CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION TO MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.2 MACHINE LEARNING IN BANKING & FINANCE

Until recently, only the hedge funds were the primary users of AI and ML in Finance, but the last few years have seen the applications of ML spreading to various other areas, including banks, fintech, regulators, and insurance firms, to name a few.

Right from speeding up the underwriting process, portfolio composition and optimization, model validation, Robo-advising, market impact analysis, to offering alternative credit reporting methods, the different use cases of Artificial Intelligence and Machine Learning are having a significant impact on the financial sector.

The finance industry, including the banks, trading, and fintech firms are rapidly deploying machine algorithms to automate time-consuming, mundane processes, and offering a far more streamlined and personalized customer experience.

1.3 BENEFITS OF MACHINE LEARNING IN BANKING

✓ **Greater Automation and Improved Productivity**

Machine Learning can easily handle mundane tasks, allowing managers more time to work on more sophisticated challenges than repetitive paperwork. Automation across the entire organization will ultimately lead to greater profits.

✓ **Personalized Customer Service**

Automated solutions with Big Data capabilities can track and store as much information about the bank's customers as needed, providing the most precise and personalized customer experience. Optimizing the customer footprint allows banks to leverage analytical capabilities of Artificial Intelligence and Machine Learning to detect even the most subtle tendencies in customer behavior, which helps create a more personalized experience for each individual client.

✓ **More precise Risk Assessment**

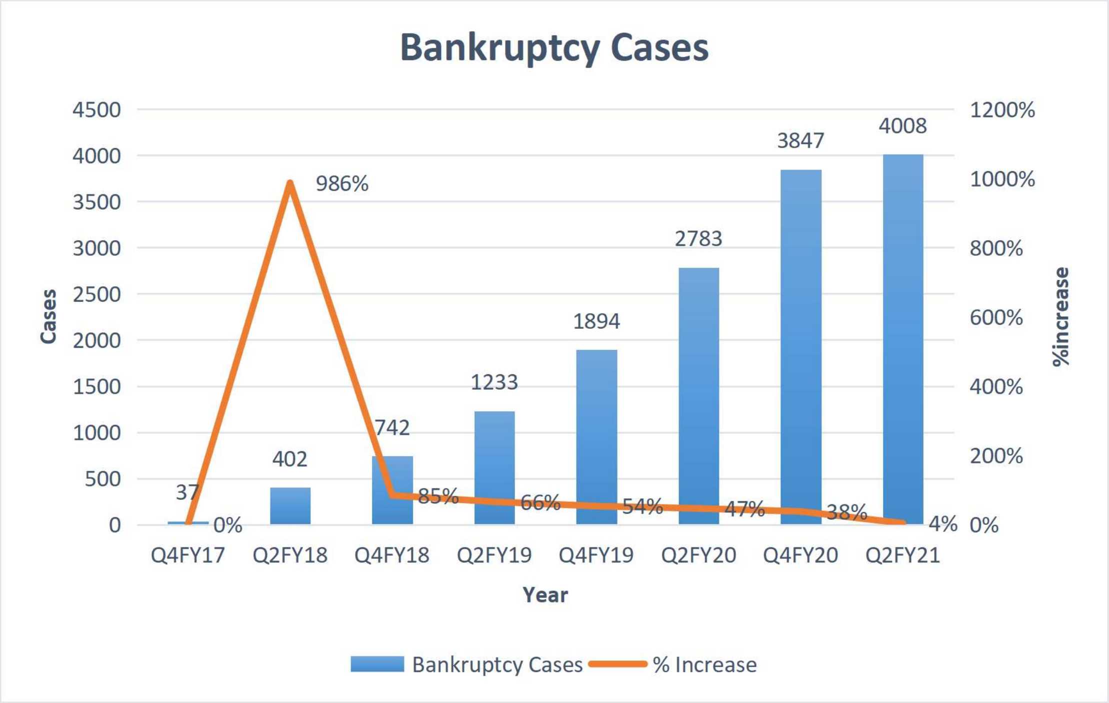
Having an accurate digital footprint of each customer also can help banks reduce uncertainty for managers working with individual clients. The automated system is more accurate than a human in such areas as analysis of loan underwriting, eliminating any possible human bias.

✓ **Advanced Fraud Detection and Prevention**

This is probably the top benefit of AI/ML for any financial institution because there has historically been, and will continue to be, criminals who are devising methods to commit financial fraud. Fortunately, there are currently a wide range of proven methods and techniques of ML-powered Fraud Detection on the market.

1.4 INTRODUCTION TO BANKRUPTCY PREDICTION

Bankruptcy is the most crucial process in Financial risk management. The bankruptcy cases keep on increasing every financial quarter in India. The number of cases are slowed down to 161 in the first half of FY21 compared with 889 cases admitted during the same period in FY20. With this, the September quarter witnessed a drop of around 85 per cent compared with the previous year, said a CARE Ratings report. This can be attributed to the suspension of fresh bankruptcy proceedings for COVID-19 defaults. Even though, Bankruptcy cases keep on increasing.



Bankruptcy prediction have been known as critical study and have been studied widely in the finance literature. The research is important for lending decision and profitability of financial institutions. Bank needs to predict the potential of organization before lending to avoid failures. Thus, bankruptcy prediction become eminent for financial institutions.

1.5 OBJECTIVES

The main endowment of this paper is:

- To provide a best predictor for the bank to avoid failures
- To provide a relatable factors regarding bankruptcy.
- Covering different aspects of bankruptcy assessment
- Identifying key factors for bankruptcy
- To provide higher accuracy

CHAPTER 2
LITERATURE REVIEW

2.1 LITERATURE SURVEY

I have reviewed several research papers and articles related to Bankruptcy prediction. Below are the brief descriptions of it:

A BANKRUPTCY PREDICTION MODEL USING RANDOM FOREST

Shreya Joshi, Rachana Ramesh, Shagufta Tahsildar

Bankruptcy Prediction is important for an organization as well as decision makers such as financiers and investors. The Machine Learning tool used for prediction and various other factors are essential in creating an efficient prediction model. The dataset includes financial ratios as attributes that are derived from the financial statements of various companies.

The most influencing ratios are selected on the basis of the Genetic Algorithm which filters out the most weightage ratios for determining the bankruptcy. These ratios of different companies are fed as an input to train the model been implemented in R.

Random Forest prediction algorithm is trained on a dataset consisting the ratios and finally the model can predict accurate results on various test cases. This algorithm is implemented in RStudio. The Graphical User Interface required for the interaction with the user was developed with the help of the Shiny Web App which is an integrated tool available in the RStudio.

ENTERPRISE BANKRUPTCY PREDICTION USING NOISY-TOLERANT SUPPORT VECTOR MACHINE

International Seminar on Information Technology and Management Engineering (2008)

Zhong Gao , Meng Cui and Lai-Man Po

Increasing the accuracy of bankruptcy forecast requires a powerful learning machine algorithm capable of good generalization on financial data. Therefore, classification algorithms like Support Vector Machine (SVM) are popular for modeling. However, making inferences and choosing appropriate responses based on incomplete, uncertainty and noisy data is challenging in financial settings particularly in bankruptcy prediction.

In this paper, they proposed a approach which uses a novel Support Vector Machine and K-nearest neighbor (KNN-SVM) to remove noisy training examples. The experimental results show that the generalization performance and the accuracy of classification are improved 12% significantly compared to that of the traditional SVM classifier, and adapt to engineering applications.

This combined classifier based on these features could lead to a more satisfied results in enterprise bankruptcy prediction, thus establishing the potential usefulness in computerized system application.

A HYBRID METAHEURISTIC METHOD IN TRAINING ARTIFICIAL NEURAL NETWORK FOR BANKRUPTCY PREDICTION

IEEE Access Journal (2020)

Abdollah Ansari, Ibrahim Said Ahmad, Azuraliza Abu Bakar, Mohd Ridzwan Yaakub

Corporate bankruptcy prediction is an important task in the determination of corporate solvency, such whether a company can meet up to its financial obligations or not. In recent years, machine learning techniques, particularly Artificial Neural Network (ANN) was used mostly, since it have proven to be a good predictor.

A critical process in learning a network is weight training. ANN is more efficient in a weight training process. Many studies improved ANN's weight training using metaheuristic algorithms such as Evolutionary Algorithms (EA), and Swarm Intelligence (SI) approaches for bankruptcy prediction.

In this study, two metaheuristics algorithms, Magnetic Optimization Algorithm (MOA) and Particle Swarm Optimization (PSO), have been enhanced through hybridization to propose a new method MOA-PSO. Hybrid algorithms have been proven to be capable of solving optimization problems faster, with better accuracy. The proposed hybrid MOA-PSO algorithm shows results with a faster and more accurate prediction, with 99.7% accuracy.

Future work is to evaluate the approach with more recent but equally reputable datasets. Other extensions of MOA like Functional Sized Population MOA (FSMOA) could also be investigated for bankruptcy prediction.

