TAIWAN COMPANY BANKRUPTCY PREDICTION

Farhan Chughtai - Sai Vineeth Kaza - Pratyusha Parashar - Nithya Balachandiran

1) Summary:

Bankruptcy can be considered a curse for the organization and the investors. It is expressed as the inability of a company to pay its debts to its creditors. The bankruptcy of a company and even the possibility of going bankrupt is important for the company's investors and society. Therefore, bankruptcy prediction is a crucial step for each organization before the company goes bankrupt and appropriate models can be built for the development of the organization. Effective bankruptcy prediction is crucial for companies to make appropriate business decisions. In general, the input variables (or features), such as financial ratios, and prediction techniques, such as statistical and machine learning techniques, are the two most important factors affecting the prediction performance. While many related works have proposed novel prediction techniques, very few have analyzed the discriminatory power of the features related to bankruptcy prediction.

The goal of our project is to find the key features that helped to predict bankruptcy. Secondly, visualize interesting patterns present in the data. And finally, predict bankruptcy effectively and find the best-performing model. This will help the organizations to make wise business decisions, invest wisely and reap profits.

After the initial analysis, we found that the dataset has no null or empty values. So, there is no necessity to perform any data cleaning. We performed exploratory data analysis (EDA) on the dataset and try finding some patterns in the dataset, checking for outliers, identifying any potential clusters, etc. As the target label is imbalanced, we employed SMOTE and oversampling techniques to rebalance the distribution for the imbalanced dataset. After EDA, we performed logarithmic data transformation, followed by normalization to take care of the outliers before we feed the data to the models. We performed two test-train splits, an initial Split of 90/10 to get training and test data, and another split of 70/30 was performed on the training data to get the validation data and final training data. We performed randomized search cross-validation with stratified k fold(k=5) on the obtained final training data to get the best parameters for the models. We predicted the bankruptcy by modeling the final training data with Logistic Regression, Random Forest Classifier, Support Vector Classifier, KNN, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), CATBoost, and XGBoost to predict whether the company will go bankrupt or not. Upon considering metrics like precision, recall, and F1-score the Logistic regression and Random Forest Classifier have performed best on the data in this scenario. We also showed the most prominent features of the model by performing feature importance using Random Forest.

2) Exploratory Data Analysis:

The dataset is about the company’s financial data from the Taiwan economic journal for the years 1999 to 2009, which has listed the details of the company's bankruptcy based on the business regulations of the Taiwan Stock Exchange. It has over 900 listed companies. The dataset has 6819 rows and 93 numerical variables, 2 categorical variables, and the target variable is Bankrupt?

The Exploratory Data analysis helps us in analyzing to find out different underlying patterns and summarize how each feature behaves. After performing EDA, we observed the following plots as shown below:

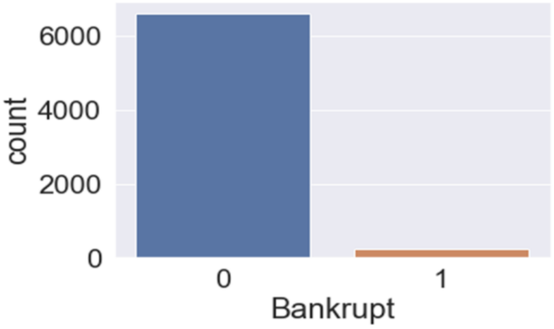


Figure 1 shows that our dataset is imbalanced, as our target variable consists of more '0's (company failed to go bankrupt) compared to '1' (company got bankrupted). We perform SMOTE and Over-sampling techniques to handle the imbalanced data.

