Analysis of Financial Health of Banks in India

Prajwaldeep Kamble[1], Mulugu Vishwanath Sharma[1], Lokesh Surana[1], and
Supervisor: Dr. Sameen Naqvi[1]

[1]Indian Institute of Technology, Hyderabad

Abstract

This project aimed to analyze and classify the financial health of banks in India using publicly available financial data, such as balance sheets and results submitted to the exchange by the banks.

To accomplish this, we reduced the number of financial indicators through financial relevance and correlation analysis and used the remaining indicators to classify the banks based on their relative return to the benchmark index, NIFTY, while assuming the efficient market hypothesis.

We then applied various machine learning algorithms, including boosting and bagging, to classify the banks based on their financial health. The results showed that the non-binary approach using decision tree methods, specifically XgBoost, provided the best accuracy of 88%.

This approach allows for a more nuanced understanding of a bank's financial standing and can be used by investors and policymakers to make informed decisions. Overall, this project highlights the importance of financial health analysis and the value of machine learning techniques in this field.

1. Introduction

The financial health of banks is a critical aspect of the overall health of the financial system. Banks are responsible for collecting deposits, making loans, and facilitating transactions, which are essential activities in any economy. As such, the financial health of banks has a significant impact on the stability and growth of the economy. Therefore, it is essential to assess the financial health of banks to ensure that they are sound and can perform their critical functions effectively.

Assessing the financial health of banks involves analyzing various parametric ratios and indicators. These ratios and indicators provide insights into the banks' financial performance and can help identify potential issues or areas of concern. Understanding the financial health of banks is essential for investors, regulators, and other stakeholders who rely on banks to manage their money and facilitate transactions.

In this report, we will explore the use of machine learning models for predicting the financial health of banks. We will analyze a variety of financial data and market data to train and evaluate machine learning models. The models will be trained using a range of algorithms, including decision trees, random forests, and Support vectors, to identify the most accurate and effective models for predicting financial health.

The report will also examine the various factors that can impact the accuracy of machine learning models in predicting financial health. These factors include the quality and completeness of the data used to train the models, the selection of appropriate variables and features, and the choice of algorithms and parameters.

Ultimately, the report aims to provide valuable insights into the use of machine learning models for predicting the financial health of banks. By examining the strengths and weaknesses of these models, we hope to contribute to a better understanding of how these technologies can be used effectively in the financial industry to improve decision-making and risk management.

2. Literature Review

2.1. Data mining methods in the prediction of Dementia [2]

According to the paper, factors that can affect the performance of data mining methods in classification tasks include the choice of predictors, data assumptions, sample size, and parameter tuning. The performance of data mining methods such as neural networks and support vector machines can be heavily dependent on the chosen values for tuning parameters. The study also suggests that for classification tasks where the classes can be linearly separated and sample size may compromise training and testing, Random

Forests and Linear Discriminant Analysis may be more effective than other data mining methods. Additionally, the skill of the analyst who applies the data mining methods and their ability to properly tune parameters can also have an impact on overall performance.

The paper suggests that sample size plays an important role in the accuracy of Neural Networks. The study used a relatively small sample size, which was in the lower limit for recommended data set dimensions for Neural Networks applications. The number of cases for the training and testing sets are at a lower limit for recommended data set dimensions for Neural Networks applications, which may limit the performance of some data mining methods assessed in this study. Larger datasets requirements are also found in Logistic Regression, but less in Linear Discriminant Analysis if the model assumptions are met. However, there are studies with relatively small samples where data mining techniques, like SVM and Neural Networks, have been used with high accuracy in classification problems. So, while a larger sample size may improve the accuracy of Neural Networks, smaller samples have still produced high accuracy in certain studies.

Data mining classifiers like Logistic Regression, Neural Networks, Support Vector Machines, and CHAID trees showed low sensitivity and are not recommended for predicting conversion into dementia. Instead, Random Forests and Linear Discriminant Analysis were found to be more suitable for this binary classification task as they had high accuracy, sensitivity, specificity, and discriminant power.

Overall, the paper suggests that traditional classifiers such as LDA can hold up against newer, computationally intensive classifiers in medical classification problems.

2.2. Discriminant analysis as a tool for forecasting companies' financial health [1]

Discriminant analysis is a useful tool in assessing the financial health and predicting the potential bankruptcy of a company. It is a statistical method that allows a company to predict whether an element belongs to the advanced set group, and in this case, whether or not a company is heading toward bankruptcy.

The aim is to find a prediction model that can classify new objects into classes based on their degree of similarity to existing groups. The essence of discriminant analysis is to examine the dependence of one qualitative variable on several quantitative variables. The goal is to find the optimal attributing rules that will minimize the likelihood of erroneous classification of elements.

The canonical discriminant analysis is used to statistically separate two or more groups and can help determine variables that have the highest ability to distinguish the groups to which the object belongs.

2.3. Application of Discriminant Analysis to Diagnose the Financial Distress [4]

The study ”Application of Discriminant Analysis to Diagnose the Financial Distress” by Rashmi Soni, Pompe, and Bilderbeek observed the different phases of bankruptcy for small and medium size firms and found that each calculated ratio has indicative power of financial distress. They studied the predictive power of different financial ratios and found that the different phases of bankruptcy can be predicted using ratio analysis. Altman's Z-score model is considered an effective tool for predicting bankruptcy and financial distress. The study highlights the accuracy of the model in different countries. The study uses discriminant analysis and investigates the different ratios suggested by Altman to predict financial distress and the possibility of bankruptcy of Ruchi Soya Ltd. The study concludes that the company was facing severe financial distress and was at high risk of bankruptcy. Therefore, calculated ratios are essential in determining financial distress, and Altman's Z-score model has proven to be a reliable and effective tool in predicting financial distress and bankruptcy.

Various studies have used the Altman Z-score model to predict financial distress and solvency positions of companies like IOCL and BSE Small Cap Index. G R Bal and Raja studied the earnings management and techniques used to predict the solvency position of IOCL using the Z-score and concluded that the financial position of the company was not good. Dr. M M Sulphey Nisa assessed the solvency position of 220 companies listed in the BSE Small Cap Index using Z-score and found that only 79 companies were in the safe zone, 117 companies were in the grey zone, and 24 were in the distress zone. Amalendu Bhunia et al. studied the capability to detect potential financial problems at a premature stage and found that the Z-score model showed good performance with a highly correct categorization factuality rate of more than 80%. Setyani Dwi et al. aimed to obtain empirical evidence about the state of financial distress prediction using the Altman Z-score and ratio-ratio test Z-score in influencing the price of shares in the chemical subsectors listed in the Indonesia Stock Exchange 2009-2014 period.

The results of these studies, particularly the application of the Altman Z-score model, can be used by potential investors while making investment decisions. The model has been found to be an effective tool in predicting financial distress and the possibility of bankruptcy of companies. The ratios suggested by Altman which have been found significant to discriminate among failed and non-failed companies are working capital to total asset, retained earnings to total asset, earnings before interest and tax to total assets, market value of equity to book value of debt, and sales to total assets. These ratios can be used by investors to assess the financial health of a company and make informed decisions about investing in it. Altman's original formula for manufacturing firms is

$$Z = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 0.999X_5 \quad (1)$$

1. X_1 represents working capital to total assets
2. X_2 represents retained earnings to total assets
3. X_3 represents earnings before interest and taxes to total assets
4. X_4 represents the market value of equity to the book value of debt
5. X_5 represents sales to total assets.

