“Bankruptcy Prediction Using Different Classification Models”

ABSTRACT

The "Bankruptcy Prediction Using Different Classification Models" project aims to develop an effective predictive model to assess the likelihood of a company facing bankruptcy based on historical financial data. Bankruptcy prediction is a critical task for financial institutions, investors, and regulatory bodies to make informed decisions and mitigate risks.

The project utilizes a diverse set of classification models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and, optionally, a Neural Network. The dataset comprises historical financial metrics, and the models are trained to distinguish between financially stable and distressed companies.

The documentation provides a comprehensive overview of the project, including data preprocessing steps, exploratory data analysis, feature engineering techniques, and the selection and training of various classification models. The comparative analysis of model performances, evaluation metrics, and insights derived from feature importance are presented.

Additionally, the documentation discusses potential deployment strategies, such as model deployment considerations and integration into a web application or API. The project's findings, limitations, and recommendations for future work are summarized, providing a holistic view of the bankruptcy prediction system.

This documentation serves as a valuable resource for data scientists, financial analysts, and researchers interested in applying machine learning techniques to predict bankruptcy, offering insights into model selection, feature engineering, and the interpretability of predictive models in a financial context. Our bankruptcy prediction model has high predictive accuracy with clear explanations and is therefore directly applicable to the industry.

Keywords: Classification, decision tree, bankruptcy prediction, random forest classifiers.

Chapter -1

INTRODUCTION

1.1 Introduction

Bankruptcy prediction is the problem of detecting financial distress in businesses which will lead to eventual bankruptcy. Bankruptcy prediction has been studied since at least 1930s. The early models of bankruptcy prediction employed univariate statistical models over financial ratios. The univariate models were followed by multi-variate statistical models such as the famous Altman Z-score model. The recent advances in the field of Machine learning have led to the adoption of Machine learning algorithms for bankruptcy prediction. Machine Learning methods are increasingly being used for bankruptcy prediction using financial ratios. A study by Barboza, Kimura and Altman found that Machine Learning models can outperform classical statistical models like multiple discriminant analysis (MDA) by a significant margin in bankruptcy prediction (Barboza et al., 2017).

Bankruptcy prediction is an important for modern economies because early warnings of bankrupt help not only the investor but also public policy makers to take proactive steps to minimize the impact of bankruptcies.The distress caused by financial failure disturbs the firm's liabilities that are disproportionate to its assets. This loss could degrade the capabilities of stockholders, employees, customers, and investors. Hence, the prediction of bankruptcy plays a significant role in economics and finance research fields. The prediction process is to make decision systems related to business operations in a reliable manner. Most of the business entities wish to have relationships in the long term. Hence, it is valuable to foresee the possibility of the bankruptcy of a business customer (or) partner. To enhance the accuracy of bankruptcy prediction, researchers concentrated on two paths: first, exploring the significant variables and enhancing the methodologies of the bankruptcy prediction and its computation infrastructure. Anticipating such scenarios is crucial for making informed decisions, managing risks, and safeguarding the stability of the financial ecosystem.

In short, bankruptcy prediction is a very important task for many related financial institutions. In general, the aim is to predict the likelihood that a firm may go bankrupt. Financial institutions are in need of effective prediction models in order to make appropriate lending decisions.

1.2 Motivation

In an era where financial landscapes are continually shaped by global economic dynamics, technological advancements, and unforeseen challenges, the motivation behind undertaking the project, "Bankruptcy Prediction Using Different Classification Models," is deeply rooted in addressing critical issues faced by businesses, investors, and regulatory bodies alike.

Financial stability is the bedrock of a thriving economy, and the repercussions of financial distress extend far beyond individual businesses. As economic ecosystems become increasingly interconnected, the ability to navigate and understand the complex web of financial variables is crucial. Traditional financial analysis methods, while insightful, often struggle to contend with the volume and intricacy of contemporary financial data.

Despite the considerable amount of research on bankruptcy prediction as a critical topic in risk management, little attention has been given to feature selection, which many consider to be one of the most important steps in the data mining process. Conducting more research on feature selection could enhance our understanding of a company’s financial well-being, benefiting not only the company but also shareholders, potential investors, and financial institutions

The motivation behind this project stems from the recognition that leveraging the power of machine learning can revolutionize the way we approach financial analysis. By harnessing advanced classification models, we can unveil patterns, discern subtle indicators, and, most importantly, predict potential financial crises before they escalate.

1.3 Problem formulation

This project aims to find which classification model selects the most relevant features for bankruptcy prediction. This will be tested with a machine learning model as the classifier and a bankruptcy dataset of companies in the US stock market. The problem can be framed as binary classification, with each company belonging to one of these two classes at a specific year: bankruptcy or healthy. With multiple features and the data formatted as a time series for each company the problem to be solved is binary classification on multivariate time series data. Regarding delimitations, this thesis will focus on one machine learning model and one dataset, and a total of three classification model will be evaluated.

The intention of the study is to illustrate and investigate how machine learning can be exploited in the field of economics. More specifically, the aim is to study how machine Introduction learning methods can be used to predict corporate bankruptcies.

The central problem addressed by this project is the prediction of bankruptcy in companies based on historical financial data. The challenge is to design and implement a predictive model that can effectively discern patterns indicative of financial distress, providing timely insights to stakeholders.

1.4 Aims and Objective

The aims and objectives of the project, "Bankruptcy Prediction Using Different Classification Models," are outlined to provide a clear roadmap and purpose for the undertaking. This section delineates the overarching goals and specific objectives that the project aims to achieve.

Aims:-

1. Development of Robust Predictive Models:

The primary aim of this project is to develop robust predictive models capable of assessing the likelihood of a company facing bankruptcy. Leveraging machine learning techniques, the models will be designed to analyze historical financial data and provide accurate predictions.

2. Model Comparison and Evaluation:

The project aims to systematically compare and evaluate various classification models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and, optionally, a Neural Network. This comparative analysis seeks to identify the most effective model for bankruptcy prediction.

3. Feature Engineering and Selection:

An aim of the project is to explore feature engineering techniques to enhance the predictive power of the models. This involves transforming variables, creating new features, and strategically selecting relevant financial indicators for inclusion in the predictive models.

4. Interpretability and Transparency:

The project places a strong emphasis on ensuring the interpretability and transparency of the predictive models. Stakeholders should be able to understand the rationale behind the models' predictions, fostering trust and confidence in the insights provided.

Objectives:-

1. Data Preprocessing:

Develop a robust preprocessing pipeline to handle complexities in financial datasets, including handling missing values, outlier detection, and feature scaling.

2. Identification of Key Financial Indicators:

Identify key financial indicators and relationships to create a meaningful feature set for predictive modeling.

3. Model Implementation:

Implement and train various classification models, including Decision Tree, Random Forest, Support Vector Machine (SVM).

4. Performance Evaluation:

Evaluate and compare the performance of each model using appropriate metrics, including accuracy, precision, recall, F1-score, and area under the Receiver Operating Characteristic (ROC) curve.

5. Feature Importance Analysis:

Conduct feature importance analysis to provide insights into the key financial indicators influencing predictions.

