['Oppland', 752, 157, 'Eastern Norway (Østlandet)', 20.88,], ['Buskerud', 1421, 337, 'Eastern Norway (Østlandet)', 23.72,], ['Vestfold', 1284, 262, 'Eastern Norway (Østlandet)', 20.40,], ['Telemark', 885, 184, 'Eastern Norway (Østlandet)', 20.79,], ['Aust-Agder', 596, 113, 'Southern Norway (Sørlandet)', 18.96,], ['Vest-Agder', 1081, 229, 'Southern Norway (Sørlandet)', 21.18,], ['Rogaland', 2483, 475, 'Western Norway (Vestlandet)', 19.13,], ['Hordaland', 2950, 734, 'Western Nor-

way (Vestlandet)', 24.88,], ['Sogn og Fjordane', 441, 60, 'Western Norway (Vestlandet)', 13.61,], ['Møre og Romsdal', 1152, 239, 'Western Norway (Vestlandet)', 20.75,], ['Nordland', 1088, 226, 'Northern Norway (Nord-Norge)', 20.77,], ['Troms', 699, 145, 'Northern Norway (Nord-Norge)', 20.74,], ['Finnmark', 367, 78, 'Northern Norway (Nord-Norge)', 21.25,], ['Trøndelag', 2212, 408, 'Trøndelag (Midt-Norge)', 18.44,],]);

```
// Set chart options
var options =
{
title: 'Correlation between All Company Counts,
Bankruptcy Company Counts and Percentage of
Banruptcy in each county in 2016',
hAxis: {title: 'The Number of all the companies
in each county in 2016'},
vAxis: {title: 'The Number of bankruptcy
companies in each county in 2016'},
bubble: {textStyle: {fontSize: 15}},
'width':2000,
'height':1000,
sizeAxis: {minSize: 10, maxSize:30},
};

// Instantiate and draw the chart.
//var chart = new google.visualization.BubbleChart
(document.getElementById('container'));

var chart_div = document.getElementById('container');
var chart = new google.visualization.BubbleChart(chart_div);


// Wait for the chart to finish drawing before
// calling the getImageURI() method.
google.visualization.events.addListener
(chart, 'ready', function () {
chart_div.innerHTML = '<img src="' +
chart.getImageURI() + '">';
console.log(chart_div.innerHTML);
});

chart.draw(data, options);
}
google.charts.setOnLoadCallback(drawChart);
</script>
</body>
</html>
```

```
// This is a JavaScript script
// This is the file for the Geochart of
// bankruptcy percentage of 2016 by region

<html>
  <head>
    <script type='text/javascript' src='https://www.gstatic.com/
    charts/loader.js'></script>
    <script type='text/javascript'>
    google.charts.load('current', {
    'packages': ['geochart'],
    // Note: you will need to get a mapsApiKey for your project.
     // See: https://developers.google.com/chart/interactive/
     docs/basic_load_libs#load-settings
    'mapsApiKey': myKey
    });
    google.charts.setOnLoadCallback(drawMarkersMap);

    function drawMarkersMap() {
    var data = google.visualization.arrayToDataTable([
```

['County', 'Bankruptcy percentages'], ['ØSTFOLD', 0.2532], ['Akershus', 0.2015], ['Oslo', 0.2029], ['Hedmark', 0.2152], ['OPPLAND', 0.2088], ['BUSKERUD', 0.2372], ['VESTFOLD', 0.2040], ['TELEMARK', 0.2079], ['AUST-AGDER', 0.1896], ['VEST-AGDER', 0.2118], ['ROGALAND', 0.1913], ['HORDALAND', 0.2488], ['SOGN OG FJORDANE', 0.1361], ['MØRE OG ROMSDAL', 0.2075], ['NORDLAND', 0.2077], ['TROMS', 0.2074], ['FINNMARK', 0.2125], ['NORD-TRNDELAG', 0.1844], ['Sør-Trøndelag', 0.1844],

```
    ]);

      var options = {
        region: 'NO',
        resolution: 'provinces',
        displayMode: 'region',
        colorAxis: {colors: ['#FFFF00', 'red']},
      };

      var chart = new google.visualization.GeoChart(document.
      getElementById('chart_div'));
      chart.draw(data, options);
    };
    </script>
  </head>
  <body>
```

```
    <div id="chart_div" style="width: 900px;
    height: 500px;"></div>
  </body>
</html>


// This is a JavaScript script
// This is the file for the Geochart of the
// number of bankruptcy companies of 2016 by region

<html>
  <head>
    <script type='text/javascript' src='https://www.
    gstatic.com/charts/loader.js'></script>
    <script type='text/javascript'>
     google.charts.load('current', {
       'packages': ['geochart'],
       // Note: you will need to get a mapsApiKey
       // for your project.