For non-manufacturing firms, the formula is different such as

1. X_1 as (Current Assets - Current Liabilities)/Total Assets
2. X_2 as retained earnings to total assets
3. X_3 as earnings before interest and taxes to total assets
4. X_4 as the book value of equity to total liabilities

2.4. A Rule-Based Model for Bankruptcy Prediction Based on an Improved Genetic Ant Colony Algorithm [5]

The data we are dealing with is of high dimensional and it is very complex to predict Bankruptcy. This paper introduces a new approach, the Genetic Ant Colony Algorithm (GACA). The process this paper followed is as follows

1. **Feature Selection**: Because the data is high-dimensional there is a high chance that the data is very sparse (i.e., more number of empty data/ NA) this will lead to many more problems. So, we select a few features to work on. This paper used a method of sequential feature selection.

2. **Rule-Based Approach**: As we know that NN is also called a ”Black Box”. Because we can’t get an inference from what’s happening inside. So, a rule-based approach is introduced to solve this problem. In this method, we can extract classification rules which are easier to understand

3. **Fitness-Scaling Chaotic GACA**: Zhang and Wu proposed the genetic ant colony algorithm (GACA) that combines the genetic algorithm (GA) and the ant colony algorithm (ACA). The GACA performs well to find global, but it is easy to be trapped in local minima. There are methods to improve the algorithm

   (a) Traditional feature selection forces the population to gather near the best individual. But the power-rank scaling method makes the population diverse.

   (b) Introduce chaos to increase the robustness of the GACA.

   The improved algorithm is called Fitness-scaling chaotic GACA (FSCGACA).

4. **Cross-Validation** Cross-validation is used to reduce the chances of overfitting. And selecting a novel powerful global optimization method with reduce the chances of the underfitting problem.

The main drawback of the paper is the absence of a solid background for the Rule-based approach used in the paper.

2.5. Using boosting algorithms to predict bank failure: An untold story [3]

The aim of the study published in the International Review of Economics and Finance is to investigate the heterogeneity results of boosting algorithms in bankruptcy prediction, specifically in predicting bank failure. The study compares the performance of different boosting algorithms, including AdaBoost, Gradient Boosting, and XGBoost, using a quarterly dataset of 180 US national banks from 2009 to 2019. The researchers carefully selected predictor variables and criteria for prediction quality to conduct a comprehensive study that clarifies existing debates on the accuracy of boosting algorithms in bankruptcy prediction. The study aims to provide insights that could be used to forecast bank failure in future studies.

The causes leading to bias in using XGBoost in bank failure prediction were identified as not assigning a correct value to target variables in some cases and the chosen predictor set not being optimal. Target variables were not clearly classified or classified inappropriately in previous studies, and predictor selection lacked optimization. To improve predictor selection, this study carefully selected predictors based on various techniques, resulting in seven out of 21 financial indicators, much fewer than existing studies. The predictors selected provided enough information for the boosting models to give near-absolute predictive results. Removing correlated variables in the set of predictors resulted in better predictions, which was not consistently mentioned in previous studies.

The study used cross-valuation to evaluate the quality of the prediction accuracy of boosting algorithms, including the confusion matrix, precision, recall and accuracy, and the area under the ROC curve (AUC) scores and receiver operating characteristic (ROC) curve. These parameters were evaluated in this same order. The confusion matrix showed that all of the boosting algorithms made early warnings for failed banks only, and no false predictions appeared in active banks. The AUC scores and ROC curve also demonstrated the accuracy of the predictions.

The study found that XGBoost is a powerful algorithm in predicting bank failure, with greater accuracy than AdaBoost and gradient boosting. However, the accuracy results vary across different studies due to technical factors such as data frequency, multicollinearity, and neutral thresholds. The paper suggests that quarterly data may offer better results in predicting bank failures. The study selected predictor variables and criteria for prediction quality to conduct a comprehensive study that clarifies existing debates on the heterogeneity results of boosting algorithms in bankruptcy prediction. The research is a continuation of a topic that has been widely studied but still poses technical and mathematical questions for future research.

3. Data

3.1. Intial phase

1. In the preliminary phase of our project, we procured balance sheet data for Indian banks from the reputable source https://screener.in.

2. The dataset comprised ten parameters, namely Share Capital, Reserves, Borrowings, Other Liabilities, Total Liabilities, Fixed Assets, CWIP, Investments, Other Assets, and Total Assets.

3. We discovered that some of these parameters were expressed in absolute terms, which could potentially be misleading in interpreting a bank's financial health. For instance, a bank with a higher share capital might erroneously appear to be performing better than a bank with a lower share capital, even if the latter had lower liabilities and was more financially stable.

4. Our initial attempts at modeling with this dataset yielded unsatisfactory results, prompting us to seek alternative, publicly available variables that would offer more relevant insights into a bank's financial standing.

3.2. Intermediate dataset

We opted to use a few fundamental ratios that are commonly used to evaluate the financial health of banks. These ratios include

1. **Net NPA (Non-Performing Assets):** A measure of the bad loans in a bank's portfolio

2. **Net Interest Margin:** The difference between the interest earned by a bank on its loans and the interest paid to depositors.

3. **Loan to Assets Ratio:** The percentage of a bank's assets that are tied up in loans. This is more important in economic slowdown scenarios.

4. **Return on Assets Ratio:** How efficiently a bank is using its assets to generate profits.

5. **CASA Ratio (Current Account Savings Account):** A measure of a bank's low-cost deposits compared to its total deposits.

6. **Capital Adequacy Ratio:** A measure of a bank's ability to absorb potential losses.

We believe that these ratios provide a more accurate and nuanced understanding of the financial standing of the banks we analyzed. In particular, these ratios are widely used in the banking industry and by regulatory bodies to assess the financial health of banks.

Later we obtained financial data from a wider range of sources, including balance sheets, fundamental ratios, yearly results, and PL statements, which we scraped from https://www.moneycontrol.com/ using the **Beautiful Soup** module in Python.

This approach allowed us to collect a more comprehensive set of financial data, which we then used to develop and test various machine learning models to predict the financial health of the banks in our dataset.

3.3. Final dataset

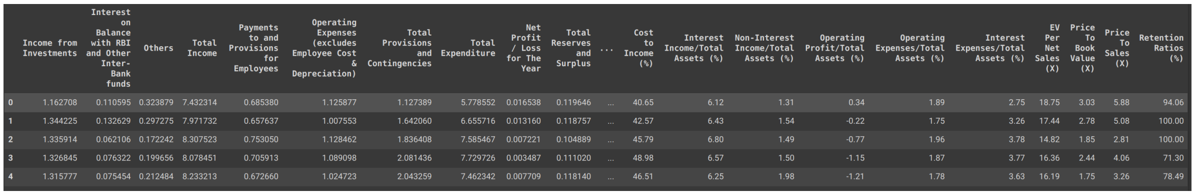
This provided us with a total of 166 parameters to work with. However, we encountered the issue of many of these parameters being highly redundant and correlated in nature, with many being simply related by mathematical operators to each other.

To address this issue, we used financial knowledge and analysis to remove several parameters based on several criteria, including repetitiveness, data availability, and financial relevance.

We also normalized some of the parameters relative to appropriate major parameters to address the earlier-mentioned issue with absolute parameters. Here is the change we made for our final data:

1. Dropped those columns which have NaN values for few of the banks for multiple years. Note that these were not important and mandatory parameters to publicize by banks, hence it was not available for all banks over all years.

2. Few columns were just acting as headers. For example, income was a blank column followed by other columns including income breakdown from other sources and then total incomes. All such section headers were removed.