Fig 1. Count plot of the Target Variable

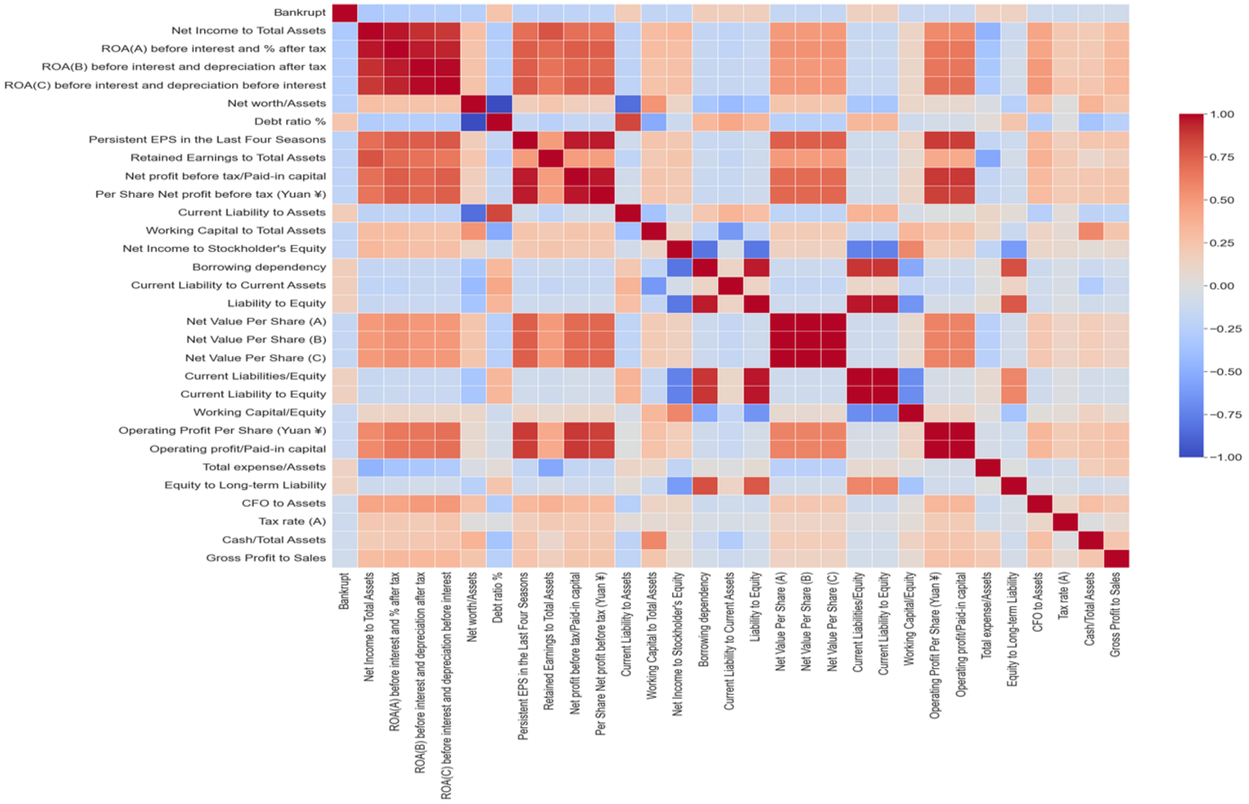


Fig 2. Correlation plot of the top 30 predictor variables

Figure 2 shows the correlation plot for the top 20 features with how they correlate with the target variable. We can observe from the correlation plot that 'Debt Ratio $', 'Current Liability to Assets' and 'Borrowing Dependency' are the top 3 features with the strongest positive correlation, whereas 'Net Income to Total Assets', 'Net worth/Assets' and

Persistent EPS in the Last Four Seasons' are the top 3 strongest negative correlated variables with the target variable.