1.5 Scope and Limitation

The project's scope is focused on the predictive modeling of bankruptcy using different classification algorithms. However, it's important to acknowledge certain limitations: -

- The predictive nature of the models does not imply causation. The goal is to identify associations and patterns indicative of financial distress.
- The effectiveness of the models is contingent on the quality and representativeness of the historical financial data.
- External economic factors, unforeseen events, or changes in financial regulations may influence the accuracy of predictions.
- The project involves a comprehensive comparison of different classification models, including Decision Tree, Random Forest, Support Vector Machine (SVM). The comparative analysis will assess their performance in terms of accuracy, precision, recall, F1-score, and the area under the ROC curve.

- The availability of sufficient data for training and testing is essential. However, limitations in data volume may affect the robustness of the models, particularly for complex algorithms.

1.6 Related Work

This part describes the prior techniques related to the topic. The effects of bankruptcy were described by several researchers in financial sectors. Most of the studies were carried out by supervised approaches. The data is collected, as per the supervised approach, to design predictive models.

The first documented attempts of bankruptcy predictions were carried out by Patrick (1932). At that time no statistical model was used, the predictions were based on Patrick's own interpretations of the financial ratios and the trends he could discern. First in the 1960s statistical models and hypothesis testing were used for bankruptcy prediction. The work was initiated by Beaver (1966) and two years later Altman (1968) proposed the use of multiple discriminant analysis for bankruptcy prediction. This work was trend-setting and in the years that followed Altman's ideas inspired many others.

Another turning point in the field was the initiation of the generalized linear models (Ohlson, 1980). Generalized linear models have some advantages, firstly, they allow for analysis of the certainty of the predictions and, secondly, it is possible to analyze the effect of each predictor individually.

In recent years machine learning methods have at several times been successfully used to predict corporate bankruptcies. Neural networks trained with back propagation are the most common method for this type of problem (Tsai and Wu, 2008). In a study of small and medium-sized Belgian companies it was shown that relatively good results could be achieved by using only a small number of easily accessible financial ratios as inputs to an artificial neural network (Bredart, 2014).

The study of the Belgian companies was not the first of its kind. It was inspired by, among others, Shah and Murtaza (2000) who used a neural network to predict bankruptcies among US companies between 1992 and 1994 and Becerra et al. (2005) that studied British corporate bankruptcies between 1997 and 2000 using a similar method.

Lately, some attention has also been devoted to ensemble classifiers. It has by Alfaro et al. (2008) and Zi˛eba et al. (2016) been shown that ensemble classifiers can successfully be applied to bankruptcy prediction and significantly outperform other methods.

1.7 System Specifications

This section describes the hardware components and software requirements needed foreffective and efficient running of the system.

Table: 1 Hardware Requirements

In this table we explained the hardware requirement of the project.

SL	Hardware	Processor
01	Processor	2.4 GHz Processor speed
02	Memory	4 GB RAM
03	Disk Space	500 GB

Table: 2 Software Requirements

In this table we explained the hardware requirement of the project

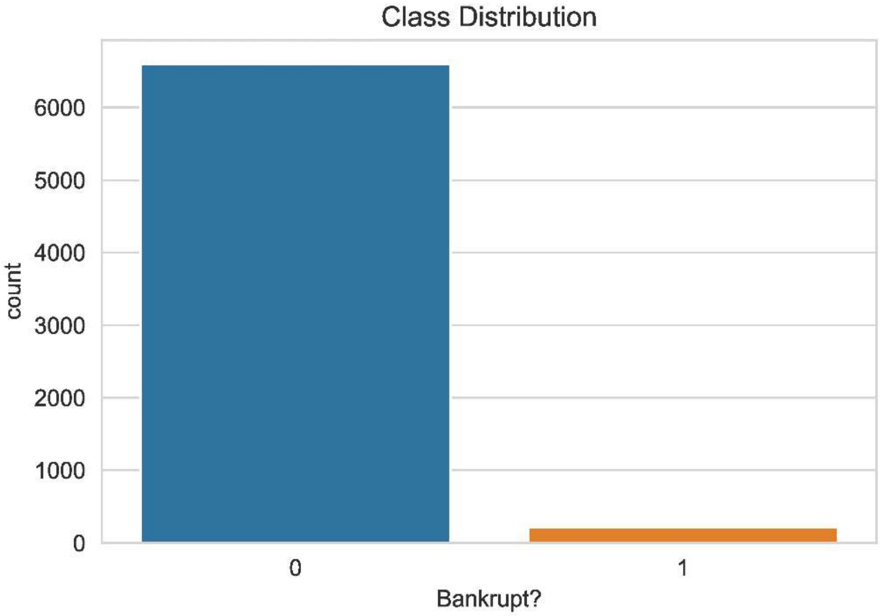
SL	**Software**	**Minimum System Requirement**
1	Operating System	Windows 7,10,11 or MAC Ox 10.8,10.9, or 10.11,LINUX
2	Runtime Environment	Jupyter Notebook, Google Colab
3	Technical Skills	Python , Sklearn, pandas, matplotlib, seaborn

Chapter -2

MATERIALS AND METHODS

2.1 Dataset:

In this project, we have used the data collected from the Taiwan Economic Journal for the years 1999 to 2009. Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. Taiwan stock exchange defines the mentioned corporate bankruptcy based on their business rules. The dataset contains 6819 enterprises, 96 features, two categories. The dataset contains a huge difference between number of bankrupt cases and non-bankrupt cases which results in class imbalance problem, that is likely to degrade the final prediction performance. Fig 1 shows the huge imbalance with 96.774% non-bankruptcy enterprises and 3.226% bankruptcy enterprises. Bankrupt and non-bankrupt firms are marked as ‘1’ and ‘0’ respectively.

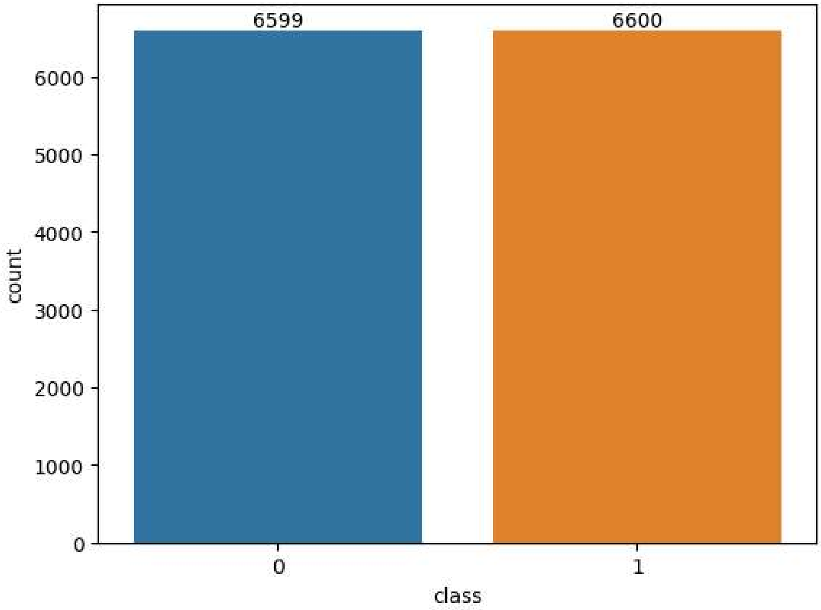


(Fig 2.1: Imbalance between bankrupt and non-bankrupt firms)

Therefore, stratified method is used to collect the same number of bankrupt and non-bankrupt sampling. The dataset that is been collected contains 6599 non-bankrupt cases and 6600 bankrupt cases with each company represented by 95 FRs and 95 CGIs as the input variables. Each variable is normalized into range from 0 and 1 by

$$\forall x \in F, \text{normalize}(x) = \frac{x - \min(F)}{\max(F) - \min(F)}$$

where F is a specific feature set(variable), x is the feature value, and max(F) and min(F) represents the maximum and minimum values of specific feature set. Fig 2 represents the number of bankrupt and non-bankrupt firms.