	Income from Investments	Interest on Balance with RBI and Other Inter-Bank funds	Others	Total Income	Payments to and Provisions for Employees	Operating Expenses (excludes Employee Cost & Depreciation)	Total Provisions and Contingencies	Total Expenditure	Net Profit / Loss for The Year	Total Reserves and Surplus	...	Cost to Income (%)	Interest Income/Total Assets (%)	Non-Interest Income/Total Assets (%)	Operating Profit/Total Assets (%)	Operating Expenses/Total Assets (%)	Interest Expenses/Total Assets (%)	EV Per Net Sales (X)	Price To Book Value (X)	Price To Sales (X)	Retention Ratios (%)
0	1.162708	0.110595	0.323879	7.432314	0.685380	1.125877	1.127389	5.778552	0.016538	0.119646	...	40.65	6.12	1.31	0.34	1.89	2.75	18.75	3.03	5.88	94.06
1	1.344225	0.132629	0.297275	7.971732	0.657637	1.007553	1.642060	6.655716	0.013160	0.118757	...	42.57	6.43	1.54	-0.22	1.75	3.26	17.44	2.78	5.08	100.00
2	1.335914	0.062106	0.172242	8.307523	0.753050	1.128462	1.836408	7.585467	0.007221	0.104889	...	45.79	6.80	1.49	-0.77	1.96	3.78	14.82	1.85	2.81	100.00
3	1.326845	0.076322	0.199656	8.078451	0.705913	1.089098	2.081436	7.729726	0.003487	0.111020	...	48.98	6.57	1.50	-1.15	1.87	3.77	16.36	2.44	4.06	71.30
4	1.315777	0.075454	0.212484	8.233213	0.672660	1.024723	2.043259	7.462342	0.007709	0.118140	...	46.51	6.25	1.98	-1.21	1.78	3.63	16.19	1.75	3.26	78.49

Figure 1. Glimpse of the Final Data

3. Normalized the parameters to make sense out of it. For example, income of a bank is divided into multiple parts such as discount on advances, income from investments, interest on balance with RBI, and other sources of income. If we just have absolute values, it will show that banks with large books are always much better or much worse than smaller banks. Hence in this case, we made the total income as 1 and other income normalized compared to it. This shows the diversification of income for each bank and its in percentage terms which makes sense to compare.

4. Expenditure included interest expended, payments/provision to employees, depreciation, other provision and contingencies, etc. This was normalized against total expenditure.

5. Banks have assets in multiple sections such as cash with RBI, fixed assets, balances with banks money at short and call notice, and many more. Again, there is not much sense in having these values in absolute terms, so we normalized it relative to total assets.

6. There were few columns such as loss and profit forwarded from previous years, appropriations columns which states transfer of funds from one type to another such as statutory to capital reserve. This does not matter, only thing that should matter is the final values of parameters.

7. We removed the EPS parameters as EPS is earning per share which is more important from an investor's perspective. This is directly related to net profit and loss that we mentioned above, which just increases redundancy.

8. We removed the dividend-oriented parameters as in recent years, following Western global markets, Indian companies also started investing excess cash in business expansion to attain more growth rather than giving it back out to shareholders. Hence this parameter is not much strong to judge when we have other parameters such as capital, reserves, cash surplus.

9. Few of the parameters such as gross NPA, net NPA, income and expenditure sources were mentioned twice in two different sections. We identified them and removed them.

3.4. Classifier

In the classification section, we began with binary classification methods based on the Efficient Market Hypothesis (EMH). According to EMH, all information is priced in, and a bank's financial distress should be reflected in its market capitalization. This has been observed to hold true in the Indian markets, as evidenced by the examples of YES Bank and RBL Bank, which saw significant declines in their stock prices.

1. Our initial classifier identified a bank as bankrupt if its stock price fell by more than 40% in a week, assuming that such a drastic reaction from investors indicated a severe blowup event.

2. There is no specific data to support the choice of a 40% fall as a good measure to understand that a bank is doing bad. The decision was based on the assumption that a drastic drop in stock price would indicate a severe blowup event and intense reaction from investors.

3. For example, Yes Bank's stock price fell by around 83% in the first week of March 2020, while the Nifty index fell by around 9% during the same period. Therefore, the relative drop in Yes Bank's share price with respect to the Nifty was around 74%.

However, we recognized that few banks in India officially go bankrupt, and instead decided to shift our focus to classifying banks based on their financial health.

To evaluate a bank's financial health, we categorized them into three groups: good, neutral, and bad. The classification was based on their relative return compared to the benchmark index Nifty. Nifty was chosen as a benchmark for comparison because it is a well-established and widely followed index in the Indian market.

1. The analysis was based on the financial data of each bank for a one-year time period, from March to March.

2. If a bank's relative return was greater than 20%, it was classified as good.

3. If the relative return was between -20% to +20%, it was classified as neutral.
4. If the relative return was less than -20%, it was classified as a bad bank.

It's worth noting that these rankings were applied to financial data from the previous year, as the market prices are believed to reflect future expectations. We aimed to discount one year in our analysis, following the example of the Altman Z-score, which predicts financial distress up to two years in the future.

4. Our Methodology

While conducting the literature survey we got to know that the method of Neural networks is computationally very expensive so we tried to implement machine learning and statistical methods. Also, In the EDA (Exploratory Data Analysis) we did Principle Component Analysis and Factor Analysis, We found out that 95 per cent of the variance of data can be explained by 11 principal components. So we applied all the methods on the 54-features of the original data as well as the 11-principle components.

- **Logistic regression** Logistic regression is a popular binary classification method it uses the sigmoid function with an S-shaped curve for the classification. Also called logit, it can be extended to multi-class classification problems too.

- **Support Vector Machine** Like Logit SVM is also a popular Binary classification algorithm. Which can be extended to multi-class classification problems. The main idea of SVM is to find the Hyperplanes separating the classes, we have to find the best boundary such that separation is maximum.
  SVM is quite useful when the data is not linearly separable. The kernel trick is used to map the data into a higher dimensional space where it becomes linearly separable.
  We experimented with Various kernels, and the 'rbf' kernel gave the best results. The RBF kernel is a type of Gaussian kernel, which is defined as:
  $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$
  where x_i and x_j are feature vectors of the training examples, and γ is a hyperparameter that determines the width of the Gaussian kernel. The RBF kernel is used to compute the similarity between two examples as a function of their distance in feature space.

- **Discriminant Analysis** Discriminant analysis is a supervised learning method used to predict the class of a given data sample. It is used to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. It has 2 types:
  1. Linear Discriminant Analysis (LDA) in which the discriminant functions are linear functions of the predictor variables. it's popularly also called the Fischer Discriminant Analysis for binary classification, LDA can be extended for multi-class problems.
  2. Quadratic Discriminant Analysis (QDA) in which the discriminant functions are quadratic functions of the predictor variables. It is used when we do not assume that each class has its covariance matrix. Primarily used when data is not linearly separated.

- **Decision Trees** The decision tree is a powerful yet simple method for classification. It works by recursively partitioning the input space into smaller regions based on a set of decision rules learned from the training data. A tree consists of nodes that represent the decision rules, edges that represent the outcomes of the decision rules, and leaves that represent the predicted class labels or values. At each internal node of the tree, a decision is made based on the values of one of the input features, and the tree branches out to two or more child nodes based on the decision outcomes. We continue this process until a stopping criterion is met. That is a minimum number of samples at each leaf node Or a maximum depth of the tree Or a minimum improvement in the prediction accuracy. Here, we have implemented the algorithm with parameters defined as

  max_ depth=5, random_ state=42, criterion=entropy,
  splitter = best

  As we know decision trees often tend to overfit when the dataset is small. a lot and the same happened with our data set. So, we are trying to ensemble decision trees or boost the algorithms by joining decision trees to reduce overfitting and increase the accuracy of the output.

- **Random Forests** is a machine learning method that uses multiple decision trees to predict the output and then takes the average of all the outputs and gives the result. The randomness ensures that the trees in the forest are diverse and reduces the risk of overfitting.

  The following are a few Boosting methods that join many of the decision trees and give an accurate output. These algorithms use decision trees as their base weak learners.