       // See: https://developers.google.com/chart/
       interactive/docs/basic_load_libs#load-settings
       'mapsApiKey': myKey
     });
     google.charts.setOnLoadCallback(drawRegionsMap);

      function drawRegionsMap() {
      var data = google.visualization.arrayToDataTable([
```

['Region', 'Bankruptcy counts'], ['ØSTFOLD', 337], ['Akershus', 0660], ['Oslo', 1186], ['Hedmark', 178], ['OPPLAND', 157], ['BUSKERUD', 337], ['VESTFOLD', 262], ['TELEMARK', 184], ['AUST-AGDER', 113], ['VEST-AGDER', 229], ['ROGALAND', 475], ['HORDALAND', 734], ['SOGN OG FJORDANE', 60], ['MØRE OG ROMSDAL', 239], ['NORDLAND', 226], ['TROMS', 145], ['FINNMARK', 78], ['NORD-TRØNDELAG', 408], ['Sør-Trøndelag', 408],

```
    ]);

    var options = {
        region: 'NO',
        resolution: 'provinces',
        displayMode: 'region',
        colorAxis: {colors: ['#FFFF00', 'red']},
      };

      var chart = new google.visualization.GeoChart(document.
      getElementById('chart_div'));
```

```
      chart.draw(data, options);
    };
    </script>
  </head>
  <body>
    <div id="chart_div" style="width: 900px;
    height: 500px;"></div>
  </body>
</html>


// This is a JavaScript script
// This is the file for the Geochart of the
// bankruptcy percentage of 2018 in the form of Marker

<html>
  <head>
    <script type='text/javascript' src='https://www.gstatic.com/
    charts/loader.js'></script>
    <script type='text/javascript'>
      google.charts.load('current', {
        'packages': ['geochart'],
        // Note: you will need to get a mapsApiKey for your project.
        // See: https://developers.google.com/chart/interactive/
        docs/basic_load_libs#load-settings
        'mapsApiKey': myKey
      });
      google.charts.setOnLoadCallback(drawMarkersMap);

       function drawMarkersMap() {
       var data = google.visualization.arrayToDataTable([
```

['City', 'Bankruptcy', 'Bankruptcy percentage'], ['Sarpsborg', 'fylke: ØSTFOLD', 0.2827], ['Fenstad', 'fylke: Akershus', 0.2426], ['The Fram Museum', 'fylke:Oslo', 0.2093], ['Deset', 'fylke: Hedmark', 0.2623], ['Otta', 'fylke: OPPLAND', 0.2102], ['Dagali', 'fylke: BUSKERUD', 0.2493], ['Tønsberg', 'fylke: VESTFOLD', 0.2311], ['Seljord', 'fylke: TELEMARK', 0.2523], ['Dølemo', 'fylke: AUST-AGDER', 0.2121], ['Eiken', 'fylke: VEST-AGDER', 0.2234], ['Stavanger', 'fylke: ROGALAND', 0.2232], ['Bergen', 'fylke: HORDALAND', 0.2758], ['Hermansverk', 'fylke: SOGN OG FJORDANE', 0.1832], ['Molde', 'fylke: MØRE OG ROMSDAL', 0.2300], ['Bodø', 'fylke: NORDLAND', 0.1966], ['Tromsø', 'fylke: TROMS', 0.1919], ['Vadsø', 'fylke: FINNMARK', 0.2338], ['Steinkjer', 'fylke: TRØNDELAG', 0.1981],

```
       ]);

       var options = {
```

```
        region: 'NO',
        resolution: 'provinces',
        displayMode: 'markers',
        colorAxis: {colors: ['green', 'red']},
        sizeAxis: {minSize: 2.5, maxSize:6.5},
      };

      var chart = new google.visualization.GeoChart(document.
      getElementById('chart_div'));
      chart.draw(data, options);
    };
    </script>
  </head>
  <body>
  <div id="chart_div" style="width: 900px;
    height: 500px;"></div>
  </body>
</html>


// This is a JavaScript script
// This is the file for the Geochart of the number of
// bankruptcy companies of 2018 in the form of Marker

<html>
  <head>
    <script type='text/javascript' src='https://www.gstatic.com/
    charts/loader.js'></script>
    <script type='text/javascript'>
     google.charts.load('current', {
       'packages': ['geochart'],
       // Note: you will need to get a mapsApiKey for your project.