(Fig 2.2: Representation of bankrupt and non-bankrupt firms after normalization)

To avoid overfitting, cross validation method is used to divide the dataset into training and testing subsets with which we have trained and tested the prediction model. 80% of the data from dataset i.e., 10559 is selected for training data and rest data i.e., 2640 is selected for testing data. Training data input and training data output is represented by x_train and y_train while testing data input and testing data output is represented by x_test and y_test respectively.

Features of the dataset used is defined as following:

Y - Bankrupt?: Class label
X1 - ROA(C) before interest and depreciation before interest: Return On Total Assets(C)

X2 - ROA(A) before interest and % after tax: Return On Total Assets(A)

X3 - ROA(B) before interest and depreciation after tax: Return On Total Assets(B)

X4 - Operating Gross Margin: Gross Profit/Net Sales

X5 - Realized Sales Gross Margin: Realized Gross Profit/Net Sales

X6 - Operating Profit Rate: Operating Income/Net Sales

X7 - Pre-tax net Interest Rate: Pre-Tax Income/Net Sales

X8 - After-tax net Interest Rate: Net Income/Net Sales

X9 - Non-industry income and expenditure/revenue: Net Non-operating Income Ratio

X10 - Continuous interest rate (after tax): Net Income-Exclude Disposal Gain or Loss/Net Sales

X11 - Operating Expense Rate: Operating Expenses/Net Sales

X12 - Research and development expense rate: (Research and Development Expenses)/Net Sales

X13 - Cash flow rate: Cash Flow from Operating/Current Liabilities

X14 - Interest-bearing debt interest rate: Interest-bearing Debt/Equity

X15 - Tax rate (A): Effective Tax Rate

X16 - Net Value Per Share (B): Book Value Per Share(B)

X17 - Net Value Per Share (A): Book Value Per Share(A)

X18 - Net Value Per Share (C): Book Value Per Share(C)

X19 - Persistent EPS in the Last Four Seasons: EPS-Net Income

X20 - Cash Flow Per Share

X21 - Revenue Per Share (Yuan ¥): Sales Per Share

X22 - Operating Profit Per Share (Yuan ¥): Operating Income Per Share

X23 - Per Share Net profit before tax (Yuan ¥): Pretax Income Per Share

X24 - Realized Sales Gross Profit Growth Rate

X25 - Operating Profit Growth Rate: Operating Income Growth

X26 - After-tax Net Profit Growth Rate: Net Income Growth

X27 - Regular Net Profit Growth Rate: Continuing Operating Income after Tax Growth

X28 - Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth

X29 - Total Asset Growth Rate: Total Asset Growth

X30 - Net Value Growth Rate: Total Equity Growth

X31 - Total Asset Return Growth Rate Ratio: Return on Total Asset Growth

X32 - Cash Reinvestment %: Cash Reinvestment Ratio

X33 - Current Ratio

X34 - Quick Ratio: Acid Test

X35 - Interest Expense Ratio: Interest Expenses/Total Revenue

X36 - Total debt/Total net worth: Total Liability/Equity Ratio

X37 - Debt ratio %: Liability/Total Assets

X38 - Net worth/Assets: Equity/Total Assets

X39 - Long-term fund suitability ratio (A): (Long-term Liability+Equity)/Fixed Assets

X40 - Borrowing dependency: Cost of Interest-bearing Debt

X41 - Contingent liabilities/Net worth: Contingent Liability/Equity

X42 - Operating profit/Paid-in capital: Operating Income/Capital

X43 - Net profit before tax/Paid-in capital: Pretax Income/Capital

X44 - Inventory and accounts receivable/Net value: (Inventory+Accounts Receivables)/Equity

X45 - Total Asset Turnover

X46 - Accounts Receivable Turnover

X47 - Average Collection Days: Days Receivable Outstanding

X48 - Inventory Turnover Rate (times)

X49 - Fixed Assets Turnover Frequency

X50 - Net Worth Turnover Rate (times): Equity Turnover

X51 - Revenue per person: Sales Per Employee

X52 - Operating profit per person: Operation Income Per Employee

X53 - Allocation rate per person: Fixed Assets Per Employee

X54 - Working Capital to Total Assets

X55 - Quick Assets/Total Assets

X56 - Current Assets/Total Assets

X57 - Cash/Total Assets

X58 - Quick Assets/Current Liability

X59 - Cash/Current Liability

X60 - Current Liability to Assets

X61 - Operating Funds to Liability

X62 - Inventory/Working Capital

X63 - Inventory/Current Liability

X64 - Current Liabilities/Liability

X65 - Working Capital/Equity

X66 - Current Liabilities/Equity

X67 - Long-term Liability to Current Assets

X68 - Retained Earnings to Total Assets

X69 - Total income/Total expense

X70 - Total expense/Assets

X71 - Current Asset Turnover Rate: Current Assets to Sales

X72 - Quick Asset Turnover Rate: Quick Assets to Sales

X73 - Working capital Turnover Rate: Working Capital to Sales

X74 - Cash Turnover Rate: Cash to Sales

X75 - Cash Flow to Sales

X76 - Fixed Assets to Assets

X77 - Current Liability to Liability

X78 - Current Liability to Equity

X79 - Equity to Long-term Liability

X80 - Cash Flow to Total Assets

X81 - Cash Flow to Liability

X82 - CFO to Assets

X83 - Cash Flow to Equity

X84 - Current Liability to Current Assets

X85 - Liability-Assets Flag: 1 if Total Liability exceeds Total Assets, 0 otherwise

X86 - Net Income to Total Assets

X87 - Total assets to GNP price

X88 - No-credit Interval

X89 - Gross Profit to Sales

X90 - Net Income to Stockholder's Equity

X91 - Liability to Equity

X92 - Degree of Financial Leverage (DFL)

X93 - Interest Coverage Ratio (Interest expense to EBIT)

X94 - Net Income Flag: 1 if Net Income is Negative for the last two years, 0 otherwise

X95 - Equity to Liability

2.2 Decision Tree Classifier:

Decision tree is a supervised machine learning algorithm used for both regression and classification tasks. It is a tree-like structure where decision based on the value of particular feature is represented by internal node, outcome of that decision is represented by branch, and final output or the predicted output is represented by the leaf.