A MCDM-BASED EVALUATION APPROACH FOR IMBALANCED CLASSIFICATION METHODS IN FINANCIAL RISK PREDICTION

IEEE Access Journal (2019)
Yong Ming Song, Yi Peng

Various classifiers have been proposed for financial risk prediction. This paper proposed a multi-criteria decision making (MCDM) based approach to evaluate bankruptcy risk prediction by considering multiple performance metrics simultaneously.

An experiment was designed to evaluate the proposed approach using 7 financial imbalanced binary data sets from the UCI Machine Learning repository. The experiment makes use of four standard classifiers (LR, SVM, MLP and C4.5) combined with three groups of imbalanced techniques, namely cost-sensitive learning, resampling (RUS and SMOTE), and hybrid approaches.

Six frequently used performance metrics for imbalanced learning: G-mean, F-measure, AUC, FP rate, FN rate, and time were used in the experiment. TOPSIS, MCDM method, was applied to rank the imbalanced learning approaches. The final ranking results indicate that SMOTE-based ensemble classifiers outperform other groups of imbalanced learning algorithms, SMOTEBoost-C4.5, SMOTE-C4.5, and SMOT-MLP were ranked as the top three classifiers based on their performances on the six criteria.

The proposed MCDM-based evaluation approach for imbalanced learning approaches can make up the shortfall of single criteria evaluation. Establishing an assembled algorithm based on MCDM method to classify the financial imbalanced data sets is the future work.

A HYBRID SWITCHING PSO ALGORITHM AND SUPPORT VECTOR MACHINES FOR BANKRUPTCY PREDICTION

International Conference on Mechatronics and Control (2014)
Yang Lu, Jingfu Zhu, Nan Zhang, Qing Shao

In this paper, a hybrid switching PSO and SVM algorithm used for predicting bankruptcy. Initially, they processed the bankruptcy sample data sets come from UCI Machine Learning Repository. Then the parameters of SVM optimized by a recently proposed switching PSO algorithm. In the end, the incorporated model has been successfully applied to predict bankruptcy.

It was shown that the proposed algorithm gives much improved performance over the traditional SVM-based methods combined with genetic algorithm (GA) or particle swarm optimization (PSO).

BANKRUPTCY PREDICTION USING EXTREME LEARNING MACHINE AND FINANCIAL EXPERTISE

IEEE Access Journal (2019)
Qi Yu, Yoan Miche, Eric Séverin, Amaury Lendasse

Bankruptcy prediction has been widely studied as a binary classification problem using financial ratios methodologies. In this paper, Leave-One-Out-Incremental Extreme Learning Machine (LOO-IELM) was used. LOO-IELM operates in an incremental way to avoid inefficient and unnecessary calculations and stops automatically with the neurons of which the number is unknown.

Moreover, Combo method and further Ensemble model are investigated based on different LOO-IELM models and the specific financial indicators. These indicators are chosen using different strategies according to the financial expertise. The entire process has shown its good performance with a very fast speed, and also helps to interpret the model and the special ratios.

A MULTI-INDUSTRY BANKRUPTCY PREDICTION MODEL USING BACK-PROPAGATION NEURAL NETWORK AND MULTIVARIATE DISCRIMINANT ANALYSIS

Expert system application science direct journal (2012)
Sangjae Lee, Wu Sung Choi

The accurate prediction of corporate bankruptcy for the firms in different industries was a concern to investors and creditors. This paper shows a multi-industry investigation of the bankruptcy of Korean companies using back-propagation neural network (BNN). The industries include construction, retail, and manufacturing.

The prediction accuracy of BNN is compared to that of multivariate discriminant analysis. The prediction accuracy was 6–12% greater for industry specific prediction model than the general model which has prediction accuracy of 81.43 for BNN and 74.82 for MDA. The prediction accuracy of bankruptcy using BNN is greater than that of MDA.

BANKRUPTCY PREDICTION BY DEEP LEARNING

IEEE Access Journal (2020)
Giovanni Cialone

Among all models there are the most used ones such as Multiple Discriminant Analysis and Logistic Regression, machine learning techniques, such as Random Forests, Boosting and NN. In recent years, advanced machine learning techniques, in particular the Deep Neural Networks, have been studied extensively.

In this paper, the large amount of data for small and medium-sized Italian companies collected from financial and income statements have been processed , applying two different Neural Networks architectures: (i) a deep sequential model and (ii) a Convolutional architecture.

The model with the best performances was the Sequential Architecture which reached the highest AUC value, 0.90 and the highest sensibility 0.8205. The CNN Architecture showed the best specificity (numbers of True Negatives caputered). Ultimately, the models can find wider application, not only to the italian case but also to other countries where accounting standards are similar and the inputs variables have same metrics.

SELECTING BANKRUPTCY PREDICTORS USING A SUPPORT VECTOR MACHINE APPROACH

Alan Fan, Marimuthu Palaniswami

Conventional Neural Network approach has been found useful in predicting corporate distress from financial statements. In this paper, a Support Vector Machine approach was used. A new way of selecting bankruptcy predictors is shown, using the Euclidean distance based criterion calculated within the SVM kernel.

They examined the practicality and performance of the Support Vector Machine approach to predict Australian business failure. Empirical results showed that SVM was competitive and outperformed other classifiers in terms of generalization performance. They proposed an Euclidean distance based input selection criterion, which can provide a selection of variables that tends to discriminate within the SVM kernel used.

USING NEURAL NETWORK ENSEMBLES FOR BANKRUPTCY PREDICTION AND CREDIT SCORING

Expert system with application (2008)

Chih-Fong Tsai , Jhen-Wei Wu

Bankruptcy prediction and credit scoring have long been regarded as critical topics and have been studied extensively in the accounting and finance literature. Artificial intelligence and machine learning techniques have been used to solve these financial decision-making problems.

In this paper, they compared the performance of the single neural network classifier with the (diversified) multiple neural network classifiers over three datasets for the bankruptcy prediction and credit scoring problems. Theoretically, multiple classifiers should perform better than single classifiers. However, regarding the experimental results of average prediction accuracy, multiple neural network classifiers do not outperform a single best neural network classifier in many cases.

In particular, the results imply that the single best neural network classifier is more suitable than multiple or diversified multiple neural network classifiers for the bankruptcy prediction and credit scoring domains. On the other hand, by examining the Type I and Type II errors of these classifiers, there is no exact winner.

Regarding the experimental results, there are two issues to be discussed that multiple classifiers do not outperform single best classifiers. First, the divided training datasets may be too little to make the multiple classifiers and diversified multiple classifiers to perform worse. Second, in the binary classification domain problem as bankruptcy prediction and credit scoring, single classifiers may be a more stable model. In other words, the multiple classifiers and diversified multiple classifiers may not perform better in the binary classification problem.

AN ANALYTICAL APPROACH FOR BANKRUPTCY PREDICTION USING BIG DATA AND MACHINE LEARNING

Journal of Theoretical and Applied Information Technology (2019)

Ojini Devi, Dr. Y. Radhika

Bankruptcy is defined as a legal procedure used to claim the identity of an organization or a person on the basis of their creditworthiness and debtor. In this paper, effective predictive model has been proposed using big data analytics and Naive bayes algorithm.

The big data has been successfully stored in hadoop H-base platform and effective feature set are extracted from big data. These feature sets are further processed through naive bayes classifier and apache mahout platform for validation. The results obtained from the proposed model shows that naive bayes classifier technique along with big data hadoop tool is successful in determining bankruptcy prediction with ratio of 0.784.

Furthermore, this proposed system is proved to be an efficient tool for bank supervisions and financiers for early detection of risk profile which may arise due to bankruptcy.

MACHINE LEARNING MODELS AND BANKRUPTCY PREDICTION

Expert system with application (2017)

Flavio Barbozaa, Herbert Kimura, Edward Altman

In this study, they tested machine learning models (support vector machines, bagging, boosting, and random forest) to predict bankruptcy one year prior to the event, and compared performance with results from discriminant analysis, logistic regression, and neural networks. They used data from 1985 to 2013 on North American firms, integrating information from the Salomon Center database and Compustat, analysing more than 10,000 firm-year observations.

Comparing the best models, the machine learning technique related to random forest led to 87% accuracy, logistic regression and linear discriminant analysis led to 69%

and 50% accuracy, respectively. Bagging, boosting, and random forest models outperform the others techniques.

Future studies should extend the analysis to incorporate the growth rates and/or time effects of all variables, including the growth measures to evaluate the impact of time on default events.

A PREDICTIVE SYSTEM FOR DETECTION OF BANKRUPTCY USING MACHINE LEARNING TECHNIQUES

International Journal of Data Mining & Knowledge Management Process (2017)
Kalyan Nagaraj, Amulyashree Sridhar

Bankruptcy is a legal procedure that claims a person or organization as a debtor. In this perspective, different soft computing techniques can be employed to ascertain bankruptcy. This study proposes a bankruptcy prediction system to categorize the companies based on extent of risk using a decision support tool.