       // See: https://developers.google.com/chart/interactive/
       docs/basic_load_libs#load-settings
       'mapsApiKey': myKey
    });
    google.charts.setOnLoadCallback(drawMarkersMap);

    function drawMarkersMap() {
    var data = google.visualization.arrayToDataTable([
```

['City', 'Bankruptcy', 'Bankruptcy counts'], ['Sarpsborg', 'fylke: ØSTFOLD', 370], ['Fenstad', 'fylke: Akershus', 765], ['The Fram Museum', 'fylke: Oslo', 1140], ['Deset', 'fylke: Hedmark', 192], ['Otta', 'fylke: OPPLAND', 157], ['Dagali', 'fylke: BUSKERUD', 368], ['Tønsberg', 'fylke: VESTFOLD', 287], ['Seljord', 'fylke: TELEMARK', 216], ['Dølemo', 'fylke: AUST-AGDER', 112], ['Eiken', 'fylke: VEST-AGDER', 212], ['Sta-

vanger', 'fylke: ROGALAND', 507], ['Bergen', 'fylke: HORDALAND', 734], ['Hermansverk', 'fylke: SOGN OG FJORDANE', 85], ['Molde', 'fylke: MØRE OG ROMSDAL', 265], ['Bodø', 'fylke: NORDLAND', 208], ['Tromsø', 'fylke: TROMS', 147], ['Vadsø', 'fylke: FINNMARK', 90], ['Steinkjer', 'fylke: TRØNDELAG', 423],

```
        ]);

        var options = {
          region: 'NO',
          resolution: 'provinces',
          displayMode: 'markers',
          colorAxis: {colors: ['green', 'red']},
          sizeAxis: {minSize: 2.5, maxSize:6.5},
        };

        var chart = new google.visualization.GeoChart
        (document.getElementById('chart_div'));
        chart.draw(data, options);
      };
      </script>
    </head>
    <body>
      <div id="chart_div" style="width: 900px;
      height: 500px;"></div>
    </body>
</html>
```

A.3 Data Mining Experiments

```
# These are Python scripts running on the Jupyter of Anaconda
# This is the file to construct model on the training set of 2016
# and apply it on the testing set of 2016
# using the kNN algorithm

# coding: utf-8
import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

data = pd.read_csv("./Real Data/Real 2016.csv")
data.head()

data.shape

### Get the train and test data-set, with and without
sampling Train - Test data split without resampling

X = data.iloc[:, data.columns != 'Class'].values
y = data.iloc[:, data.columns == 'Class'].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X,y,test_size = 0.3, random_state = 0)

print("Original number transactions train dataset: ", len(X_train))
print("Original number transactions test dataset: ", len(X_test))
print("Total number of transactions: ", len(X_train)+len(X_test))

# Number of data points in the minority class
number_records_fraud = len(data[data.Class == 1])
fraud_indices = np.array(data[data.Class == 1].index)
print(number_records_fraud)
#print(data[data.Class == 1])

# Picking the indices of the normal classes
normal_indices = data[data.Class == 0].index

# Out of the indices we picked, randomly
select "x" number (number_records_fraud)
#np.random.choice(). By using this, the numbers of fraud
```

```
indices and non-fraud indices become equal.