In this project, we have imported DecisionTreeClassifier from sklearn.tree and used DecisionTreeClassifier(*max_depth, min_sample_leaf, min_sample_split*) function. These parameters are defined as follows:

1. criterion: {"gini', "entropy", "log_loss"}, default= "gini"

   It is used to measure the quality of split, which is calculated by given entropy information. Criteria are "gini" for Gini impurity and "log_loss" and "entropy" for Shannon information gain.

   Measures of impurity are following-

   Gini:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$$

   Log loss or Entropy:

$$H(Q_m) = -\sum_k p_{mk} \log(p_{mk})$$

Where m is terminal node, H() is impurity function or loss function and p_{mk} is prediction probability.

2. max_depth: int, default= None

   This parameter is used to define the maximum depth of the decision tree. If it is not defined then node are expanded until all leaves contains less than min_sample_split sample. Here, max_depth= 5 which means that the tree can have maximum of 5 decision nodes from root to leaf. This parameter controls the complexity of the tree and avoids overfitting.

3. min_sample_leaf:

   int or float, default=1

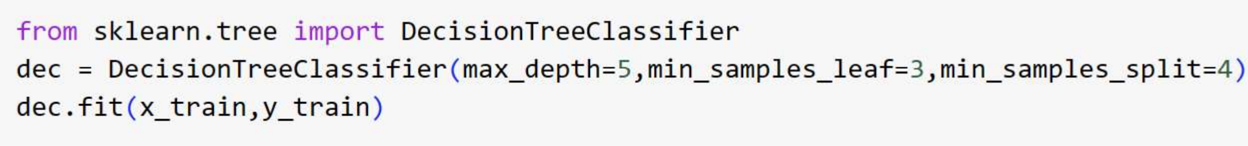
   This parameter is used to specify the minimum number of samples required to create the leaf node. Here, min_sample_leaf=3 which means that if, during construction of the tree , a split would output in a leaf node with fewer than 3 samples, the leaf is not performed and the node becomes a leaf.

4. min_samples_split:

   int or float, default=2

   This parameter is used to specify the minimum number of samples required to perform split at an internal node. Here, min_sample_split=4 which means that if a node have fewer than 4 samples, it won't be spilt further. This controls the granularity of the splits and prevents creating too many small branches in the tree.

Code:



```
from sklearn.tree import DecisionTreeClassifier
dec = DecisionTreeClassifier(max_depth=5,min_samples_leaf=3,min_samples_split=4)
dec.fit(x_train,y_train)
```

2.3 Random Forest Classifier:

Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and showing the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The "forest" it builds is an ensemble of decision trees, generally trained with the "bagging" method.

In this project, we have imported RandomForestClassifier from sklearn.sensemble and used Random_Forest_Classifier(*random_state, n_estimators, max_depth, bootstrap*) function. These parameters are defined as follows:

1. random_state: int, RandomState instance or None, default= None

   This parameter is used to set the seed value for the number random number generator. Here, random_state=42 which means that the random processed involved in building Random Forest are reproducible. It controls randomness as well as boostrapping of the samples used when building trees.

2. n_estimators: int, default=100

   This parameter defines number of decision trees in the forest. Here, n_estimators=61 which means that 61 trees have been specified. When larger number of trees are specified it leads to a more robust model, but also increases computational cost.

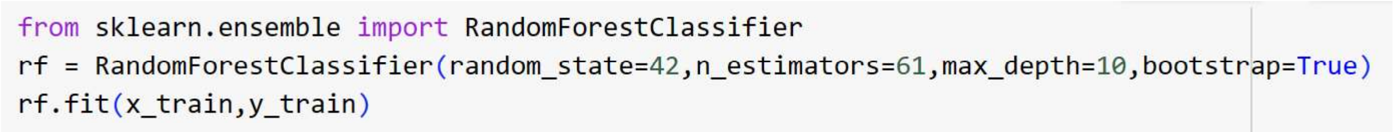
3. max_depth: int, default= None

   This parameter is used to define the maximum depth of each individual decision tree in the Random Forest. If it is not defined then node are expanded until all leaves contains less than min_sample_split sample. Here, max_depth= 10 which means that the tree can have maximum of 10 decision nodes from root to leaf. This parameter controls the complexity of the tree and avoids overfitting.

4. bootstrap: bool, default= True

   This parameter is used to define whether bootstrap samples are used when building trees. If bootstrap value is False, the whole dataset is used to build each tree.

Code:



```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state=42,n_estimators=61,max_depth=10,bootstrap=True)
rf.fit(x_train,y_train)
```

2.4 Support Vector Machines (SVM) Classifier:

Support Vector Machines (SVM) are a powerful class of supervised machine learning algorithm which is used for the both classification and regression task. The main objective of this algorithm is to find optimal hyperplane in an N-dimensional space that best separates the data into different classes. The hyperplane tries that the difference between closest points of different classes should be maximum. SVMs can be applied to learn a decision boundary that separates financially distressed companies from healthy ones.

In this project, we have imported SVC from sklearn.svm and DecisonBoundaryDisplay from sklearn.inspection and used SVC(*kernel, gamma, c*) function. These parameters are defined as follows:

1. kernel: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable, default= ‘rbf’

This parameter is used to specify the kernel used in the algorithm. If none, ‘rbf’ will be used. Here, kernel= ‘rbf’ which stands for Radical Basis Function. It maps input features into high-dimensional space where a hyperplane can be used to separate different classes.

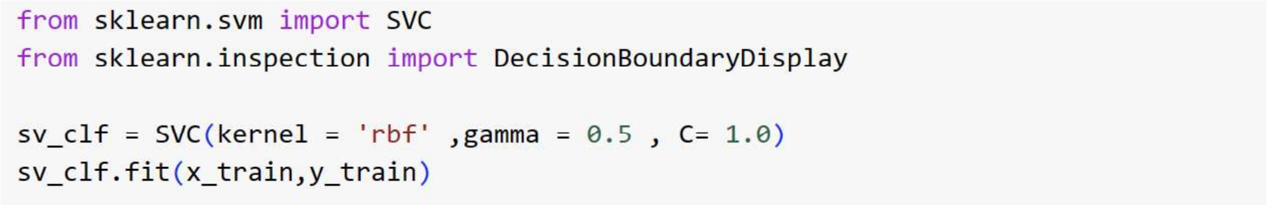
2. gamma: {‘scale’, ‘auto’} or float, default= ‘scale’

   This parameter is used to define the influence of a single training example. It determines the shape of the decision boundary. If the gamma value is smaller, it means a large similarity radius, which leads to smoother decision boundary, while if the value is larger, it means a more complex and intricate decision boundary.

3. c: float, default= 1.0

   This parameter is the regularization parameter which is also known as cost parameter. It must be positive value. Regularization is inversely proportional to c. Here, c= 1.0 which controls the trade-off between having smooth decision boundary and classifying the training points. If the value of c is smaller, it encourages a smoother decision boundary, while if the value of c is larger, it encourages harder margin by penalizing misclassification more heavily.

Code:



```
from sklearn.svm import SVC
from sklearn.inspection import DecisionBoundaryDisplay

sv_clf = SVC(kernel = 'rbf' ,gamma = 0.5 , C= 1.0)
sv_clf.fit(x_train,y_train)
```

Chapter -3

RESULT:

3.1 Confusion Matrix

A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying number of accurate and inaccurate instances from the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance

The prediction results of Confusion Matrix can be classified into four categories: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

The matrix displays, the number of instances produced by the model on the test data.