  1. **XgBoost** is a more advanced ensembling algorithm. the structure is quite similar to the ran-

dom forests but the way of algorithmic design is different. It uses gradient boosting to ensemble trees hence giving better performance and reducing the risk of overfitting.
For multi-class classification, XgBoost uses the loss function as shown below

$$L = -\sum_{i}(y_i + log(p_i)) \tag{2}$$

Where y is the one-hot encoded vector of true labels and p_i is the predicted probability of class i.
and to avoid overfitting we have used a regularisation term/ penalty term which is a robust of L_1 and L_2 norms as shown below.

$$\gamma * \sum_{j} w_j^2 + \lambda * \sum_{j} abs(w_j) \tag{3}$$

Where w_j is the weight of j^{th} feature and γ, λ are the hyperprameters.

Since we are dealing with categorical data an algorithm that has higher performance for categorical data is preferred.

2. **CatBoost** is one of the boosting algorithms which is designed to perform better on categorical data. while the structure is similar to XgBoost but it saves a lot of time and reduces the risk of errors as XgBoost will convert the categorical data into numerical form.

   The loss we have used for cat boost for multi-class classification is

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \log \hat{y}_{ij} \tag{4}$$

   where N is the number of samples, K is the number of classes, y_{ij} is the indicator function for the true class of sample i (i.e., $y_{ij} = 1$ if the true class of sample i is j, and $y_{ij} = 0$ otherwise), and $\hat{y}_{ij}$ is the predicted probability of class j for sample i.
   The penalty/ regularization term in CatBoost is Berhu penalty [6] to prevent overfitting and it is similar to the penalty we have used for XgBoost and is shown below.

$$R(\mathbf{w}) = \frac{1}{2}\left(\alpha \sum_{i=1}^{n} |w_i| + (1-\alpha)\sum_{i=1}^{n} w_i^2\right) \tag{5}$$

   where $\mathbf{w}$ is the vector of model parameters (i.e., weights), n is the number of features, and α is a hyperparameter that controls the balance between L1 and L2 regularization.
   Now, we have another problem. Which is the data we are dealing with is very complex i.e., the dimensionality of the data is very high and the relationship between the features might not be linear. So, we are in need of a method that uses ensembling of decision trees but has high performance to complex data sets.

3. **AdaBoost** is a boosting algorithm that uses an ensembling of decision trees and has high performance to complex data like the financial or market data of an organization or company. But it has a few drawbacks like high sensitivity noisy data and outliers which causes it to overfit. And it is slower when compared to other ensembling methods like random forests and XgBoost. So, we have to gather data that is less noisy and remove the outliers from the data.
   In AdaBoost, the loss function used for multi-class classification is the exponential loss function, which is defined as:

$$L(y, f(x)) = \sum_{i=1}^{n} \exp(-y_i f_i(x)) \tag{6}$$

   where y_i is the true label of sample i, $f_i(x)$ is the predicted score for sample i, and $\exp$ is the exponential function.
   The penalty term in AdaBoost is not a regularization term but instead is incorporated in the weighting of the training samples. Specifically, AdaBoost assigns higher weights to misclassified samples in each iteration, so that the subsequent weak classifiers are forced to focus on the samples that are difficult to classify. The weights are updated as follows:

$$w_i^{(t+1)} = w_i^{(t)} \exp\left(\alpha_t y_i h_t(x_i)\right) \tag{7}$$

   where $w_i^{(t)}$ is the weight of sample i at iteration t, $h_t(x_i)$ is the prediction of the weak classifier at iteration t, and α_t is a scaling factor that is chosen to minimize the exponential loss function.

5. Results

Before going to the results lets go through some terms that were used in calculating the accuracies and implement the methods.

Kappa Score: It is a statistical measure of inter-rater agreement for categorical data it ranges from -1 to 1. 1 indicating perfect arrangement of the raters, 0 indicates agreement equal to chance and negative numbers means less agreement than chances.

Grid Search: It is a technique used to find the set of parameters which give the best accuracy of the testing data. it works by exhaustively searching over the given set of parameters and finding the best set of parameters.

5.1. Logistic Regression

- **Without PCA**

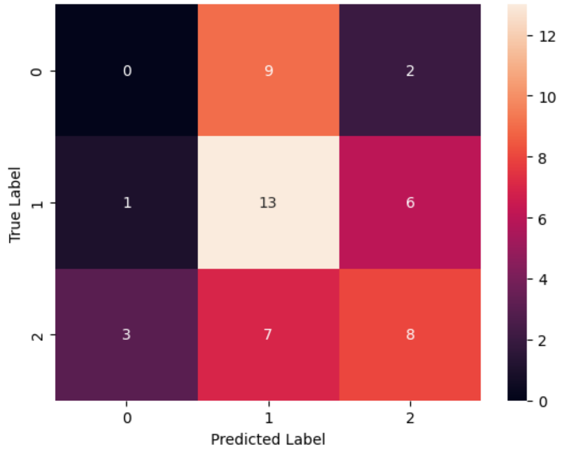


Figure 2. Logistic Regression Predictions Vs Truth Values

Best hyperparameters:
’C’: 0.1
’penalty’: ’l2’
’solver’: ’newton-cg’
Accuracies:
Test accuracy: 0.43
Kappa score: 0.08

- **With PCA**

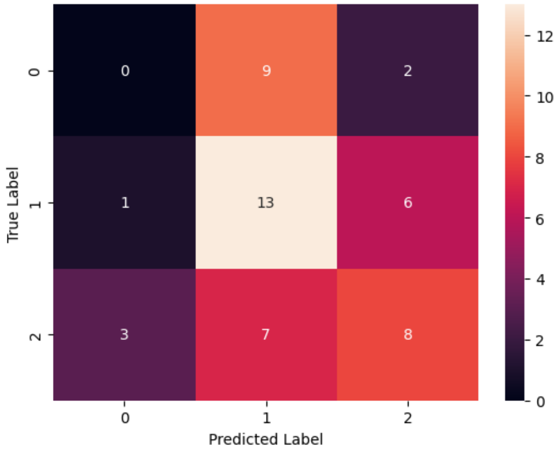


Figure 3. Logistic Regression Predictions Vs Truth Values

Best hyperparameters:
’C’: 0.1
’penalty’: ’l2’
’solver’: ’newton-cg’
Accuracies:
Test accuracy: 0.43
Kappa score: 0.08

5.2. Support Vector Machines

- **Without PCA**

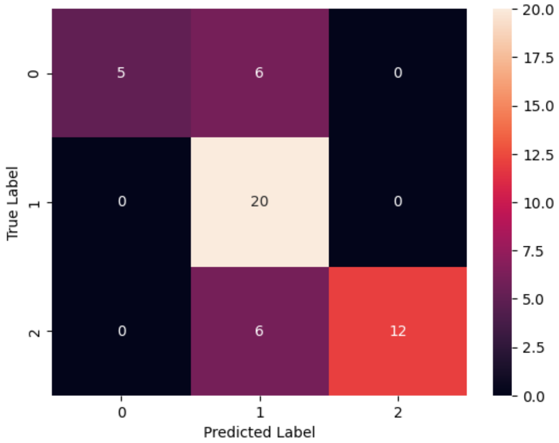


Figure 4. SVM Prediction Vs Truth Values

Best hyperparameters:
’C’: 10
’gamma’: 0.0001
’kernel’: ’rbf’
Accuracies:
Test accuracy: 0.76
Kappa score: 0.61