random_normal_indices = np.random.choice(a = normal_indices,
size = number_records_fraud, replace = False)
random_normal_indices = np.array(random_normal_indices)

# Appending the 2 indices
under_sample_indices = np.concatenate([fraud_indices,
random_normal_indices])
print(under_sample_indices)

# Under sample dataset
under_sample_data = data.iloc[under_sample_indices,:]

X_undersample = under_sample_data.iloc[:,
under_sample_data.columns != 'Class']
y_undersample = under_sample_data.iloc[:,
under_sample_data.columns == 'Class']

# Showing ratio
print("Percentage of normal transactions: ", len(
under_sample_data[under_sample_data.Class == 0])
/len(under_sample_data))
print("Percentage of fraud transactions: ", len(
under_sample_data[under_sample_data.Class == 1])
/len(under_sample_data))
print("Total number of transactions in resampled
data: ", len(under_sample_data))

# Undersampled dataset
X_train_undersample, X_test_undersample, y_train_undersample,
y_test_undersample = train_test_split(X_undersample,
y_undersample,test_size = 0.3,random_state = 0)
print("")
print("Number transactions train dataset: ",
len(X_train_undersample))
print("Number transactions test dataset: ",
len(X_test_undersample))
print("Total number of transactions: ", len(X_train_undersample
)+len(X_test_undersample))

from sklearn.preprocessing import StandardScaler

#KNN
from sklearn.neighbors import KNeighborsClassifier
```

```
#standard the data
ss = StandardScaler()
X_train_undersample = ss.fit_transform(X_train_undersample)
X_test_undersample = ss.transform(X_test_undersample)

knn = KNeighborsClassifier()
#start model training
knn.fit(X_train_undersample, y_train_undersample)
# predict the classification using the model
y_predict_undersample = knn.predict(X_test_undersample)

from sklearn.metrics import confusion_matrix,
precision_recall_curve,auc,roc_auc_score,
roc_curve,recall_score,classification_report

# Compute and plot confusion matrix
cnf_matrix = confusion_matrix(y_test_undersample,
y_predict_undersample)

#Model overall accuracy
print("the Model overall accuracy is :",(cnf_matrix[1,1]
+cnf_matrix[0,0])/(cnf_matrix[1,1]+cnf_matrix[1,0]+
cnf_matrix[1,0]+cnf_matrix[0,0]))
print()
print("the recall of fraud is :",cnf_matrix[1,1]/
(cnf_matrix[1,1]+cnf_matrix[1,0]))
print("the precision of fraud is :",cnf_matrix[1,1]/
(cnf_matrix[1,1]+cnf_matrix[0,1]))
print()
print("the recall of normal is :",cnf_matrix[0,0]/
(cnf_matrix[0,0]+cnf_matrix[0,1]))
print("the precision of normal is :",cnf_matrix[0,0]/
(cnf_matrix[0,0]+cnf_matrix[1,0]))

fig= plt.figure(figsize=(6,3))# to plot the graph
print("TP",cnf_matrix[1,1]) # no of fraud transaction
which are predicted fraud
print("TN",cnf_matrix[0,0]) # no.of normal transaction
which are predited normal
print("FP",cnf_matrix[0,1]) # no of normal transaction
which are predicted fraud
print("FN",cnf_matrix[1,0]) # no of fraud Transaction
which are predicted normal
sns.heatmap(cnf_matrix,cmap="coolwarm_r",annot=True,
linewidths=0.5)
```

```
plt.title("Confusion_matrix")
plt.xlabel("Predicted_class")
plt.ylabel("Real class")
plt.show()

#NEXT IS USE THE GLOBAL DATA

from sklearn.preprocessing import StandardScaler

#KNN
from sklearn.neighbors import KNeighborsClassifier

# standard the data
ss = StandardScaler()
X_train = ss.fit_transform(X_train)
X_test = ss.transform(X_test)

knn_ = KNeighborsClassifier(n_neighbors=4)
# train the model
knn_.fit(X_train, y_train)
# make predictions using the model and store the data
y_predict_ = knn_.predict(X_test)

# Compute and plot confusion matrix
cnf_matrix = confusion_matrix(y_test,y_predict_)

#Model overall accuracy
print("the Model overall accuracy is :",(cnf_matrix[1,1]
+cnf_matrix[0,0])/(cnf_matrix[1,1]+cnf_matrix[1,0]
+cnf_matrix[1,0]+cnf_matrix[0,0]))