- true positives (TP): occurs when the model accurately predicts a positive data point.
- true negatives (TN): occurs when the model accurately predicts a negative data point.
- false positives (FP): occurs when the model predicts a positive data point incorrectly.
- and false positives (FP)): occurs when the model predicts a negative data point incorrectly.

It has to be noted that Negative (Positive) means corporate bankruptcy (non-bankruptcy).3 Among them, the evaluation variables used by the confusion matrix after calculation include Accuracy, Precision, Recall, F1-Score, Type I Error, and Type II Error. When TP is true positive (i.e., the model correctly identified a firm that went bankrupt), FN is a False negative (the model failed to identify a firm that went bankrupt), FP is a false positive (the model incorrectly classified a firm as having gone bankrupt, when it did not); and TN is a true negative (the model correctly identified a firm that did not go bankrupt).

Let's define important metrics:

Accuracy - this measures the total number of correct classifications divided by the total number of cases. the accuracy rate is defined as the percentage of correct predictions among all samples.

$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall/Sensitivity - this measures the total number of true positives divided by the total number of actual positives. The recall rate indicates how many positive (negative) samples the model is able to successfully predict from actually positive (negative) samples.

$$\textit{Recall} = \frac{\text{TP}}{\text{TP + FN}} \text{ or } \frac{\text{True Positive}}{\text{Actual Results}}$$

Precision - this measures the total number of true positives divided by the total number of predicted positives. The precision rate indicates how many of the samples predicted to be positive (negative) by the model are actually positive (negative) samples

$$\textit{Precision} = \frac{\text{TP}}{\text{TP + FP}} \text{ or } \frac{\text{True Positive}}{\text{Predictive Results}}$$

Specificity - this measures the total number of true negatives divided by the total number of actual negatives.

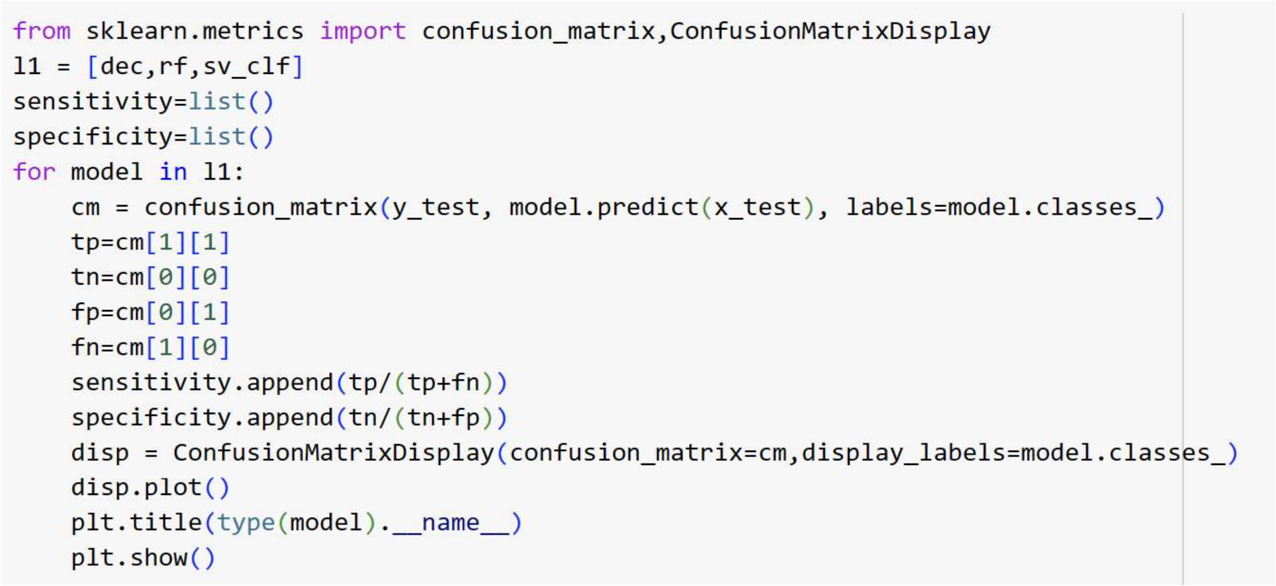
$$\textit{Specificity} = \frac{\text{TN}}{\text{TN + FP}}$$

F1 Score - is a single metric that is a harmonic mean of precision and recall. The F1-Score is a weighted average of the precision rate and recall rate

$$\textit{F - score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall + Precision}}$$

The machine learning analyses in this research focus on model accuracy and misclassification. The misclassification scenarios are Type I error and Type II error, where Type I error means that a real bankrupt firm is predicted to be a non-bankrupt firm and Type II error means that a real nonbankrupt firm is predicted to be a bankrupt firm.

Code:



```
from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
l1 = [dec,rf,sv_clf]
sensitivity=list()
specificity=list()
for model in l1:
    cm = confusion_matrix(y_test, model.predict(x_test), labels=model.classes_)
    tp=cm[1][1]
    tn=cm[0][0]
    fp=cm[0][1]
    fn=cm[1][0]
    sensitivity.append(tp/(tp+fn))
    specificity.append(tn/(tn+fp))
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=model.classes_)
    disp.plot()
    plt.title(type(model).__name__)
    plt.show()
```

Let find out these matrices for our models which we used to calculate the accuracy, precision and recall that are