The results suggest that machine learning techniques can be implemented for prediction of bankruptcy. To serve the financial organizations for identifying risk oriented customers a prediction system was implemented. The predictive system helps to predict bankruptcy for a customer dataset based on the SVM model.

BANKRUPTCY PREDICTION USING MACHINE LEARNING

Journal of Mathematical Finance (2017)
Nanxi Wang

This paper proposed three relatively newly-developed methods for predicting bankruptcy based on real-life data. Support vector machine, neural network with dropout, and autoencoder are three relatively new models applied in bankruptcy prediction problems. Their accuracies outperform those of the three older models (robust logistic regression, inductive learning algorithms, genetic algorithms).

The improved aspects include the control for overfitting, the improved probability of finding the global maxima, and the ability to handle large feature spaces. This paper compared and concluded the progress of machine leaning models regarding bankruptcy prediction, and checked to see the performance of relatively new models in the context of bankruptcy prediction that have rarely been applied in that field. However, the three models also have drawbacks. SVM does not directly give probability estimates, but uses an expensive five-fold cross-validation instead.

Also, if the data sample is not big enough, especially when outnumbered by the number of features, SVM is likely to give bad performance. With dropout, the time to train the neural network will be 2 to 3 times longer than training a standard neural network. An autoencoder captures as much information as possible, not necessarily the relevant information. And this can be a problem when the most relevant information only makes up a small percent of the input. The solutions to overcome these drawbacks are yet to be found.

A HYBRID NEURAL NETWORK MODEL BASED ON IMPROVED PSO AND SA FOR BANKRUPTCY PREDICTION

International Journal of Computer Science (2019)
Fatima Zahra Azayite, Said Achchab

In this paper, they proposed a hybrid ANN to predict failure based on Particle Swarm Optimization and variables selection techniques. The methodology proposed the contribution of variables selection models by comparing Multivariate Discriminant Analysis, Logistic Regression and Decision Trees. The results show a high performance of Decision Trees as variables selection models for ANN to discriminate between Bankrupt and non-bankrupt firms.

Furthermore, they proposed a training algorithm improved PSO_SA to find the optimum topology. The training algorithm is based on the use of Simulated Annealing that allows jumping out from local minima and the hypothesis that learning is a process based not only on good experiences but also on bad experiences. The proposed algorithm gives high performances especially when applied to the hybrid DT and ANN model. This model is a good early warning system to use by investors and creditors.

2.2 LIMITATIONS OF EXISTING WORKS

The approach is not done with more recent equally reputable datasets [1]. The most weighted factors instead of analysis of every factors while determining the bankruptcy [2]. The accuracy is very less when compared to other hybrid algorithm [3]. The five financial ratios that selected are considered to be more effective for the KNN-SVM classifier, not for hybrid algorithms [4]. Other Algorithms provides very less accuracy averagely of 92.5%. Further, 99.728% accuracy achieved through hybrid algorithm based on neural network and two optimization algorithm. But, analysis done for past data for few observation instead recent reputable datasets.

CHAPTER 3
PROBLEM DEFINITION

3.1 EXISTING WORKS

On bankruptcy prediction, machine learning models are commonly used. The most used algorithms are Support Vector Machines (SVM), ANN, Gaussian Process (GP), Classification and Regression Tree (CART), Logistic Regression (Logit), Decision Tree (DT), Random Forest (RF), Linear Discriminant Analysis (LDA), and ensemble learning techniques. Further, recently many studies agree on the benefit of uniting mechanisms from different search methods. There is a widespread trend to design hybrid techniques in operations research and artificial intelligence.

3.2 PROBLEM DEFINITION

Bankruptcy is the most crucial process in Financial risk management. Bankruptcy prediction is a crucial task in the determination of organization's economic condition, that is, whether it can meet to its financial obligations or not. It is extensively researched because it includes a crucial impact on staff, customers, management, stockholders, bank disposition assessments, and profitableness. The research is important for lending decision and profitability of financial institutions. Bank needs to predict the potential of organization before lending to avoid failures. Thus, bankruptcy relatable factors and higher accuracy in recent data for bankruptcy prediction become eminent for financial institutions.

3.3 PROPOSED WORK

Ensemble models are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking). The proposed model will provide relatable factors regarding bankruptcy, it done using the predefined Qualitative bankruptcy data set and Quantitative bankruptcy data set in Ensemble model with different models. Then, the observations will be compared with some of the widely used classification models: Extreme Gradient Boosting for Decision Trees, Random Forests, Naïve Bayes, Balanced Bagging and Logistic Regression. Finally, we analyze and evaluate the performance of the models on the validation datasets using several metrics such as accuracy, precision, recall, etc., and rank the models accordingly. This will provide highly correlated factors and increase the consciousness of the banks.

3.4 FEASIBILITY STUDY

Feasibility study helps to determine whether a project is worth doing or not. It helps to determine whether a project should be taken or not on taking into account various parameters like operational feasibility, technical feasibility, economic feasibility, constraints, etc. It is basically a test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.
The following feasibility studies have been performed to measure the feasibility of the system:

- Operational feasibility
- Technical feasibility
- Economic feasibility

3.4.1 Operational Feasibility

Operational feasibility checks the operational scope of the system. The system under consideration should have enough operational reach. It is mainly related to the human organizational and political aspect. It is found out that the proposed system is easier to incorporate as most of the advancement in the system is done with the help of existing resources only.

3.4.2 Technical Feasibility

It deals with the specifying equipment and the software that successfully satisfies the user requirement. The model is developed in python.

3.4.3 Economic Feasibility

The new system will enhance the prediction accuracy, availability and would reduce failures. Considering the cost of implementing the system it will be a worthy investment for all Banks.

3.5 SOFTWARE REQUIREMENT SPECIFICATIONS

3.5.1 Introduction

Software used in this project is reliable according to the application. The model is designed using python.

3.5.2 Purpose

The purpose of this document is to provide an insight into the requirements for the development of this project. The content of this document serves as a collective record between the system and the developer, concerning the functionalities specified that the project exhibits.

3.5.3 Scope

The proposed model is designed to enhance the prediction of the bankruptcy using ensemble. It will also enhance the knowledge about features related to the bankruptcy.

3.5.4 Glossary

LR	:	Logistic Regression
SVM	:	Support Vector Machine
GNB	:	Guassian Naive Bayes
DT	:	Decision Tree
RF	:	Random Forest
XGB	:	Extreme Gradient Boosting
BB	:	Balanced Bagging

3.5.5 Hardware Requirements

Processor : Intel i3 or more Ram : 2 GB(min)

Hard Disk : 20 GB (min)

Monitor : 14” colour

Keyboard : 104 keys

3.5.6 Software Requirements

Programming Language : Python

Developing Tools : Jupyter notebook

Operating system : Windows 10

CHAPTER 4
METHODOLOGY

In the previous section, the problem statement of bankruptcy prediction was introduced. In this section, explained about step-by-step solution of how benchmark results for bankruptcy prediction was achieved. Firstly, Let explore the Qualitative Bankruptcy data set & Quantitative Bankruptcy data set and explain the details of the dataset like features, instances, data organization, etc. Next, we delve into data preprocessing steps, where we state the problems present with the data like missing data and data imbalance, and explain how we dealt with them. Next, we introduce the classification models we have considered and explain how we train our data using these models. Later, we analyze and evaluate the performance of these models using certain metrics like accuracy, precision and recall.

4.1 Data

In this project, two dataset was used. The Qualitative dataset was used to find out highly correlated factors that impacts on bankruptcy. The Quantitative dataset was used to find out highly correlated ratios with respect to Qualitative dataset factors from bank balance sheet.

Qualitative Bankruptcy Data

The data set have considered for addressing the bankruptcy prediction problem is the Qualitative bankruptcy data, hosted by the University of California Irvine (UCI) Machine Learning Repository - a huge repository of freely accessible datasets for research and learning purposes intended for the Machine Learning/Data Science community.

The bankrupt banks were analyzed via data and questionnaires and it was defined into qualitative from quantitative. The data set is very apt for our research about bankruptcy prediction because it has highly useful econometric indicators as attributes (features).

The data set is summarized in Table 1 below.

Title	Qualitative_Bankruptcy database
Number of Instances	250
Number of Attributes	6
Attributes Information **(P=Positive, A-Average, N-negative,** **B-Bankruptcy, NB-Non-Bankruptcy)**	1. Industrial Risk: {P,A,N} 2. Management Risk: {P,A,N} 3. Financial Flexibility: {P,A,N} 4. Credibility: {P,A,N} 5. Competitiveness: {P,A,N} 6. Operating Risk: {P,A,N} 7. Class: {B,NB}
Feature Characteristics	Categorical Values
Missing Data	No
Associated Tasks	Classification
Class Distribution	[143 instances For Non-Bankruptcy] [107 instances For Bankruptcy]

Table 1: Summary of Qualitative Bankruptcy Data.