print()
print("the recall of fraud is :",cnf_matrix[1,1]/
(cnf_matrix[1,1]+cnf_matrix[1,0]))
print("the precision of fraud is :",cnf_matrix[1,1]/
(cnf_matrix[1,1]+cnf_matrix[0,1]))
print()
print("the recall of normal is :",cnf_matrix[0,0]/
(cnf_matrix[0,0]+cnf_matrix[0,1]))
print("the precision of normal is :",cnf_matrix[0,0]/
(cnf_matrix[0,0]+cnf_matrix[1,0]))

fig= plt.figure(figsize=(6,3))# to plot the graph
print("TP",cnf_matrix[1,1]) # no of fraud transaction
which are predicted fraud
print("TN",cnf_matrix[0,0]) # no.of normal transaction
```

```
which are predited normal
print("FP",cnf_matrix[0,1]) # no of normal transaction
which are predicted fraud
print("FN",cnf_matrix[1,0]) # no of fraud Transaction
which are predicted normal
sns.heatmap(cnf_matrix,cmap="coolwarm_r",annot=True,linewidths=0.5)
plt.title("Confusion_matrix using KNN in 2016")
plt.xlabel("Predicted_class")
plt.ylabel("Real class")
plt.show()


# This is the file to construct model on the data of 2018
# and apply it on the data of 2018
# This code use Random Forest, Gradient Boosting
# and the ensemble learning of Majority Voting

# coding: utf-8
import numpy as np # linear algebra
import pandas as pd # data processing
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

data16 = pd.read_csv("../Real Data/Real 2016.csv")
data16.head()

data16.shape

### Get the train and test data-set, with and without sampling
Train - Test data split without resampling
X_train = data16.iloc[:, data16.columns != 'Class'].values
y_train = data16.iloc[:, data16.columns == 'Class'].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split

# Number of data points in the minority class
number_records_fraud = len(data16[data16.Class == 1])
fraud_indices = np.array(data16[data16.Class == 1].index)
print(number_records_fraud)
#print(data[data.Class == 1])

# Picking the indices of the normal classes
normal_indices = data16[data16.Class == 0].index
```

```
# Out of the indices we picked, randomly select "x"
number (number_records_fraud)
#np.random.choice(). By using this, the numbers of fraud
indices and non-fraud indices become equal.
random_normal_indices = np.random.choice(a = normal_indices,
size = number_records_fraud, replace = False)
random_normal_indices = np.array(random_normal_indices)

# Appending the 2 indices
under_sample_indices = np.concatenate([fraud_indices,
random_normal_indices])
print(under_sample_indices)

# Under sample dataset
under_sample_data16 = data16.iloc[under_sample_indices,:]

X_undersample = under_sample_data16.iloc[:,
under_sample_data16.columns != 'Class']
y_undersample = under_sample_data16.iloc[:,
under_sample_data16.columns == 'Class']

# Showing ratio
print("Percentage of normal transactions: ",
len(under_sample_data16[under_sample_data16.Class ==0]
)/len(under_sample_data16))
print("Percentage of fraud transactions: ",
len(under_sample_data16[under_sample_data16.Class == 1]
)/len(under_sample_data16))
print("Total number of transactions in resampled data: ",
len(under_sample_data16))

# Undersampled dataset
X_train_undersample, X_test_undersample, y_train_undersample,
y_test_undersample = train_test_split(X_undersample,y_undersample,
test_size = 0,random_state = 0)
print("")
print("Number transactions train dataset: ", len(X_train_undersample))
print("Number transactions test dataset: ", len(X_test_undersample))
print("Total number of transactions: ", len(X_train_undersample)+
len(X_test_undersample))

from sklearn.metrics import confusion_matrix,precision_recall_curve,
auc,roc_auc_score,roc_curve,recall_score,classification_report
```

```
# Notice: we have used all the data in 2016 as training data set.
# Now we're going to train the model on the 2016 data,
# and test them on the 2018 data set.