3.2 Decision Tree

true positives (TP): 1196

true negatives (TN): 1226

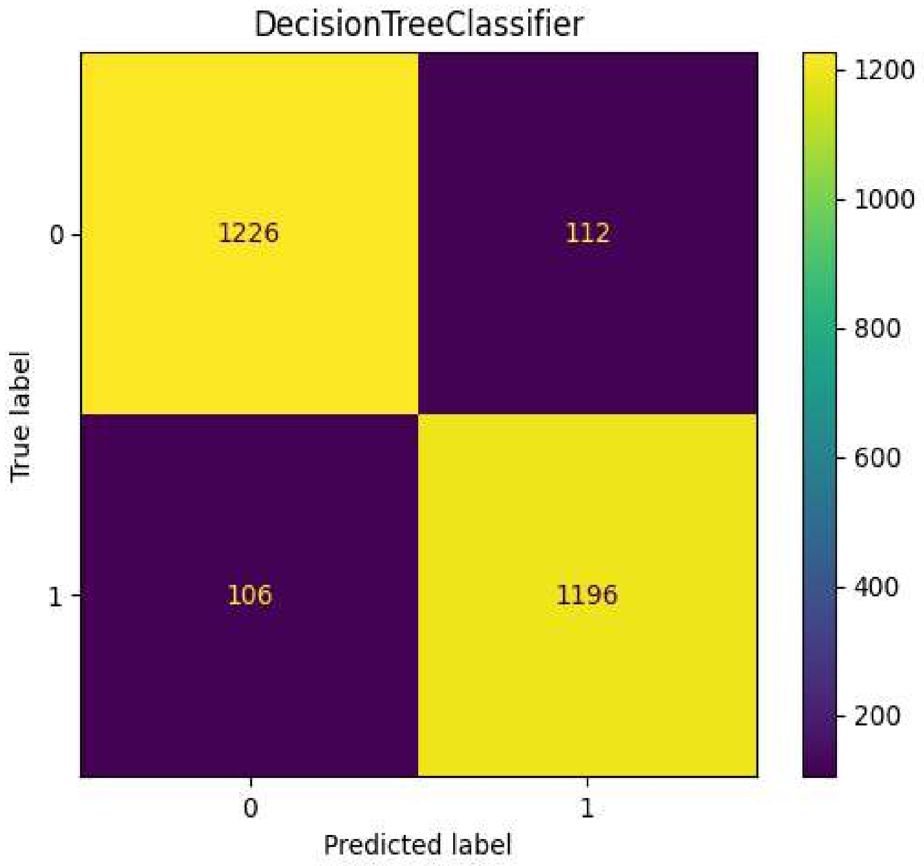
false positives (FP): 112

false negatives (FN): 106

So, by the formula mentioned above, accuracy = 0.917424

Precision = 0.914373

Recall = 0.918587



(Fig 3.1: Confusion matrix for Decision Tree)

3.3 Random Forest Classifier:

true positives (TP): 1282

true negatives (TN): 1236

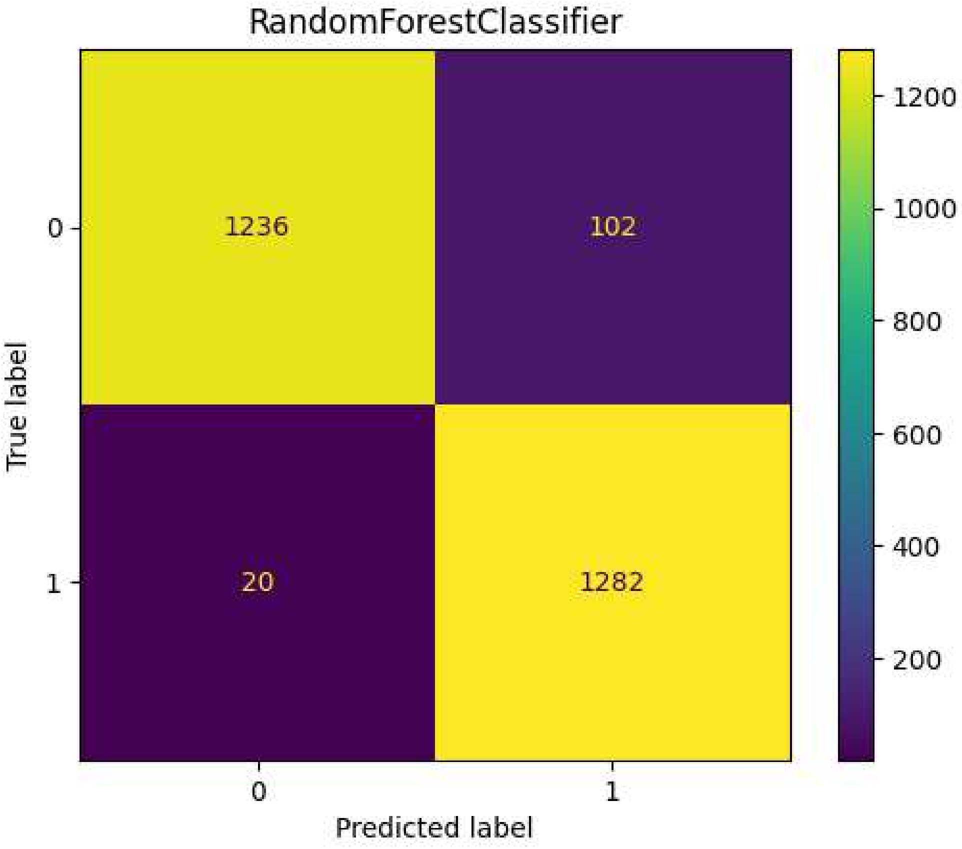
false positives (FP): 102

false negatives (FN): 20

So, by the formula mentioned above, accuracy = 0.953788

Precision = 0.926301

Recall = 0.984639



(Fig 3.2: Confusion matrix for Random Forest)

3.4 Support Vector Classifier:

true positives (TP): 1302

true negatives (TN): 27

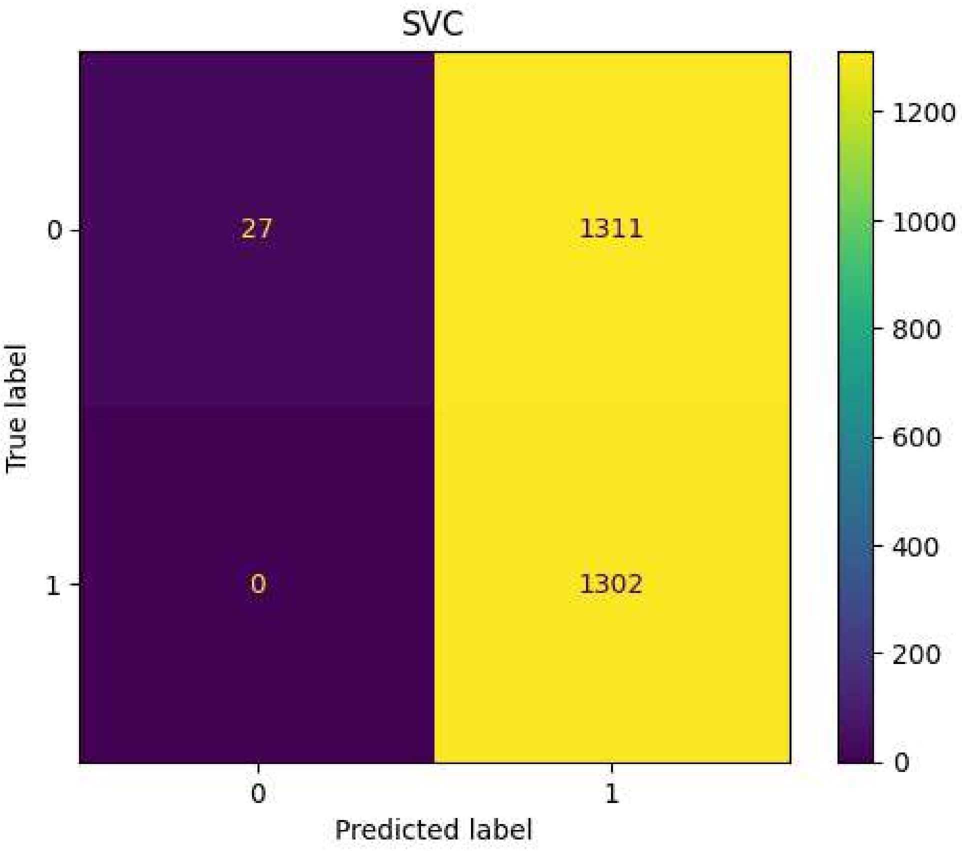
false positives (FP): 1311

false negatives (FN): 0

So, by the formula mentioned above, accuracy = 0.503409

Precision = 0.498278

Recall =   0.000000



(Fig 3.3: Confusion matrix for SVC)

The above model we used, among them random forest classifier works best in the bankruptcy prediction as it gives the maximum accuracy, precision or recall.