- **With PCA**

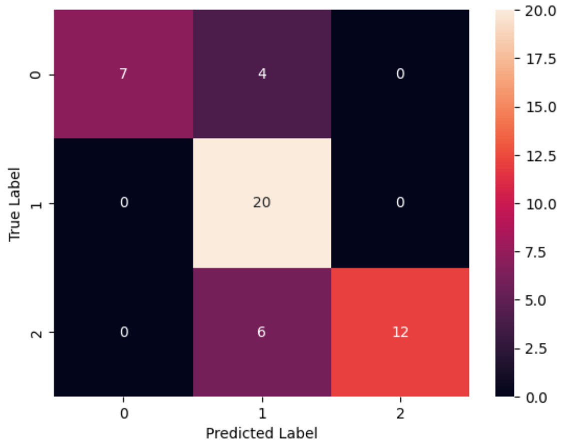


Figure 5. SVM Prediction Vs Truth Values

Best hyperparameters:
’C’: 1
’gamma’: 0.0001
’kernel’: ’rbf’
Accuracies:
Test accuracy: 0.80
Kappa score: 0.68

5.3. Linear Discrminant Analysis

- **Without PCA**

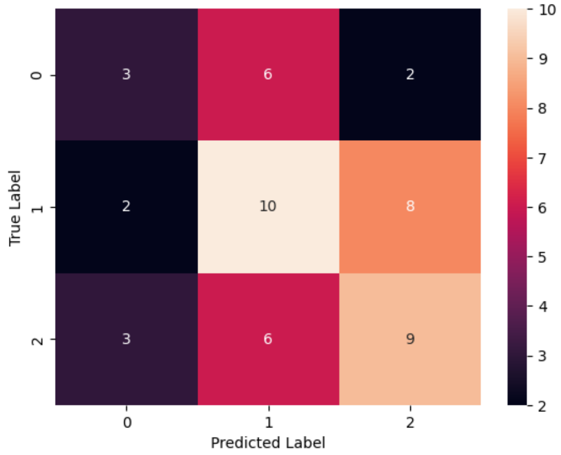


Figure 6. Linear Discriminant Analysis Predictions Vs Truth Values

Best hyperparameters:
’solver’: ’lsqr’
Accuracies:
Test accuracy: 0.45
Kappa score: 0.14

- **With PCA**

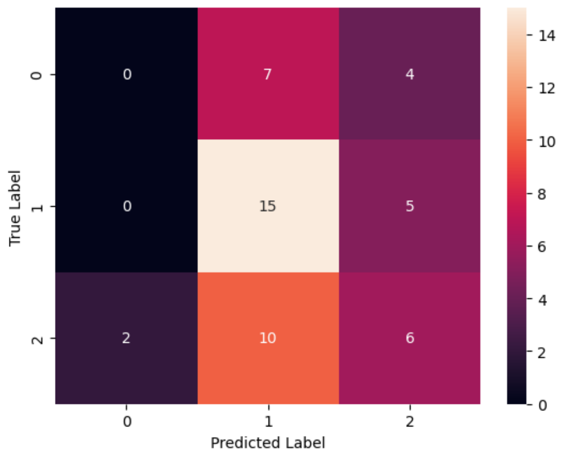


Figure 7. Linear Discriminant Analysis Predictions Vs Truth Values

Best hyperparameters:
’solver’: ’svd’
Accuracies:
Test accuracy: 0.43
Kappa score: 0.07

5.4. Quadratic Discrminant Analysis

- **Without PCA**

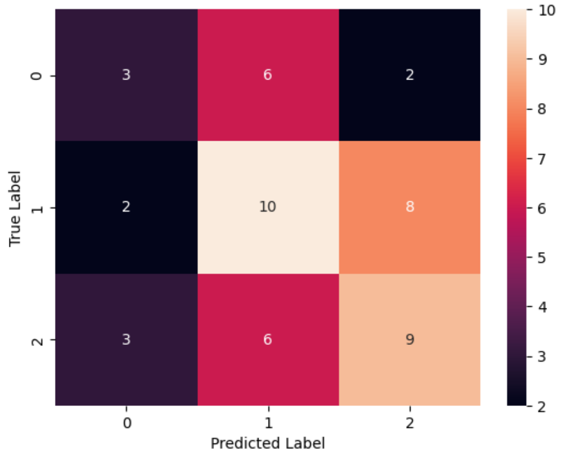


Figure 8. Quadratic Discriminant Analysis Predictions Vs Truth Values

Best hyperparameters:
’reg_param’: 0.1

Accuracies:
Test accuracy: 0.59
Kappa score: 0.35

- **With PCA**

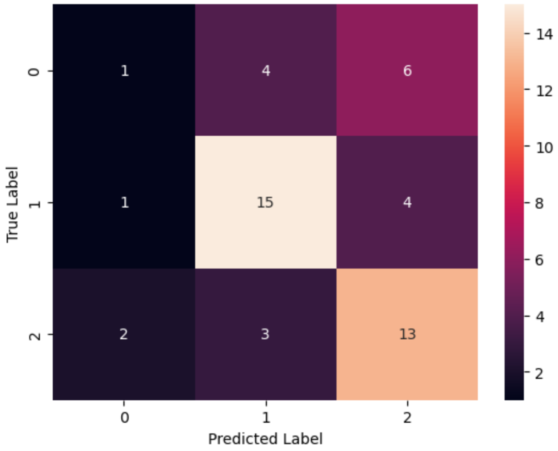


Figure 9. Quadratic Discriminant Analysis Predictions Vs Truth Values

Best hyperparameters:
’reg_param’: 0.1
Accuracies:
Test accuracy: 0.59
Kappa score: 0.35

5.5. Decision Trees

- **Without PCA**

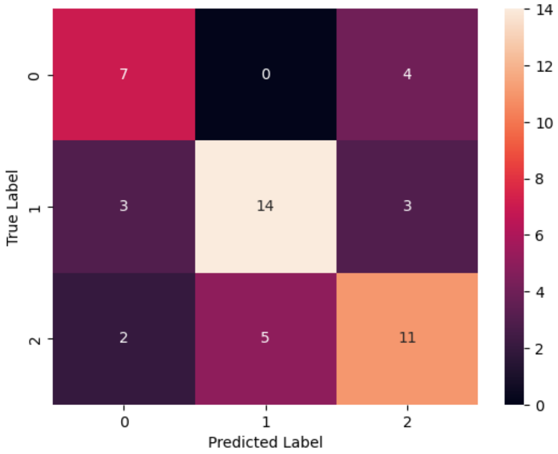


Figure 10. Decision Tree Predictions Vs Truth Values

Best hyperparameters:
’criterion’: ’entropy’
’max_depth’: 20
’$min\_samples\_split$’: 2
Accuracies:
Test accuracy: 0.65
Kappa score: 0.47

- **With PCA**

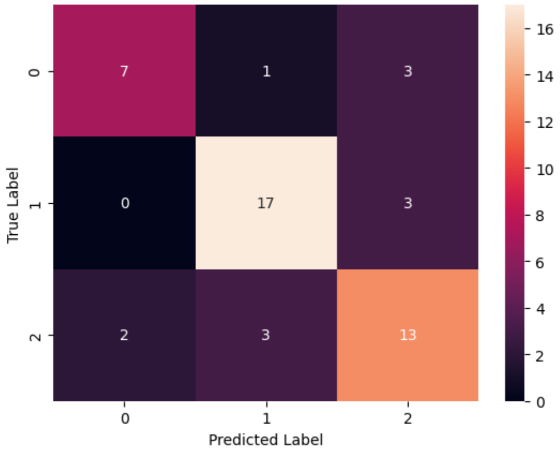


Figure 11. Decision Tree Predictions Vs Truth Values

Best hyperparameters:
’criterion’: ’gini’
’max_depth’: 10 ’$min\_samples\_split$’: 2
Accuracies:
Test accuracy: 0.80
Kappa score: 0.68