# ### First step: make the 2018 data as the test data.

data18 = pd.read_csv("../Real Data/Real 2018.csv")
data18.head()

data18.shape

X_test = data18.iloc[:, data18.columns != 'Class'].values
y_test = data18.iloc[:, data18.columns == 'Class'].values


# ### Second step: construct model on the 2016 data and apply
it upon the 2018 data

# Voting Ensemble for Classification
import pandas
from sklearn import model_selection

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier

#results = model_selection.cross_val_score(ensemble,
X_train_undersample,y_train_undersample.values.ravel(), cv=kfold)
#print(results.mean())

gb = GradientBoostingClassifier(n_estimators=100)
rf = RandomForestClassifier(n_estimators=100)

#seed = 7
#kfold = model_selection.KFold(n_splits=10, random_state=seed)
# create the sub models
estimators = []
estimators.append(('GB', gb))
estimators.append(('RF', rf))

# create the ensemble model
ensemble = VotingClassifier(estimators,voting='hard')
ensemble.fit(X_train_undersample,y_train_undersample.values.ravel())

y_pred_under = ensemble.predict(X_test)
```

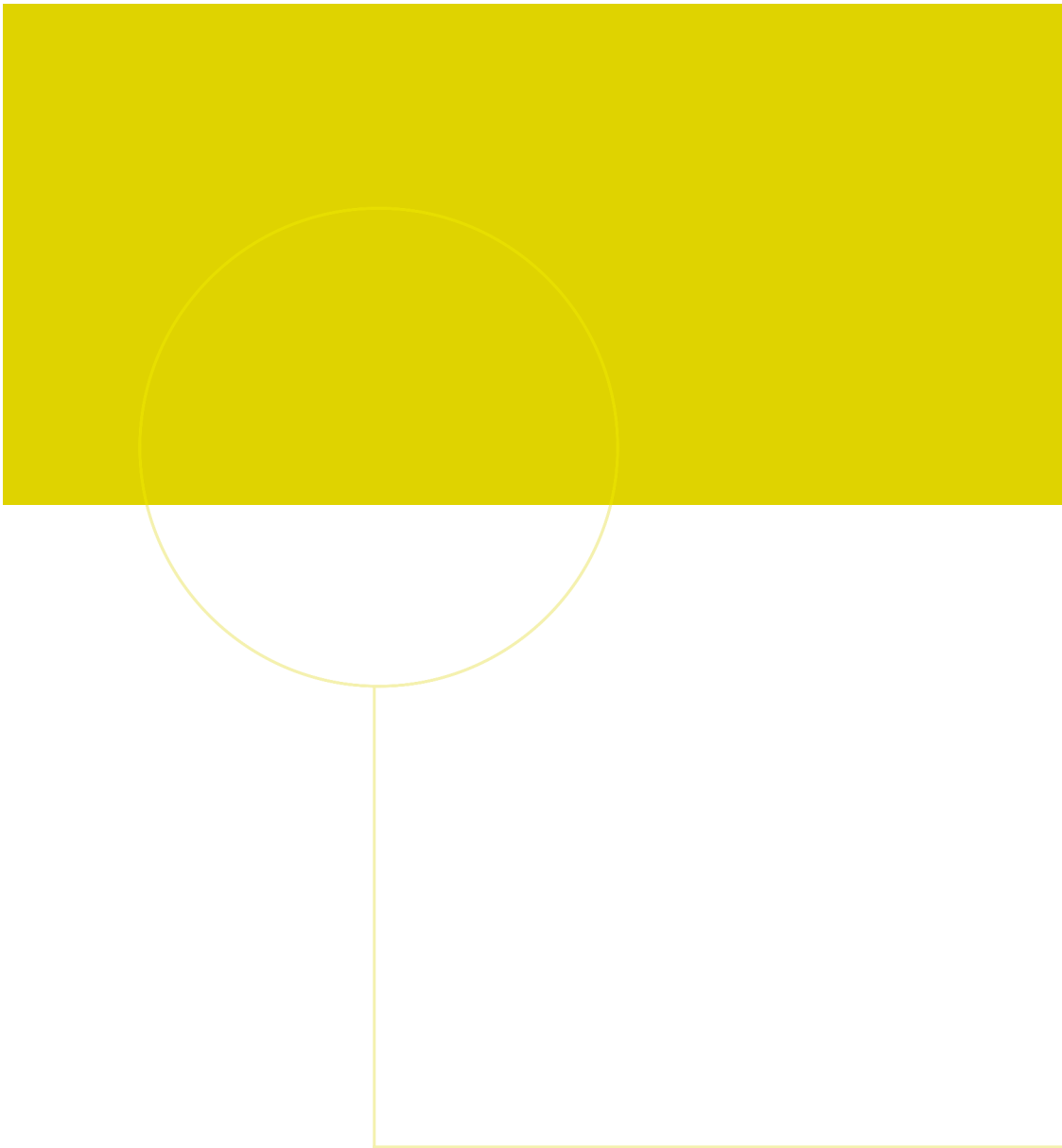
```
# Compute and plot confusion matrix
cnf_matrix = confusion_matrix(y_test,y_pred_under)

#Model overall accuracy
print("the Model overall accuracy is :",(cnf_matrix[1,1]
+cnf_matrix[0,0])/(cnf_matrix[1,1]+
cnf_matrix[1,0]+cnf_matrix[1,0]+cnf_matrix[0,0]))
print()
print("the recall of fraud is :",cnf_matrix[1,1]
/(cnf_matrix[1,1]+cnf_matrix[1,0]))
print("the precision of fraud is :",cnf_matrix[1,1]
/(cnf_matrix[1,1]+cnf_matrix[0,1]))
print()
print("the recall of normal is :",cnf_matrix[0,0]
/(cnf_matrix[0,0]+cnf_matrix[0,1]))
print("the precision of normal is :",cnf_matrix[0,0]
/(cnf_matrix[0,0]+cnf_matrix[1,0]))

fig= plt.figure(figsize=(6,3))# to plot the graph
print("TP",cnf_matrix[1,1]) # no of fraud transaction which
are predicted fraud
print("TN",cnf_matrix[0,0]) # no.of normal transaction which
are predited normal
print("FP",cnf_matrix[0,1]) # no of normal transaction which
are predicted fraud
print("FN",cnf_matrix[1,0]) # no of fraud Transaction which
are predicted normal
sns.heatmap(cnf_matrix,cmap="coolwarm_r",annot=True,linewidths=0.5)
plt.title("Confusion_matrix")
plt.xlabel("Predicted_class")
plt.ylabel("Real class")
plt.show()


#NEXT IS USE THE GLOBAL DATA

#seed = 7