Description:

i.**Industry risk (IR) :**

Government policies and International agreements,

Cyclicality,

Degree of competition,

The price and stability of market supply,

The size and growth of market demand,

The sensitivity to changes in macroeconomic factors,

Domestic and international competitive power,

Product Life Cycle.

ii.**Management risk(MR):**

Ability and competence of management,

Stability of management,

The relationship between management/ owner,

Human resources management,

Growth process/business performance,

Short and long term business planning,

achievement and feasibility.

iii.**Financial Flexibility(FF):**

Direct financing,

Indirect financing,

Other financing

iv.**Credibility (CR):**

Credit history,

reliability of information,

The relationship with financial institutes.

v.**Competitiveness (CO):**

Market position,

The level of core capacities,

Differentiated strategy,

vi.**Operating Risk (OP):**

The stability and diversity of procurement,

The stability of transaction,

The efficiency of production,

The prospects for demand for product and service,

Sales diversification,

Sales price and settlement condition,

Collection of A/R,

Effectiveness of sale network.

Quantitative Bankruptcy Data

This data set was generated with the help of Altman Bankruptcy Model and Ratios. The bankruptcy equation of Altman bankruptcy model is given below,

$$\boxed{\mathbf{Z = 0.012\ X1 + 0.014\ X2 + 0.033\ X3 + 0.006\ X4 + 0.999\ X5}}$$

Where,

X1 = Working capital / Total assets,

X2 = Retained earnings / Total assets,

X3 = Earnings before interest and taxes/ Total assets,

X4 = Market value of equity / Book value of total liabilities, X5 = Sales / Total assets,

Z = Altman Bankrupt Value

The performance of any business is assessed based on the value of Z. That is, when the value of Z is less than 2.675, Altman prediction would lead to bankruptcy, otherwise it assures the better performance of business.

The data set is summarized in Table 2 below.

Title	Quantitative_Bankruptcy database
Number of Instances	216216
Number of Attributes	5
Feature Characteristics	Real Values
Missing Data	No
Associated Tasks	Classification
Class Distribution	[188325 instances For Non-Bankruptcy] [27891 instances For Bankruptcy]

Table 2: Summary of Quantitative Bankruptcy Data.

4.2 Dataset Quality Assessment

Now move on to assessing the quality of the data set. As mentioned earlier, the data set suffers from missing values and data imbalance. Since there is no missing data, further proceeds with imbalance.

4.2.1 Data Imbalance

We have not dealt with the class imbalance (if any) in the data. Simply put, Data Imbalance is a condition where the samples belonging to one or more 'majority' class labels of a labeled dataset heavily outnumber the sample belonging to the other 'minority' classes. Data imbalance critically affects the modeling as the models won't have sufficient data belonging to minority classes to train on and this leads to biased models, ultimately leading to poor performance on test data.

In Qualitative data, there is very less imbalance about 7.2% and not to worry about that. But, in Quantitative data, there is very high imbalance about 37.11% . we will proceeds method only for quantitative data. Oversampling is increasing the class distribution of the minority class label whereas Under sampling is decreasing the class distribution of the majority class label. In this project, we explored Synthetic Minority Oversampling Technique or SMOTE.

4.2.2 SMOTE (Synthetic Minority Oversampling Technique)

Synthetic Minority Oversampling Technique (SMOTE) is a widely used oversampling technique. To illustrate how this technique works consider some training data which has s samples, and f features in the feature space of the data. For simplicity, assume the features are continuous. As an example, let us consider a dataset of birds for clarity. The feature space for the minority class for which we want to oversample could be beak length, wingspan, and weight. To oversample, take a sample from the dataset, and consider its k nearest neighbors in the feature space. To create a synthetic data point, take the vector between one of those k neighbors, and the current data point. Multiply this vector by a random number x which lies between 0, and 1. Adding this to the current data point will create the new synthetic data point. SMOTE was implemented from the imbalanced learn library.

4.3 Data Modeling

In this section, we will look at the various classification models that we have considered for training on the both datasets separately to achieve the task of coming up with a predictive model that would predict the bankruptcy status of a given (unseen) banks with an appreciable accuracy.

We have considered the following 7 models:

1. Logistic Regression
2. Support Vector Machine
3. Gaussian Naïve Bayes
4. Decision Tree
5. Random Forests
6. Extreme Gradient Boosting
7. Balanced Bagging

Further ensemble method was used to combine the best predictors of both data set to provide better outcome.

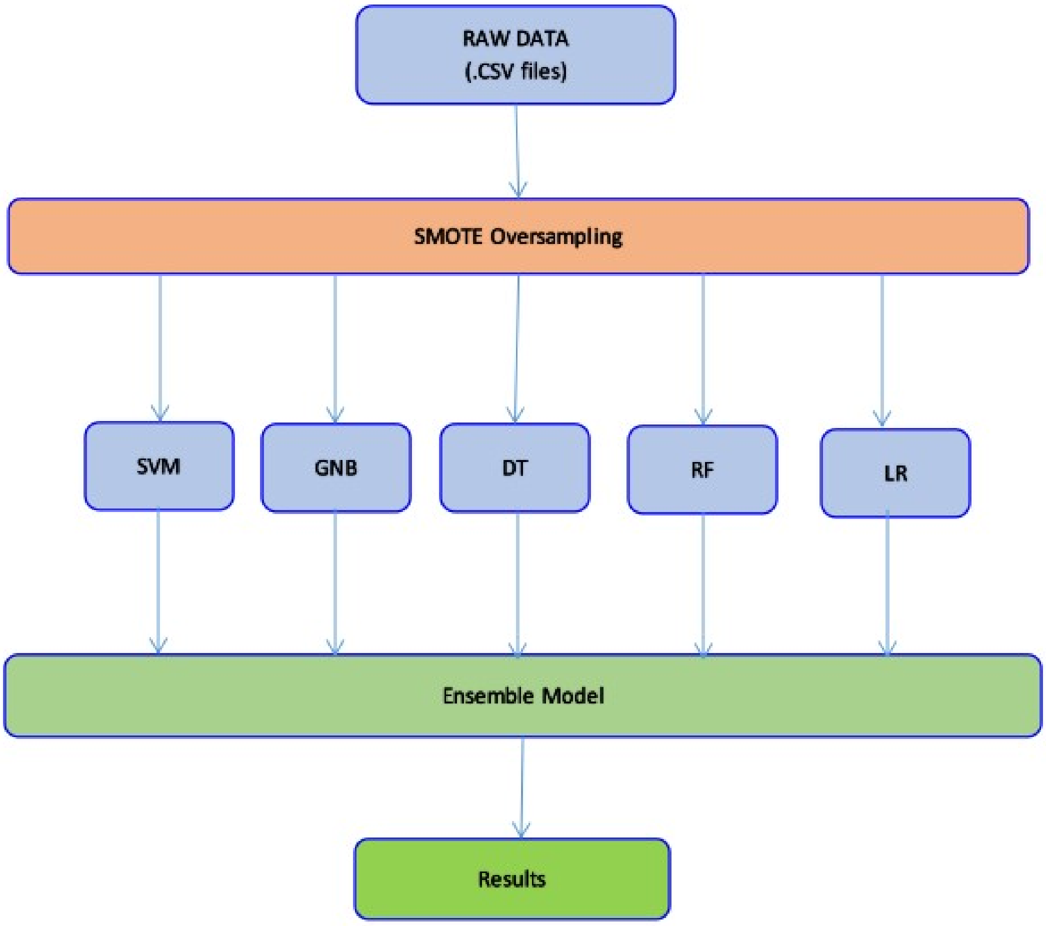


Figure 1: Pipeline for data modeling

Figure 1 shows the pipeline of data modeling for this project. After having obtained the formatted datasets from the raw data (.csv files), Later, we have oversampled the datasets with SMOTE oversampling technique. The datasets are ready for the data modeling step. We model both datasets with the 7 models listed above.

4.3.1 Logistic Regression Classifier

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

$$p = 1/1 + e^{-(\beta 0+\beta 1X1+\beta 2X2.....\beta nXn)}$$

4.3.2 Support Vector Machine

Generally, Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

4.3.3 Gaussian Naïve Bayes Classifier

Naive Bayes classifier is one of the supervised learning algorithms which is based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1 through x_n , Bayes' theorem states the following relationship:

$$P(y \mid x_1, \ \dots, x_n) = \frac{P(y)P(x_1, \ \dots, x_n \mid y)}{P(x_1, \ \dots, x_n)}$$

Using the naive independence assumption that:

$$P(x_i \mid y, x_1, \ \dots, x_{i-1}, x_{i+1}, \ \dots x_n) = P(x_i \mid y)$$

for all i,this relationship is simplified to:

$$P(y \mid x_1, \ \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ \dots, x_n)}$$

Since P is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \ \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Gaussian Naïve Bayes implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

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

The parameters σ_y and μ_y are estimated using maximum likelihood.