Chapter -4

PROJECT ANALYSIS

This section presents the bankruptcy prediction results employing the above mentioned four machine learning models under different conditions, and further analyzes and discusses whether annual variables enhance the effectiveness of bankruptcy prediction under the model In addition to the model accuracy rate, this study specifically focuses on F1-score, Type I error, and Type II error. To provide stable model results, this study implements the prediction effectiveness analyses by the average of each evaluation

ROC curve:

The receiver operating characteristic (ROC) curve, called ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier. ROC curve is generated by plotting TPR against FPR at various thresholds. We used Scikit-learn packages to generate the ROC curves for all models for the validation set. The area under the curve (AUC) score was calculated for each model and for the validation set. High AUC is good.

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. For example, given the following examples, which are arranged from left to right in ascending order of logistic regression predictions:

ROC or Receiver Operating Characteristic curve represents a probability graph to show the performance of a classification model at different threshold levels. The curve is plotted between two parameters, which are:

- True Positive Rate or TPR
- False Positive Rate or FPR

In the curve, TPR is plotted on Y-axis, whereas FPR is on the X-axis.

TPR:

TPR or True Positive rate is a synonym for Recall, which can be calculated as:

$$TPR = \frac{TP}{TP + FN}$$

FPR:

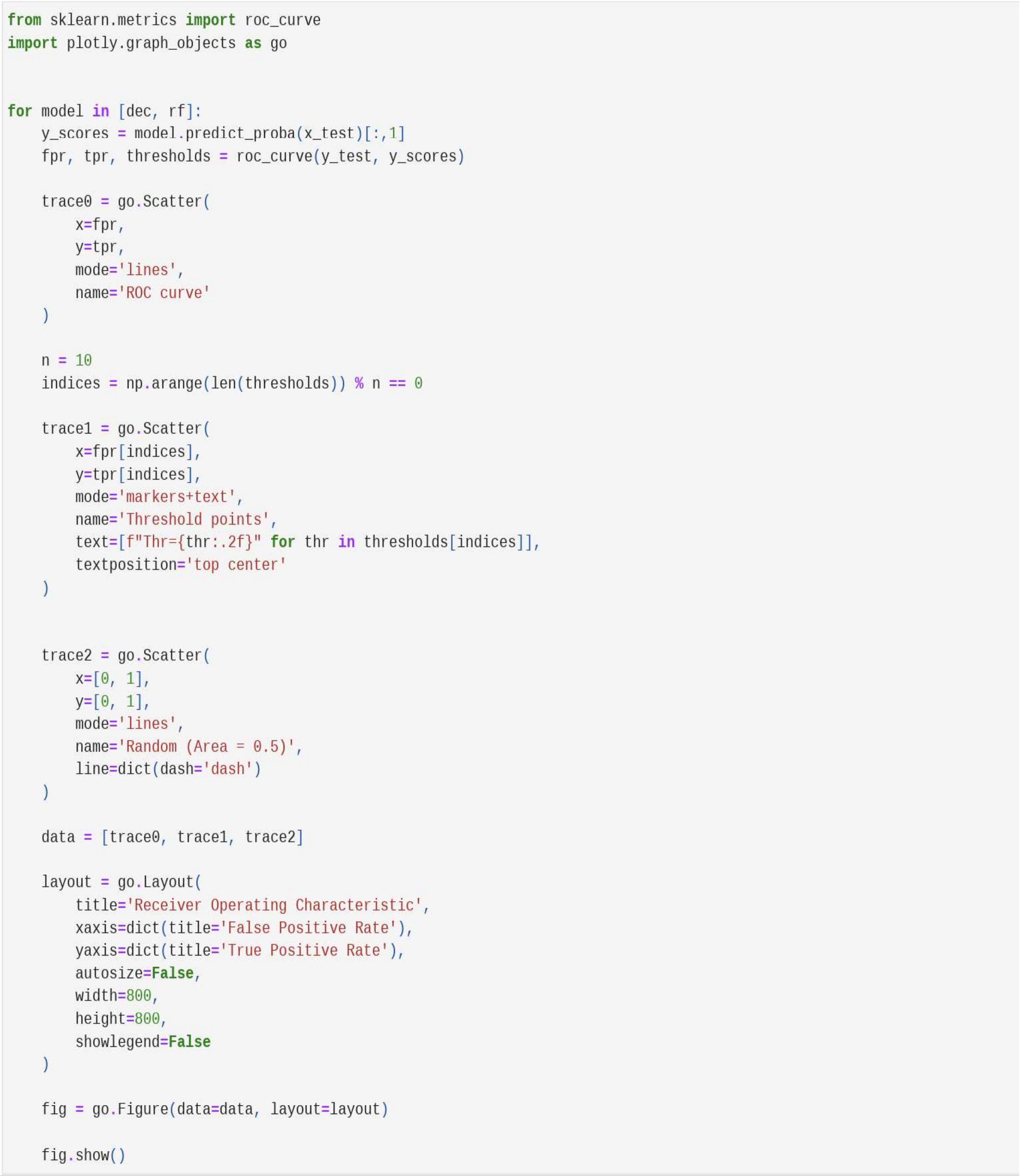
FPR or False Positive Rate can be calculated as:

$$FPR = \frac{FP}{FP + TN}$$

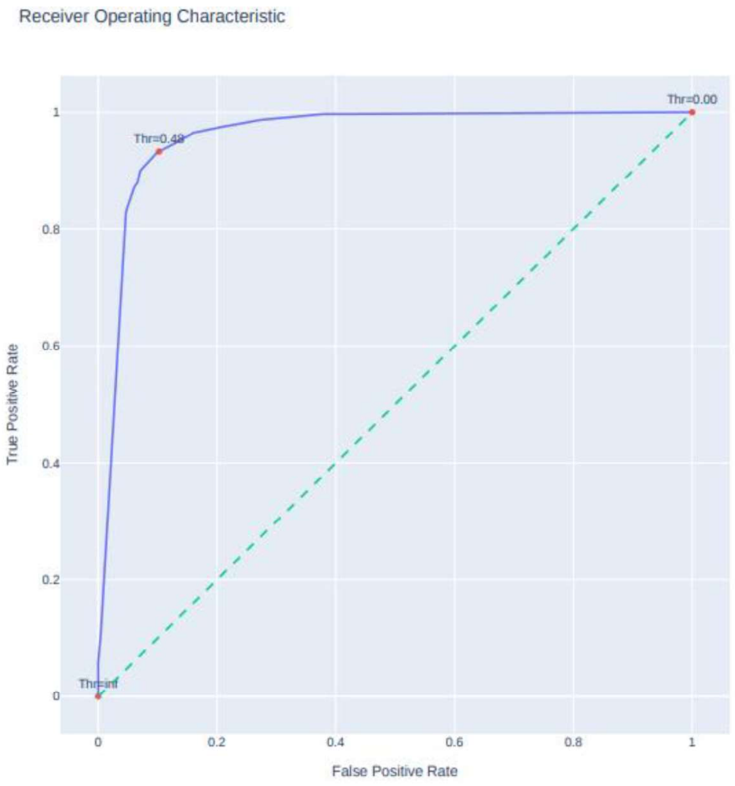
An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

In the following two models, decision tree classifier and random forest classifier we have calculated TPR and FPR at different thresholds and mark the intersection point of these values then we have plotted a ROC graph to know the correct prediction and also the area undercurve. And the alignment of ROC curve is more towards the 1, more accurate the prediction is. The area under curve of random forest classifier is more than the decision tree classifier so the RF model curve has more accurate prediction.