5.6. Random Forests

- **Without PCA**

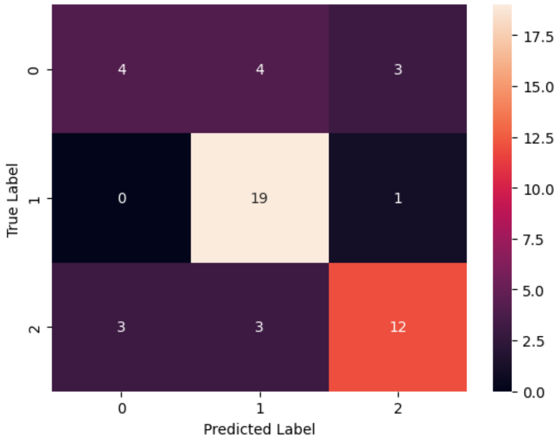


Figure 12. Random Forests Prediction Vs Truth Values

Best hyperparameters:
’max_depth’: 10

’$max\_features$’: ’sqrt’
’$n\_estimators$’: 500
Accuracies:
Test accuracy: 0.71
Kappa score: 0.55

- **With PCA**

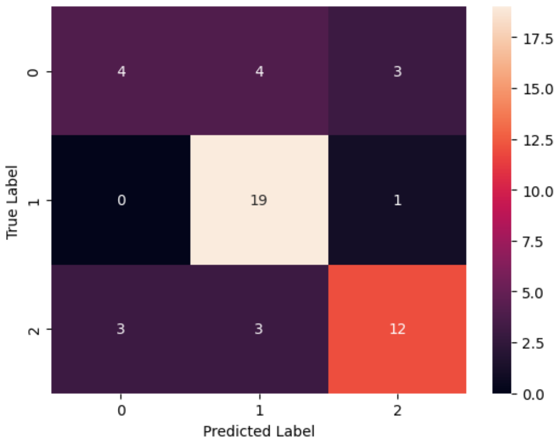


Figure 13. Random Forests Prediction Vs Truth Values

Best hyperparameters:
’max_depth’: 10
’$max\_features$’: ’sqrt’
’$n\_estimators$’: 500
Accuracies:
Test accuracy: 0.71
Kappa score: 0.55

5.7. XgBoost

- **Without Regularization**
  - **Without PCA**

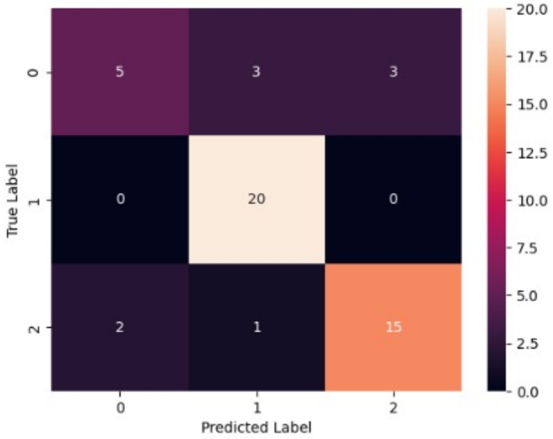


Figure 14. XgBoost Prediction Vs Truth Values

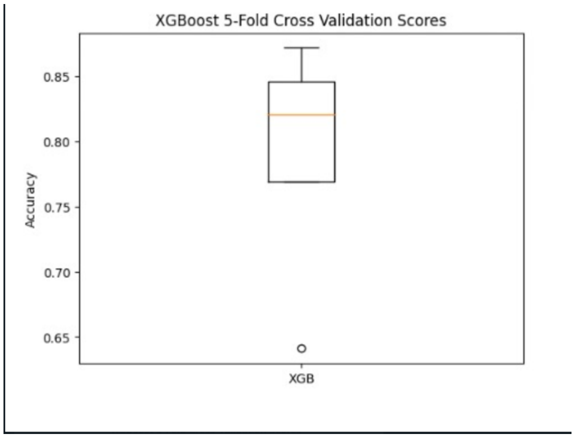


Figure 15. XgBoost Prediction Vs Truth Values

Best Hyperparameters:
’$colsample\_bytree$’: 0.5
’$eval\_metric$’: ’mlogloss’
’$learning\_rate$’: 0.3
’max_depth’: 6
’$n\_estimators$’: 200
’num_class’: 3
’subsample’: 0.7
Accuracies:
Test accuracy: 0.82
Kappa score: 0.71

- **With PCA**

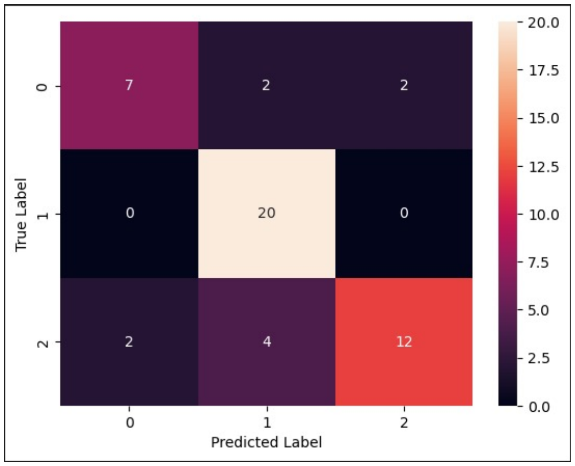


Figure 16. XgBoost Prediction Vs Truth Values

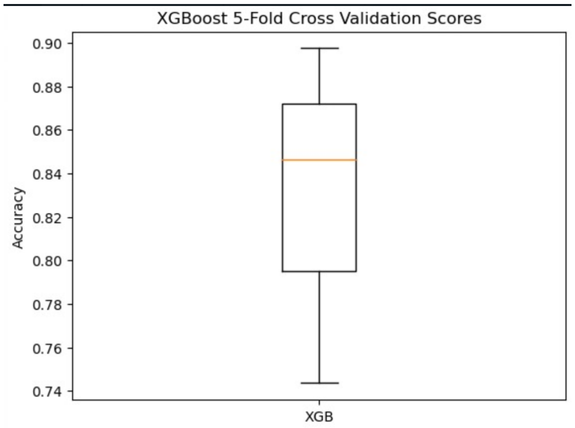


Figure 17. XgBoost Prediction Vs Truth Values

Best Hyperparameters:
’$colsample\_bytree$’: 0.7
’$eval\_metric$’: ’mlogloss’
’$learning\_rate$’: 0.1
’max_depth’: 4
’$n\_estimators$’: 100
’num_class’: 3
’subsample’: 0.7
Accuracies:
Test accuracy: 0.80
Kappa score: 0.68

- **With Regularization**

  – **Without PCA**

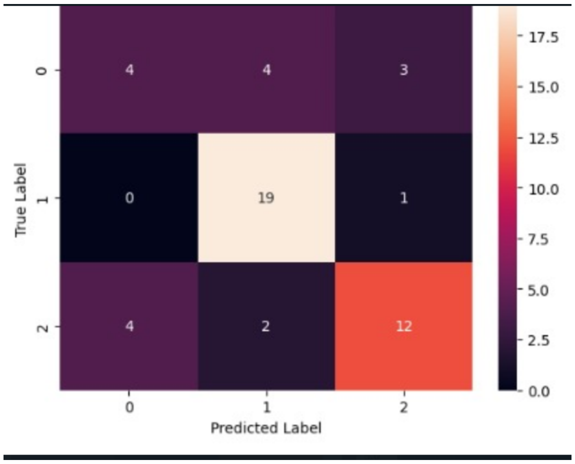


Figure 18. XgBoost Prediction Vs Truth Values

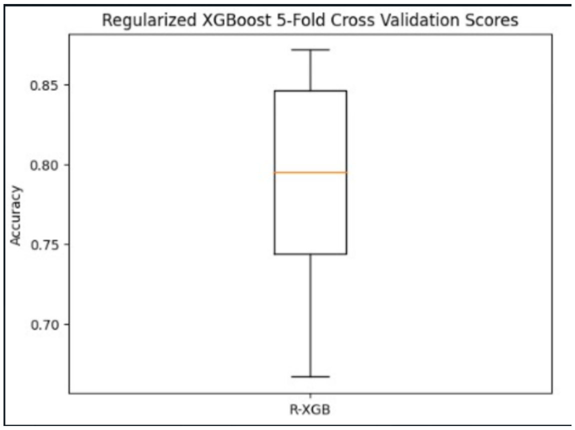


Figure 19. XgBoost Prediction Vs Truth Values

Best Hyperparameters:
’$colsample\_bytree$’: 0.5
’$eval\_metric$’: ’mlogloss’
’$learning\_rate$’: 0.1
’max_depth’: 6
’$n\_estimators$’: 200
’reg_alpha’: 0.5
’reg_lambda’: 0.1
’num_class’: 3
’subsample’: 0.7
Accuracies:
Test accuracy: 0.71
Kappa score: 0.55