4.3.4 Decision Tree Classifier

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking.

4.3.5 Random Forest

It works in four steps:

- Select random samples from a given dataset.
- Construct a decision tree for each sample and get a prediction result from each decision tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction.

4.3.6 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is built on the principles of gradient boosting framework. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. XGBoost uses a more regularized model formalization to control over-fitting, which gives it better performance. In our model, the number of estimators used are 100. The model internally uses log-linear classifier for regularizing the model with $\lambda = 1$.

4.3.7 Balanced Bagging Classifier

Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees used for classification task. We know that decision trees are sensitive to the specific data on which they are trained. If the training data is changed the resulting decision tree can be quite different and in turn the predictions can be quite different. Balancing the data set before training the classifier improve the classification performance. In addition, it avoids the ensemble to focus on the majority class which would be a known drawback of the decision tree classifiers. We have used base estimator as Random Forests in order to perform Balanced Bagging. The number of estimators used in our model are 5. We have considered 'gini' as a measure of the quality of a split.

4.4 Ensemble Model

Ensemble learning techniques attempt to make the performance of the predictive models better by improving their accuracy. Ensemble Learning is a process using which multiple machine learning models (such as classifiers) are strategically constructed to solve a particular problem.

There are different types of Ensemble Learning techniques which differ mainly by the type of models used (homogeneous or heterogeneous models), the data sampling (with or without replacement, k-fold, etc.) and the decision function (voting, average, meta model, etc). Therefore, Ensemble Learning techniques can be classified as:

- Stacking
- Blending
- Voting

Blending

Blending is a technique derived from Stacking Generalization. The only difference is that in Blending, the k-fold cross validation technique is not used to generate the training data of the meta-model. Blending implements "one-holdout set", that is, a small portion of the training data (validation) to make predictions which will be "stacked" to form the training data of the meta-model. Also, predictions are made from the test data to form the meta-model test data.

In figure 2 we can see a Blending architecture using 3 base models (weak learners) and a final classifier. The blue boxes represent that portion of the training data that is used to generate predictions (yellow boxes) to form the meta-model. The green boxes represent the test data which is used to generate predictions to form the meta-model test data (purple boxes).

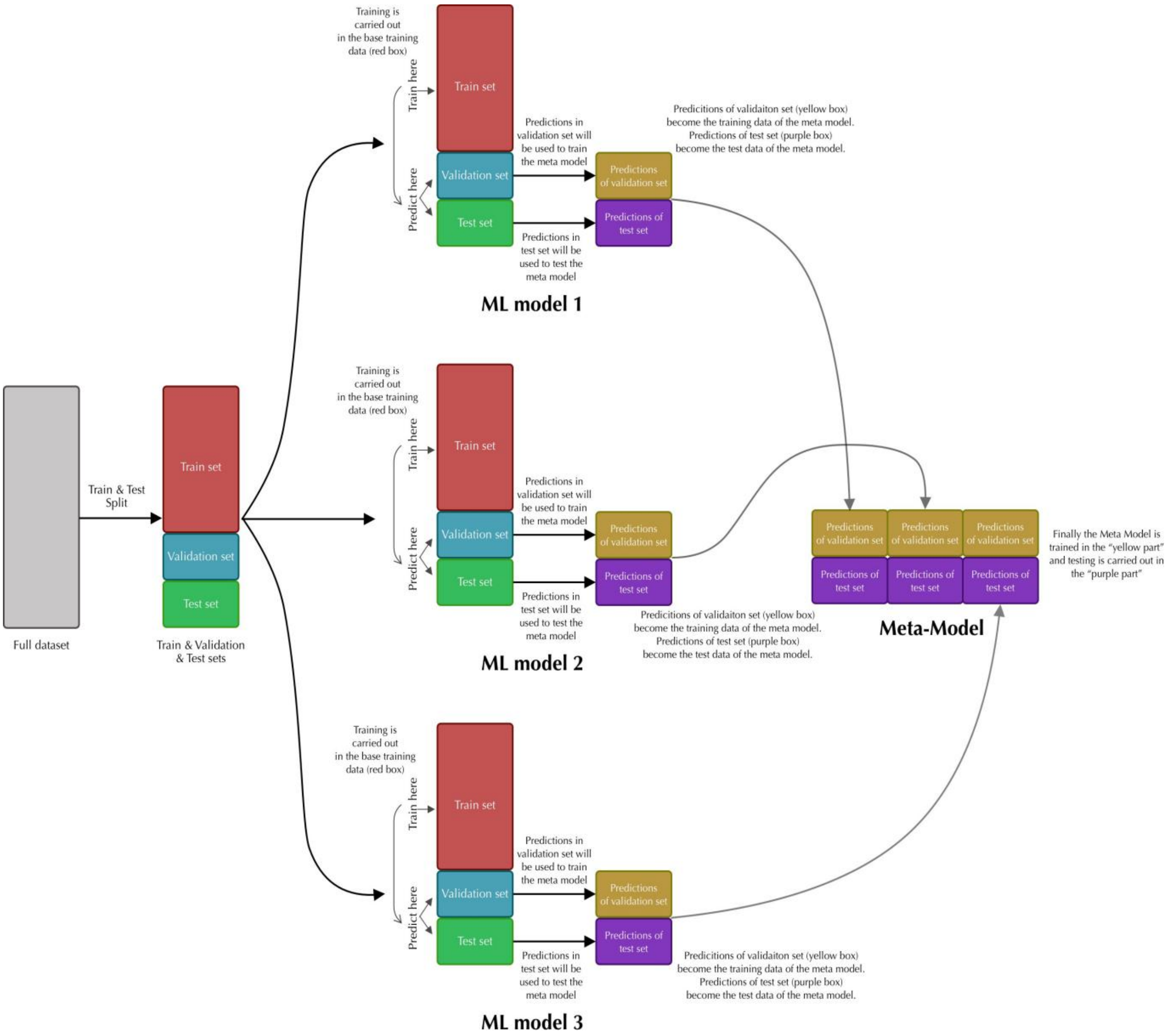

Full dataset
Train & Test Split
Train set
Validation set
Test set
Train & Validation & Test sets
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions in test set will be used to test the meta model
Predictions of validation set
Predictions of test set
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 1
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions in test set will be used to test the meta model
Predictions of validation set
Predictions of test set
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 2
Predictions of validation set
Predictions of validation set
Predictions of validation set
Predictions of test set
Predictions of test set
Predictions of test set
Finally the Meta Model is trained in the "yellow part" and testing is carried out in the "purple part"
Meta-Model
Training is carried out in the base training data (red box)
Train here
Predict here
Train set
Validation set
Test set
Predictions in validation set will be used to train the meta model
Predictions in test set will be used to test the meta model
Predictions of validation set
Predictions of test set
Predicitions of validaiton set (yellow box) become the training data of the meta model. Predictions of test set (purple box) become the test data of the meta model.
ML model 3

Figure 2: Blending Architecture

(Source: https://towardsdatascience.com)

CHAPTER 5

DESIGN

In this stage, the problem is formulated, and then a model is built based upon real–world objects. The analysis produces models on how the desired system should function and how it must be developed. The models do not include any implementation details so that it can be understood and examined by any non–technical application expert.