#kfold = model_selection.KFold(n_splits=10, random_state=seed)
# create the sub models

from sklearn.preprocessing import StandardScaler

# standard
ss = StandardScaler()
X_train = ss.fit_transform(X_train)
```

```
X_test = ss.transform(X_test)

# create the ensemble model
ensemble_ = VotingClassifier(estimators,voting='soft')
ensemble_.fit(X_train, y_train)

y_predict = ensemble_.predict(X_test)
# Compute and plot confusion matrix
cnf_matrix = confusion_matrix(y_test,y_predict)


# Compute and plot confusion matrix
cnf_matrix = confusion_matrix(y_test,y_predict)

#Model overall accuracy
print("the Model overall accuracy is :",(cnf_matrix[1,1]
+cnf_matrix[0,0])/(cnf_matrix[1,1]+
cnf_matrix[1,0]+cnf_matrix[1,0]+cnf_matrix[0,0]))
print()
print("the recall of fraud is :",cnf_matrix[1,1]
/(cnf_matrix[1,1]+cnf_matrix[1,0]))
print("the precision of fraud is :",cnf_matrix[1,1]
/(cnf_matrix[1,1]+cnf_matrix[0,1]))
print()
print("the recall of normal is :",cnf_matrix[0,0]
/(cnf_matrix[0,0]+cnf_matrix[0,1]))
print("the precision of normal is :",cnf_matrix[0,0]
/(cnf_matrix[0,0]+cnf_matrix[1,0]))

fig= plt.figure(figsize=(6,3))# to plot the graph
print("TP",cnf_matrix[1,1]) # no of fraud transaction
which are predicted fraud
print("TN",cnf_matrix[0,0]) # no.of normal transaction
which are predited normal
print("FP",cnf_matrix[0,1]) # no of normal transaction
which are predicted fraud
print("FN",cnf_matrix[1,0]) # no of fraud Transaction
which are predicted normal
sns.heatmap(cnf_matrix,cmap="coolwarm_r",
annot=True,linewidths=0.5)
plt.title("Confusion_matrix")
plt.xlabel("Predicted_class")
plt.ylabel("Real class")
plt.show()
```

Appendix B: Publications

Title: Application of data mining algorithms on a new data set of companies in Norway (In Progress)

Abstract: In recent years, more and more researchers have been focusing on the research of bankruptcy prediction. However, traditional methods based on statistical models may not be able to deal with relevant data sets, which are becoming more and more sophisticated than before. At the same time, new methods of data mining have been springing up for the last few decades. Therefore, in this project, we try to discuss some data mining algorithms and apply those algorithms upon a new data new about the bankruptcy situations of Norway for bankruptcy prediction.



NTNU
Norwegian University of
Science and Technology