– **With PCA**

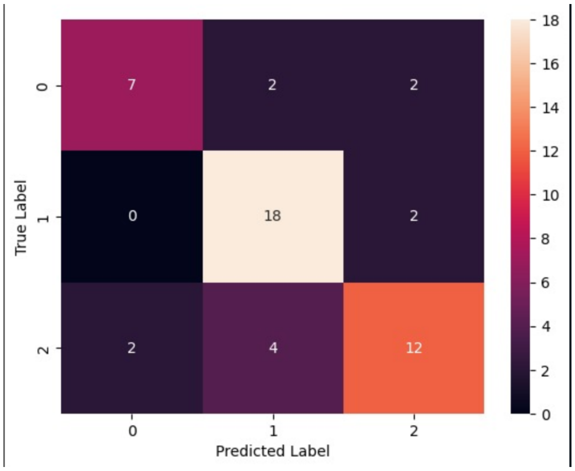


Figure 20. XgBoost Prediction Vs Truth Values

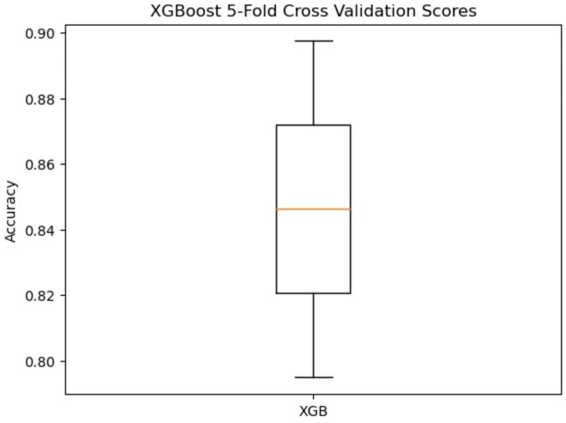


Figure 21. XgBoost Box plot

Best Hyperparameters:
’$colsample\_bytree$’: 0.7
’$eval\_metric$’: ’mlogloss’
’$learning\_rate$’: 0.3
’max_depth’: 6
’$n\_estimators$’: 200
’num_class’: 3
’reg_alpha’: 0.5
’reg_lambda’: 0.1
’subsample’: 0.7
Accuracies:
Accuracy: 0.8615384615384615
Test accuracy: 0.76
Kappa score: 0.62

5.8. CatBoost

- **Without PCA**

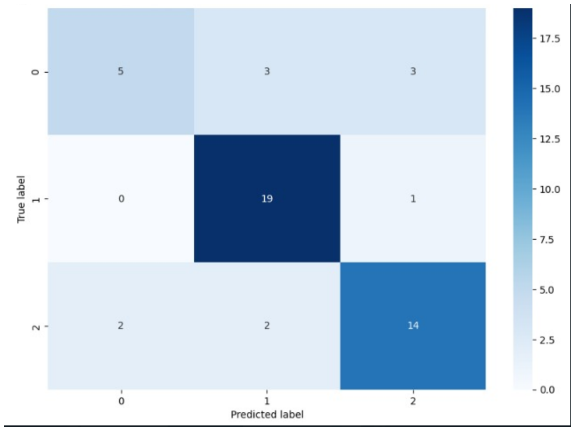


Figure 22. CatBoost Prediction Vs Truth Values

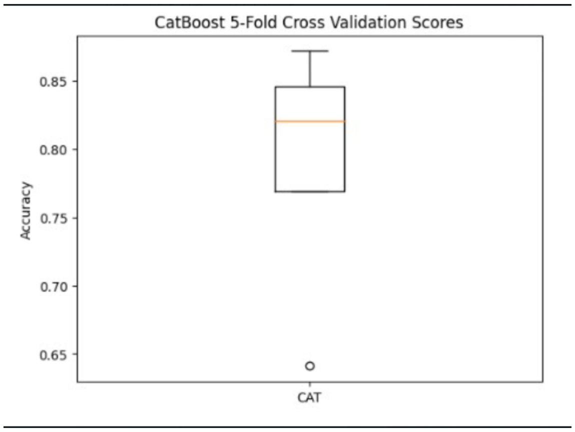


Figure 23. CatBoost Box Plot

Best Hyperparameters:
’depth’: 6
’iterations’: 200
’$l2\_leaf\_red$’: 1
’$learning\_rate$’: 0.1
Accuracies:
Test Accuracy: 0.82
Kappa Score: 0.71

- **With PCA**

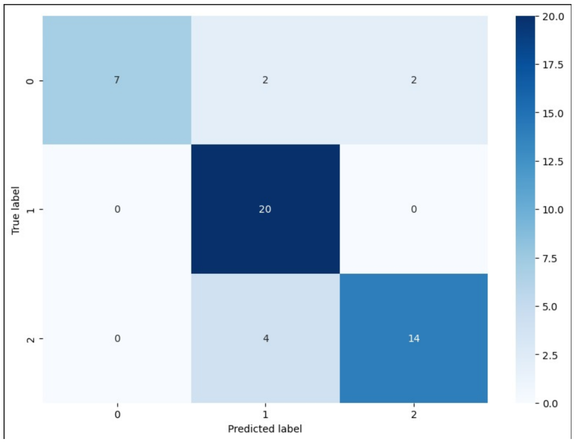


Figure 24. CatBoost Prediction Vs Truth Values

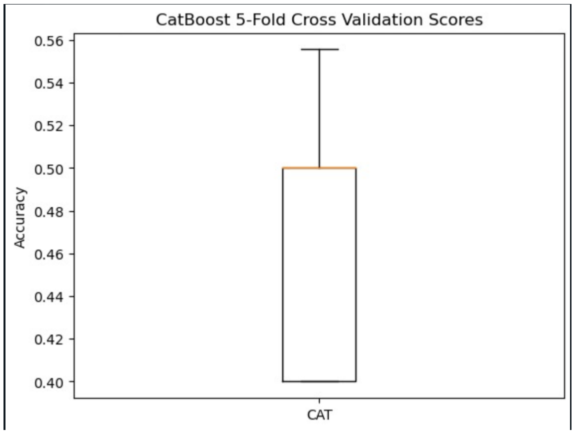


Figure 25. CatBoost Box Plot

Best Hyperparameters:
’depth’: 6
’iterations’: 200
’$l2\_leaf\_red$’: 1
’$learning\_rate$’: 0.1
Accuracies:
Test Accuracy: 0.80
Kappa Score: 0.68