5.1 USE CASES

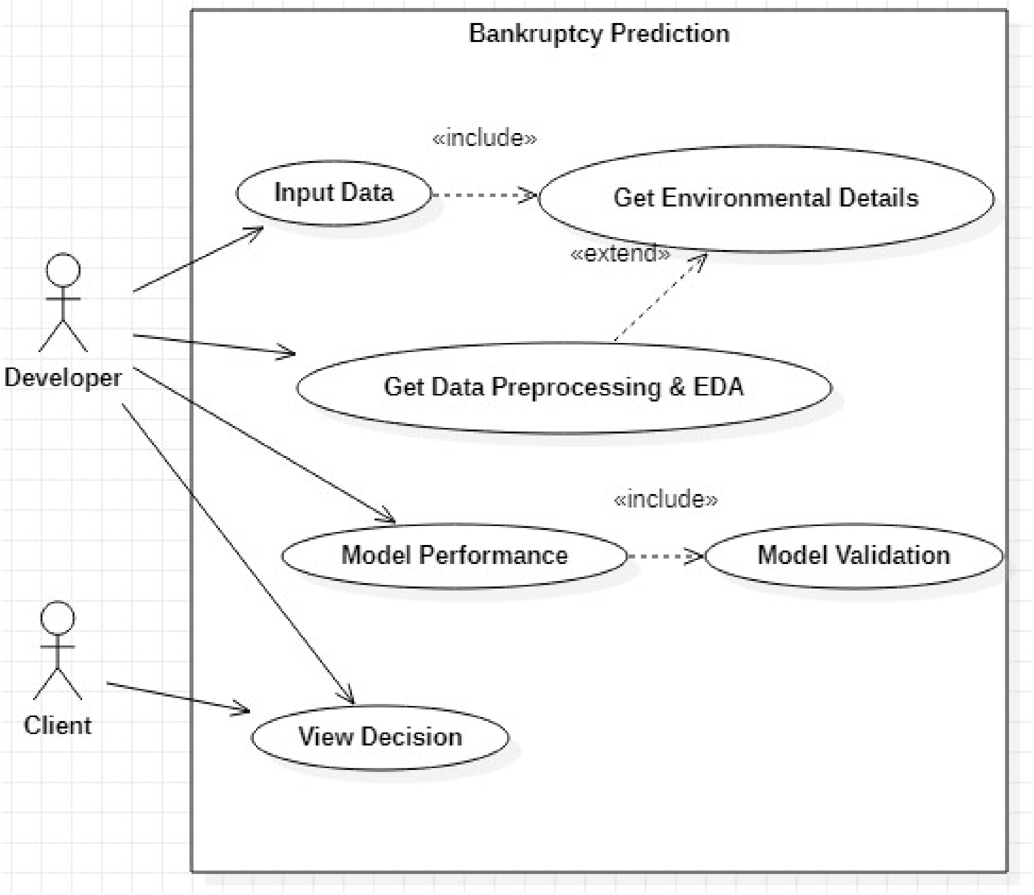


Figure 3: Use Case Diagram

Description:

Developer

- The Developer can access the data from client.
- The Developer can do feature engineering and evaluate model performance.

Client

The Client can view the decision for the problem.

5.2 CLASS DIAGRAM

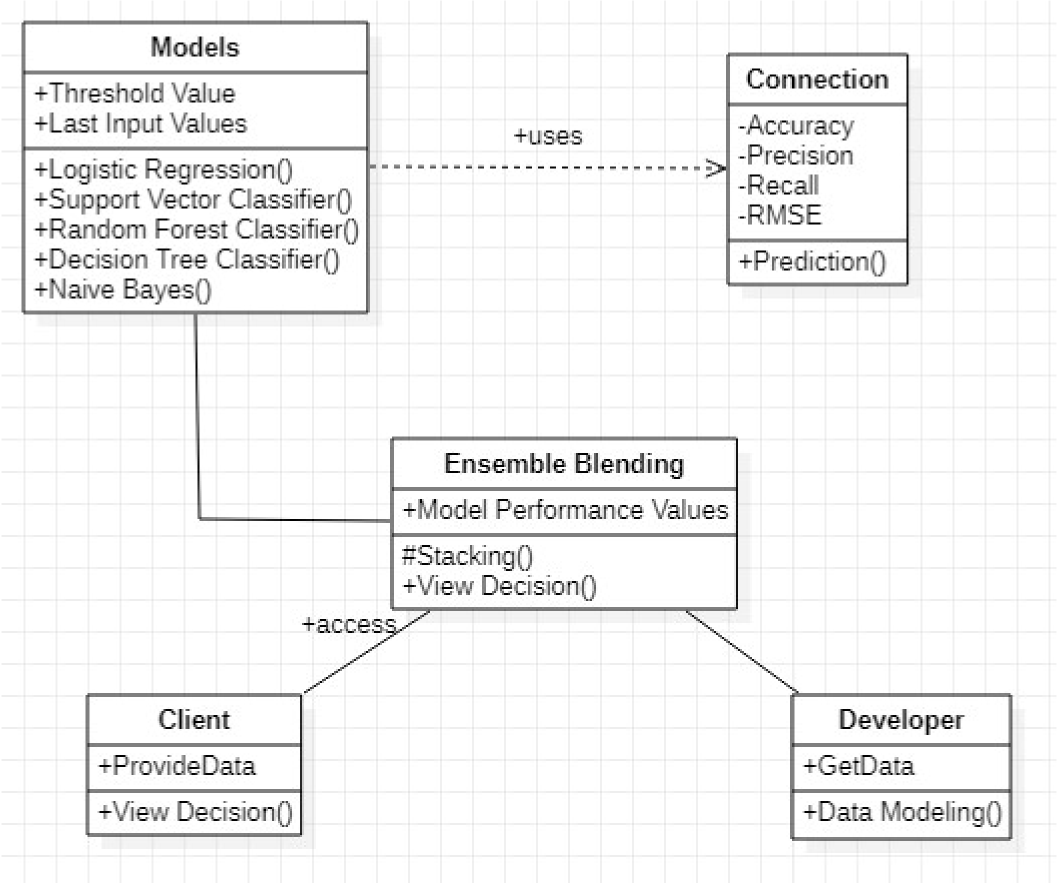


Figure 4: Class Diagram

Description:

- Different models are performed and the results are stored in stack.
- The results validated using ROC, Accuracy, Precision, Recall.
- The models performance was used in blending to provide decision.
- The client can access the decision.

5.3 ACTIVITY DIAGRAM

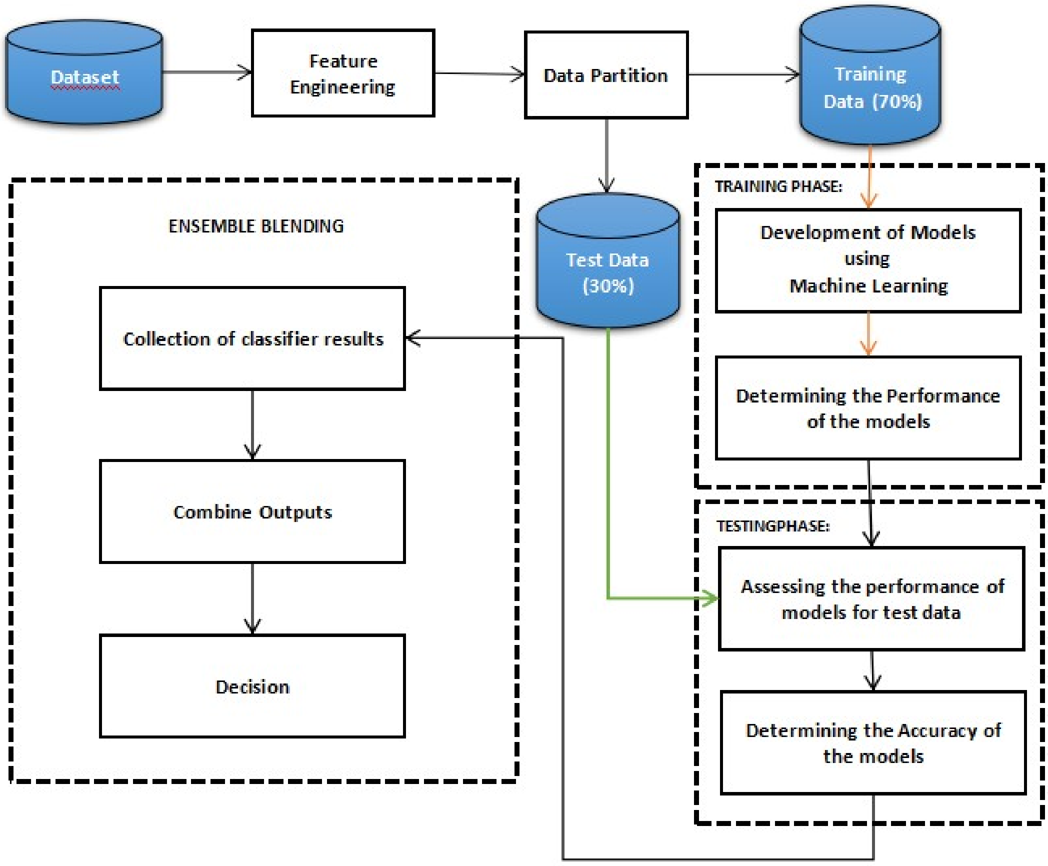

Dataset
Feature Engineering
Data Partition
Training Data (70%)
Test Data (30%)
ENSEMBLE BLENDING
Collection of classifier results
Combine Outputs
Decision
TRAINING PHASE:
Development of Models using Machine Learning
Determining the Performance of the models
TESTINGPHASE:
Assessing the performance of models for test data
Determining the Accuracy of the models

Figure 5: Activity Diagram

Description:

- The dataset was loaded, preprocessed, and partitioned into test and training data.
- Then training data was modeled using machine learning and validated using test dataset.
- The performance was evaluated using test data.
- The results was feeded into Ensemble Blending and decision was made.

CHAPTER 6
IMPLEMENTATION

6.1 Experimentation

The programming environment used for the project is Python v3.6. We used an Intel Core i3 Core processor with 4 GB Memory (RAM) and 1 TB of storage (disk space) to run our experiments. Our code workflow exactly mimics the data modeling pipeline shown in Figure 6.

We used the libraries listed in Table 3 to run our experiments and achieve our results.