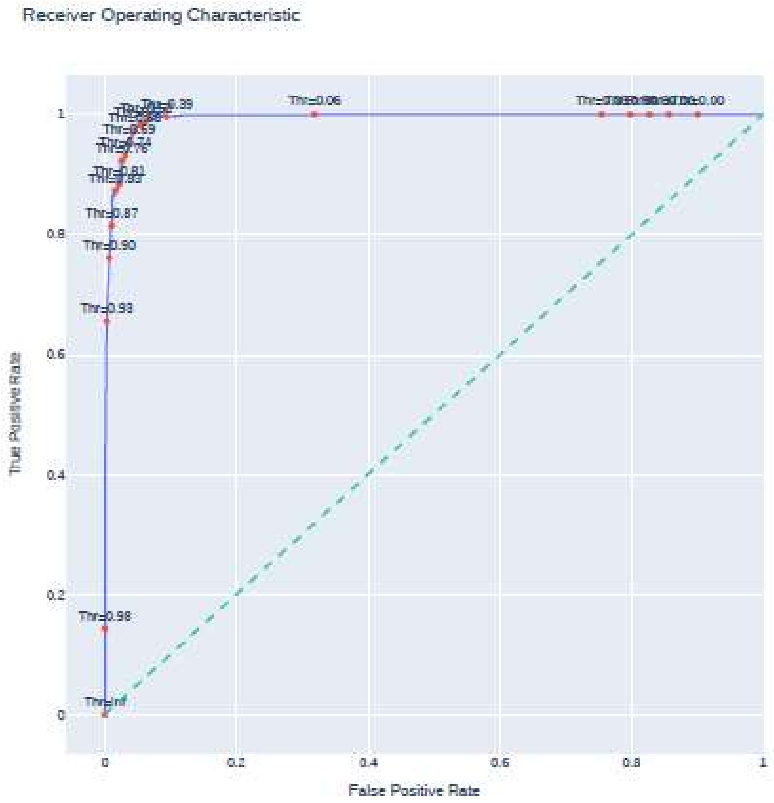
Code:



```
from sklearn.metrics import roc_curve
import plotly.graph_objects as go


for model in [dec, rf]:
    y_scores = model.predict_proba(x_test)[:,1]
    fpr, tpr, thresholds = roc_curve(y_test, y_scores)

    trace0 = go.Scatter(
        x=fpr,
        y=tpr,
        mode='lines',
        name='ROC curve'
    )

    n = 10
    indices = np.arange(len(thresholds)) % n == 0

    trace1 = go.Scatter(
        x=fpr[indices],
        y=tpr[indices],
        mode='markers+text',
        name='Threshold points',
        text=[f"Thr={thr:.2f}" for thr in thresholds[indices]],
        textposition='top center'
    )


    trace2 = go.Scatter(
        x=[0, 1],
        y=[0, 1],
        mode='lines',
        name='Random (Area = 0.5)',
        line=dict(dash='dash')
    )

    data = [trace0, trace1, trace2]

    layout = go.Layout(
        title='Receiver Operating Characteristic',
        xaxis=dict(title='False Positive Rate'),
        yaxis=dict(title='True Positive Rate'),
        autosize=False,
        width=800,
        height=800,
        showlegend=False
    )

    fig = go.Figure(data=data, layout=layout)

    fig.show()
```



(Fig 4.1: Roc curve for decision tree classifier)



(Fig 4.2: Roc curve for random forest classifier)

We trained three different models (DT, RF, SVM) on three different datasets . each dataset into subsets, and conducted cross validation (John Lu, 2010). In each step, we used three subsets as a training set, one subset as a validation, and one subset as a testing set. AUC was calculated when the models were evaluated on the test dataset and averaged over experiments.

As we can see machine learning algorithms such as Random Forest classifier achieved higher accuracy than the traditional logistic regression model.

Chapter -5

CONCLUSION

5.1 CONCLUSION

The purpose of this study is to explore whether the annual report text-based communicative value increases the predictive power of the bankruptcy prediction models with machine learning algorithm and captures more signals. In the recent years, although it is widely discussed in academic literature for bankruptcy predictions using various machine learning models; however, most of them only focus on financial variables rather than non-financial variables, which leads to a huge room for further improvement on the adoption of input variables

According to economics theory, the bankruptcy prediction model is divided into an intensity-based models, a structural models, and a statistical models Intensity based models use the bond market information; the structural model uses the stock price information; and the Altman Z-Score model (Altman, 1968), which is a representative statistical method, reflects the company's market capital information in the variable that indicates the leverage ratio. This study has limitations in using only the financial statement information, and it is expected that the accuracy of forecasting of bankruptcy will be improved when market indicators such as market capitalization, and GDP growth are used. Therefore, we can conclude that the annual report text-based communicative value indeed has a significant influence on corporate bankruptcy prediction.

5.2 FURTURE SCOPE

Despite the differences in the bankruptcy prediction models, the empirical tests of most of the models show high predictive ability. This would suggest that the models would be useful to many groups including auditors, managers, lenders, and analysts. However, it appears that bankruptcy prediction models are not utilized in practice on a widespread basis. Further, despite the vast amount of literature and models that have been developed, researchers continue to look for "new and improved" models to predict bankruptcy. With the number of models already available and the apparent limited use in practice, the question is raised: "Why do we continue to develop new and different models for bankruptcy prediction?"

We believe that the focus of future research should be on the use of existing bankruptcy prediction models as opposed to the development of new models. There are over 150 models available, many of which have been shown to have high predictive ability. Future research should consider how these models can be applied and, if necessary, refined. Researchers should consider the fact that a large number of factors does not necessarily increase a model's predictive ability. Beaver [1966J was able to predict bankruptcy with 92% accuracy using only one ratio. Jo et al.'s [1997J model that considered 57 factors yielded only an 86% accuracy rate. As Jones

[1987, p. 140J points out, "using too many ratios can actually make a model less useful." Lastly, future researchers should attempt to establish a stronger connection between research and practice, similar to other fields such as engineering and medicine. Bankruptcy prediction models could be very useful in practice provided they receive the proper exposure to auditors, managers, lenders, and analysts.

5.3 REFERENCES

Source:

Deron Liang and Chih-Fong Tsai, deronliang '@' gmail.com; cftsai '@' mgt.ncu.edu.tw, National Central University, Taiwan
The data was obtained from UCI Machine Learning Repository:

https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction

Relevant Papers:

Liang, D., Lu, C.-C., Tsai, C.-F., and Shih, G.-A. (2016) Financial Ratios and Corporate Governance Indicators in Bankruptcy Prediction: A Comprehensive Study. European Journal of Operational Research, vol. 252, no. 2, pp. 561-572.
https://www.sciencedirect.com/science/article/pii/S0377221716000412