5.9. AdaBoost

- **Without PCA**

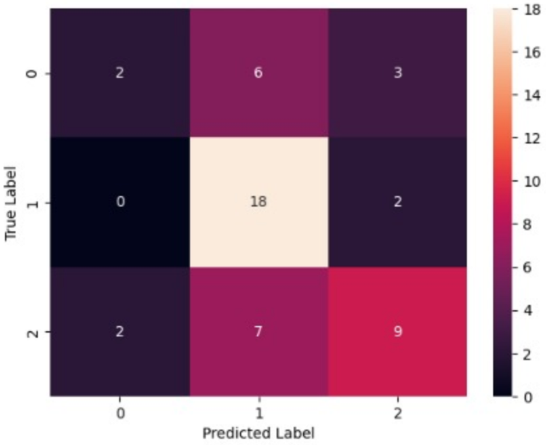


Figure 26. AdaBoost Prediction Vs Truth Values

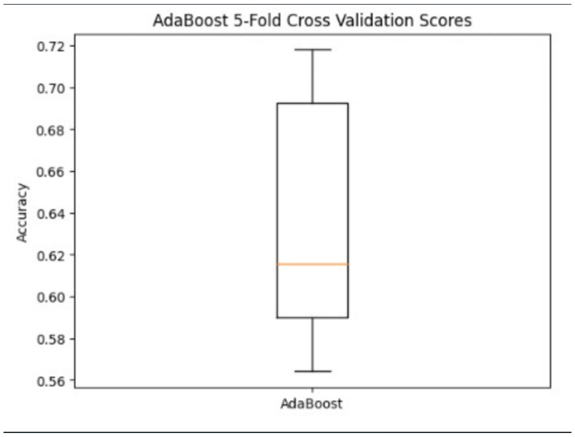


Figure 27. AdaBoost Box Plot

Best Hyperparameters:
’$learning\_rate$’: 1.0
’$n\_estimators$’: 200
Accuracies:
Test Accuracy: 0.59
Kappa Score: 0.34

- **With PCA**

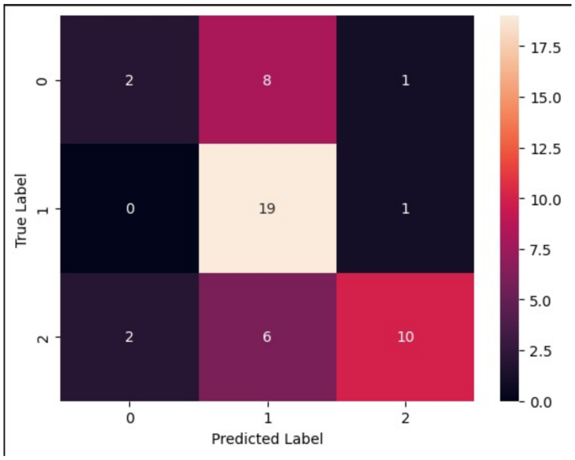


Figure 28. AdaBoost Prediction Vs Truth Values

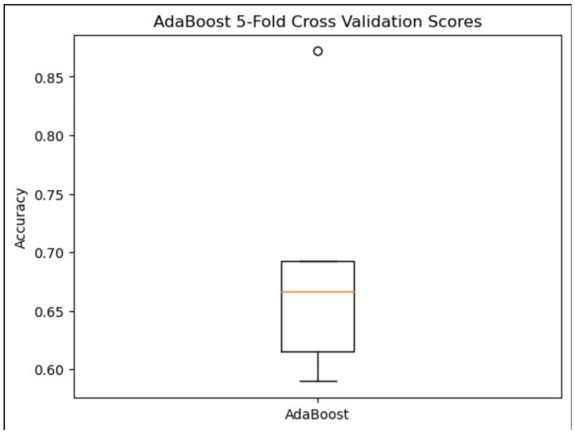


Figure 29. AdaBoost Box Plot

Best Hyperparameters:
$'learning\_rate'$: 0.1
$'n\_estimators'$: 200
Accuracies:
Test Accuracy: 0.63
Kappa Score: 0.40

6. Observations

While implementing simple machine methods we found out that the accuracy was pretty low and there were many cases of overfitting so we tried to search for more complex methods to increase the accuracy and found the ensemble methods like Random forests, XgBoost, CatBoost, and AdaBoost. These methods were getting better accuracies than the previous ones so we continued with these ensemble methods but the problem of overfitting was still remaining. To reduce the overfitting we then tried a few regularization methods like l_1 norm and l_2 norm. But the problem was persistent. Then we implemented Berhu [6] penalty/ regularization term which is a robust of l_1 and l_2 norms.
From the combined Boxplot below we can compare the performances of the models and decide which model works the best in the case of the financial/ market data of the banks in India.

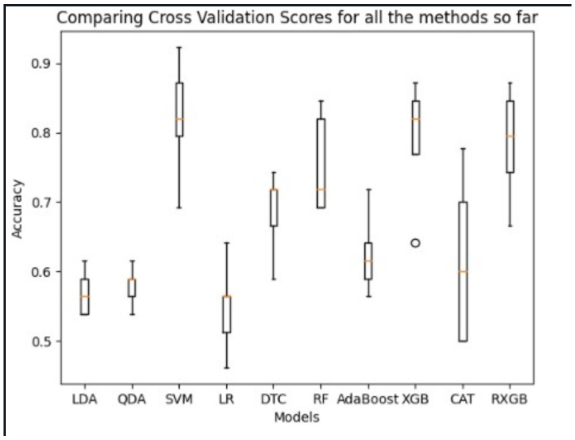


Figure 30. The Combined Box Plot without PCA

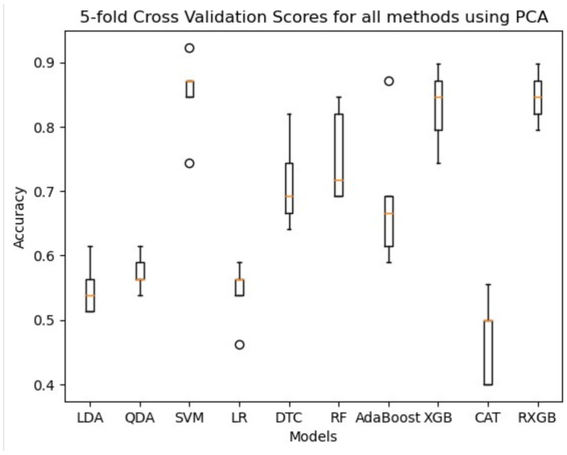


Figure 31. The Combined Box Plot with PCA

From the Box plots, we can infer the following

1. Since the Box plots for the models done using the PCA data are more thinner this can be interpreted as the models from PCA data are more stable in terms of accuracy than the models which use the original data.

2. From the plots we can see that the mean average of each model didn't have much difference between the PCA one and the non-PCA one. This supports the fact that regularization reduces overfitting while risking an increase in accuracy.

7. Conclusion

We have tried using basic machine-learning methods like Logistic regression, Support vector machines, Decision trees, and random forests but the performance is not as good as we wanted it to be. So, we tried to do some research on increasing the accuracy of the models. and we found out about the ensemble-leaning methods on decision trees. Which turned out to have a better performance than the original methods that we implemented. We hope our work will help in the progress of this interesting field.

References

[1] Katarína Kočišová and Mária Mišanková. Discriminant analysis as a tool for forecasting company's financial health. *Procedia - Social and Behavioral Sciences*, 110, 01 2014. 2

[2] João Maroco, Dina Silva, Ana Pina Rodrigues, Manuela Guerreiro, Isabel Santana, and Alexandre Mendonça. Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC research notes*, 4:299, 08 2011. 1

[3] Xuan T.T. Pham and Tin H. Ho. Using boosting algorithms to predict bank failure: An untold story. *International Review of Economics Finance*, 76:40–54, 2021. 3

[4] Soni R. Application of discriminant analysis to diagnose the financial distress, 2019. 2

[5] Yudong Zhang, Shuihua Wang, and Genlin ji. A rule-based model for bankruptcy prediction based on an improved genetic ant colony algorithm. *Mathematical Problems in Engineering*, 2013:1–10, 11 2013. 3

[6] Laurent Zwald and Sophie Lambert-Lacroix. The berhu penalty and the grouped effect, 2012. 7, 15