Library	**Description**
numpy	Data organization and statistical operations.
pandas	Data manipulation and analysis. Storing and manipulating numerical tables.
matplotlib, seaborn	Plotting library
imblearn.over_sampling.SMOTE	Perform SMOTE Oversampling
xgboost.XGBClassifier	Extreme Gradient Boosting classifier
sklearn.ensemble.RandomForestClassifier	Random Forest Classifier
sklearn.linear_model.LogisticRegression	Logistic Regression Classifier
imblearn.ensemble.BalancedBaggingClassifier	Balanced Bagging Classifier
sklearn.tree.DecisionTreeClassifier	Decision Tree Classifier
sklearn.naive_bayes.GaussianNB	Gaussian Naïve Bayes Classifier
sklearn.metrics	Performance evaluation metrics like accuracy score, recall, precision, ROC curve, etc.

Table 3: Library used in this project

1. Firstly, we imported all the libraries we listed in Table 3.
2. Then we load the raw data (.csv files) as pandas dataframes. Although the features are numeric and class labels are binary, in the dataframes, all the values were stored as objects. So we converted them to float and int values respectively.

3. Now we start the data analysis. we apply SMOTE oversampling on quantitative dataframes to get fresh dataframes of oversampled dataframes and store them in a dictionary.

4. We create (instantiate) the 7 classifier models (GNB, LR,SVM DT, RF, XGB, BB) and store them in a dictionary.

5. We iterate best classifier in ensemble models.

6. Validation using Metrics.

6.2 CODE

```
#Models
    logit = LogisticRegression(random_state= 42)
    svc = SVC(random_state= 42)
    gnb = GaussianNB()
    dtc = DecisionTreeClassifier(criterion="gini",splitter="best" ,max_depth=5)
    rf = RandomForestClassifier(random_state= 42)

# Splitting Train and Test Data
  y = df['Y']
  x = df.drop(['Y'], axis = 1)
  x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.3, random_state = 42)

#Metrics
  def result_model(model,x_test,y_test):

# Use the model on the testing data to predict the results
  model.fit(x_train, y_train)
  y_pred = model.predict(x_test)
  print("Validation Result :")

# Print the Accuracy Score
  print("Accuracy Score : ",accuracy_score(y_test,y_pred))
  print("Classification Report :")
  print(classification_report(y_test,y_pred))
# Print the R2 score
  print ("R2 score:\n")
  print (('{:.2f}'.format((100*(r2_score(y_test, y_pred))))) + " %")
```

```
    print ("\n")

# Print the mean squared error
    print ("Mean-squared error:\n")
    print(mean_squared_error(y_test, y_pred))

#ROC
    test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_pred)
    plt.grid()
    plt.plot(test_fpr, test_tpr, label=" AUC ="+str(auc(test_fpr, test_tpr)))
    plt.plot([0,1],[0,1],'g--')
    plt.legend()
    plt.xlabel("True Positive Rate")
    plt.ylabel("False Positive Rate")
    plt.title("AUC(ROC curve)")
    plt.grid(color='black', linestyle='-', linewidth=0.5)
    plt.show()
    print("Confusion Matrix:")
    sns.heatmap(confusion_matrix(y_test,y_pred), annot=True)

result_model(model,x_test,y_test)

#Blending
from numpy import hstack
from sklearn.datasets import make_classification

def get_models():
    models = list()
    models.append(('lr', LogisticRegression()))
    models.append(('svm',SVC()))
    models.append(('gnb',GaussianNB()))
    models.append(('knn', KNeighborsClassifier()))
    models.append(('cart', DecisionTreeClassifier()))
    models.append(('rf',RandomForestClassifier(n_estimators=100)))
    return models

# fit the blending ensemble
      def fit_ensemble(models, X_train, X_val, y_train, y_val):
            # fit all models on the training set and predict on hold out set
            meta_X = list()
            for name, model in models:
            # fit in training set
                        model.fit(X_train, y_train)
            # predict on hold out set
                        yhat = model.predict(X_val)
            # reshape predictions into a matrix with one column
                        yhat = yhat.reshape(len(yhat), 1)
            # store predictions as input for blending
                        meta_X.append(yhat)
            # create 2d array from predictions, each set is an input feature
                   meta_X = hstack(meta_X)
            # define blending model
                  blender = LogisticRegression()
            # fit on predictions from base models
                 blender.fit(meta_X, y_val)
                return blender

# make a prediction with the blending ensemble
    def predict_ensemble(models, blender, X_test):
    # make predictions with base models
             meta_X = list()
            for name, model in models:
            # predict with base model
                        yhat = model.predict(X_test)
            # reshape predictions into a matrix with one column
                        yhat = yhat.reshape(len(yhat), 1)
            # store prediction
                        meta_X.append(yhat)
            # create 2d array from predictions, each set is an input feature
             meta_X = hstack(meta_X)
            # predict
            return blender.predict(meta_X)

# create the base models
      models = get_models()
# train the blending ensemble
```

```
    blender = fit_ensemble(models, x_train, x_test, y_train, y_test)
# make predictions on test set
    yhat = predict_ensemble(models, blender, x_test)
# evaluate predictions
    score = accuracy_score(y_test, yhat)
    print('Blending Accuracy: %.3f' % (score*100))

#ROC
from sklearn import metrics
fpr, tpr, _ = metrics.roc_curve(y_test,  yhat)
auc = metrics.roc_auc_score(y_test, yhat)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```

For detailed implementation code, use the following resources.

GITHUB: https://github.com/rjsekar-rs/Bankruptcy-Predicition-Using-Ensemble-Model

CHAPTER 7

RESULTS

7.1 Model Performances

Accuracy and ROC of seven models for Qualitative datasets was given below:

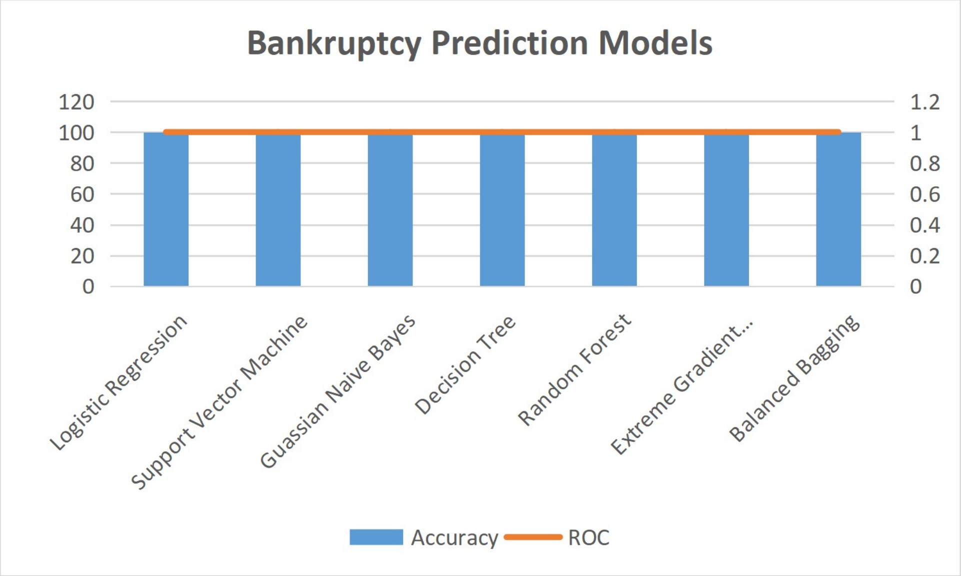


Figure 6: Qualitative Accuracy Chart

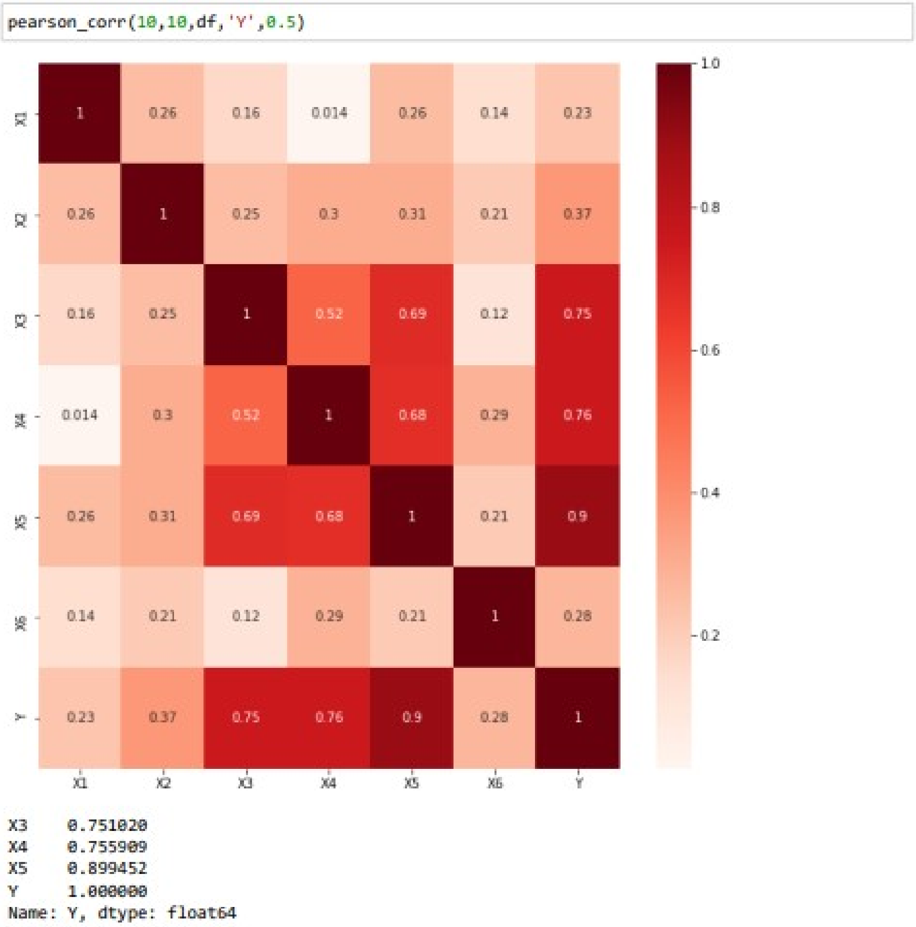
Qualitative data every mode depicts 100% accuracy. It is pure data for Bankruptcy prediction.

Features:

From Qualitative data set, we can infer from tree chart and correlation matrix that Financial Flexibility, Credibility, Competitiveness has high correlation. These factors has high impact on bankruptcy prediction. And these factors are relatable to Altman ratios. ie.,

- Management Risk is a core feature of Working capital / Total assets(X1),
- Financial Flexibility is a core feature of Retained earnings / Total assets (X2) and Earnings before interest and taxes/ Total assets (X3),
- Operating Risk is a core feature of Market value of equity / Book value of total liabilities(X4) and Sales / Total assets(X5)

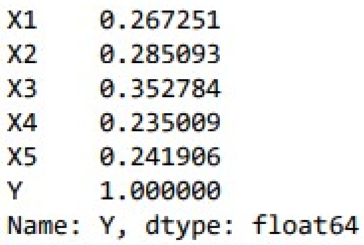
From the relation, we can infer that X2, X3 has high impact on bankruptcy prediction.



```
pearson_corr(10,10,df,'Y',0.5)
```

```
X3     0.751020
X4     0.755909
X5     0.899452
Y      1.000000
Name: Y, dtype: float64
```

Figure 7: Qualitative Features Correlation Score



```
X1     0.267251
X2     0.285093
X3     0.352784
X4     0.235009
X5     0.241906
Y      1.000000
Name: Y, dtype: float64
```

Figure 8: Quantitative Features Correlation Score

From this we can see that X2,X3 ratios has high correlation than other ratios.

Accuracy and ROC of seven models for Quantitative datasets was given below:

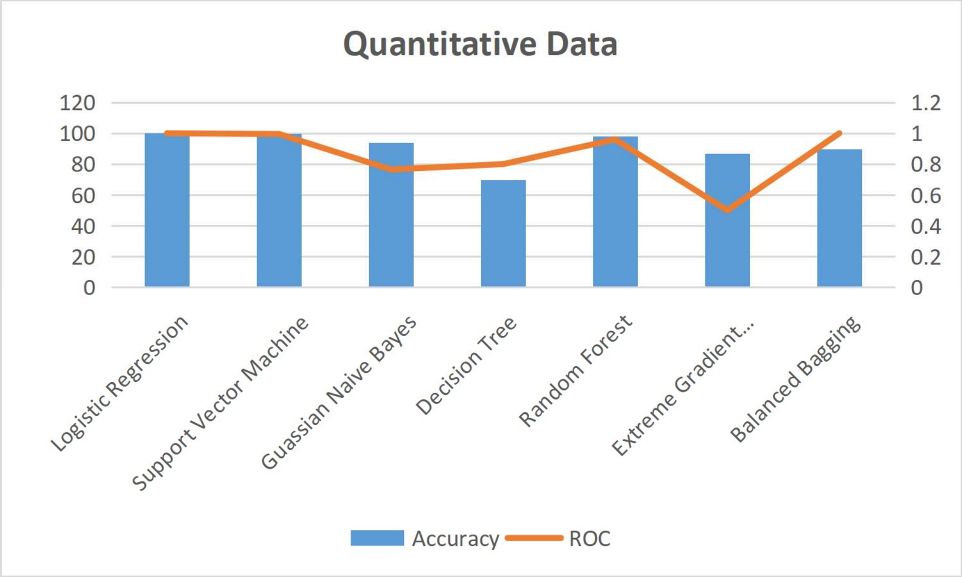
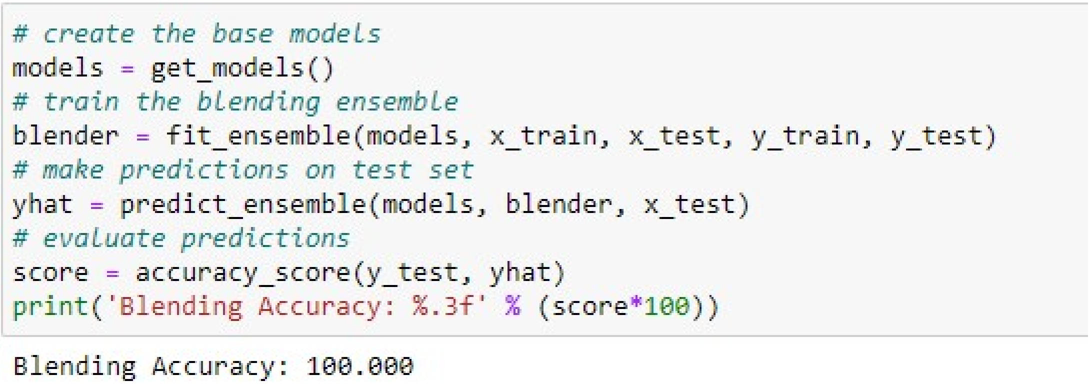


Figure 9: Quantitative Accuracy Chart

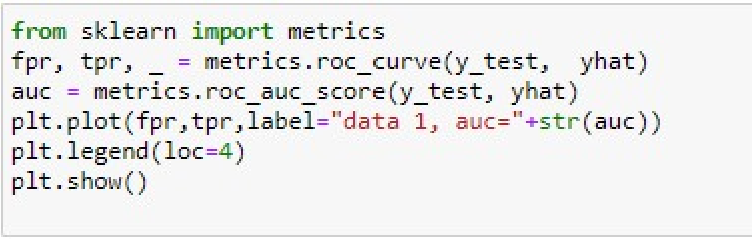
From the above chart, we can infer that Logistic Regression, Support Vector Machine, Random Forest are the best classifiers. We can infer there is drastic change in accuracy of every model due to involvement of low correlation ratios.

7.2 ENSEMBLE BLENDING RESULTS:



```
# create the base models
models = get_models()
# train the blending ensemble
blender = fit_ensemble(models, x_train, x_test, y_train, y_test)
# make predictions on test set
yhat = predict_ensemble(models, blender, x_test)
# evaluate predictions
score = accuracy_score(y_test, yhat)
print('Blending Accuracy: %.3f' % (score*100))
```

```
Blending Accuracy: 100.000
```



```
from sklearn import metrics
fpr, tpr, _ = metrics.roc_curve(y_test,  yhat)
auc = metrics.roc_auc_score(y_test, yhat)
plt.plot(fpr,tpr,label="data 1, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```

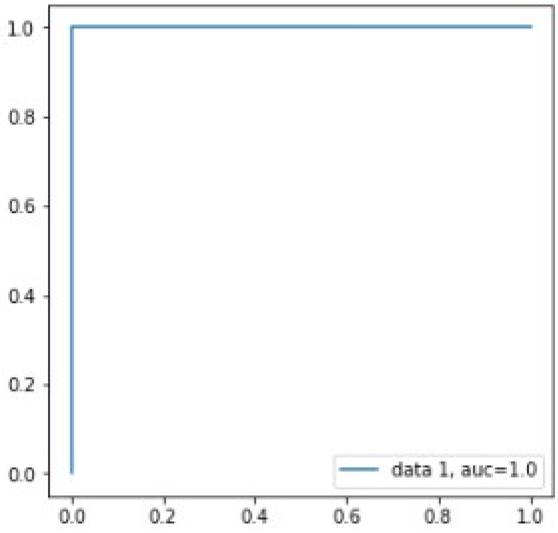


Figure 10: Blending Result

Both dataset produced 100% accuracy in Ensemble blending. It is a best model for bankruptcy prediction.