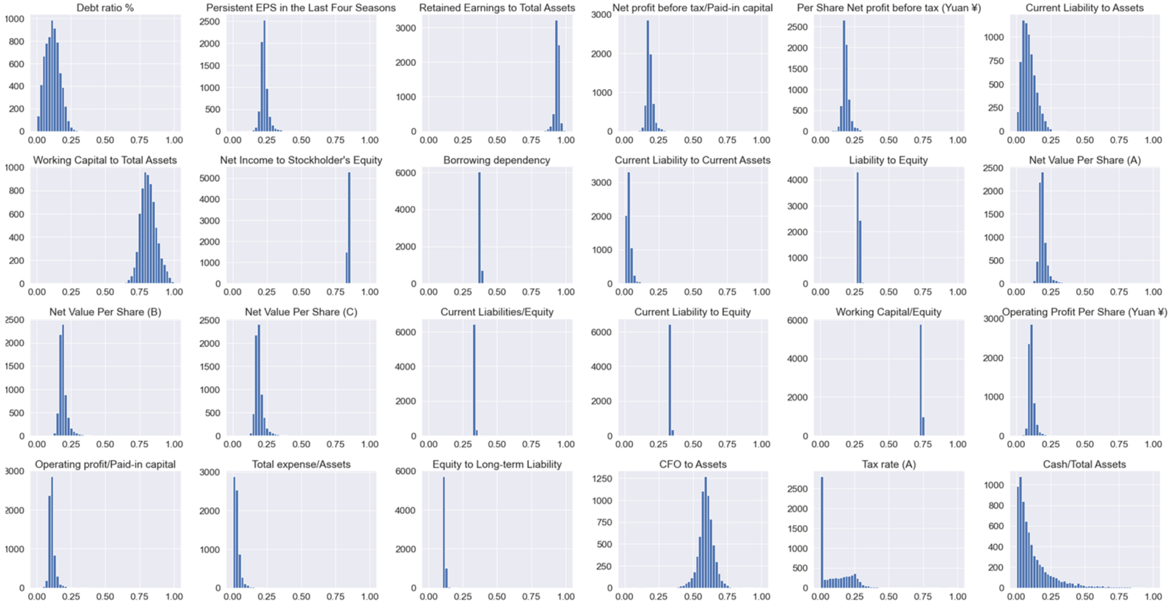


Fig. 3. Distribution of some of the 95 independent variables.

Figure 3 shows the distributions of a few of the 95 independent variables of the data.

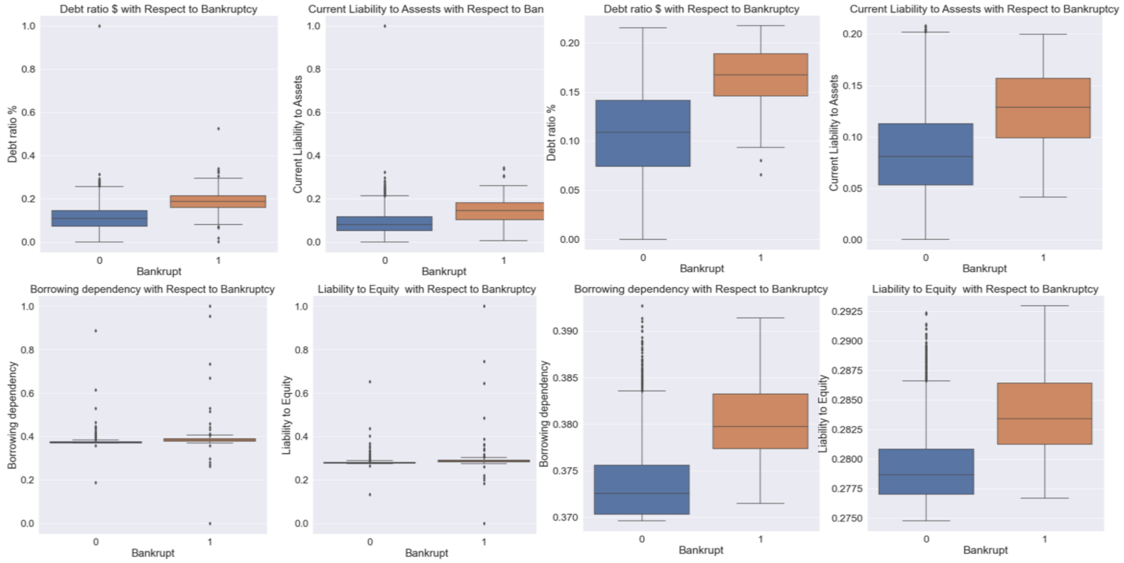


Fig 4. Distribution of the Top 4 positively correlated variables with and without Outliers.

Figure 4,5 displays the distribution of the Top 4 positively and negatively correlated variables with the target variable respectively, with and without the outliers of those variables respectively.

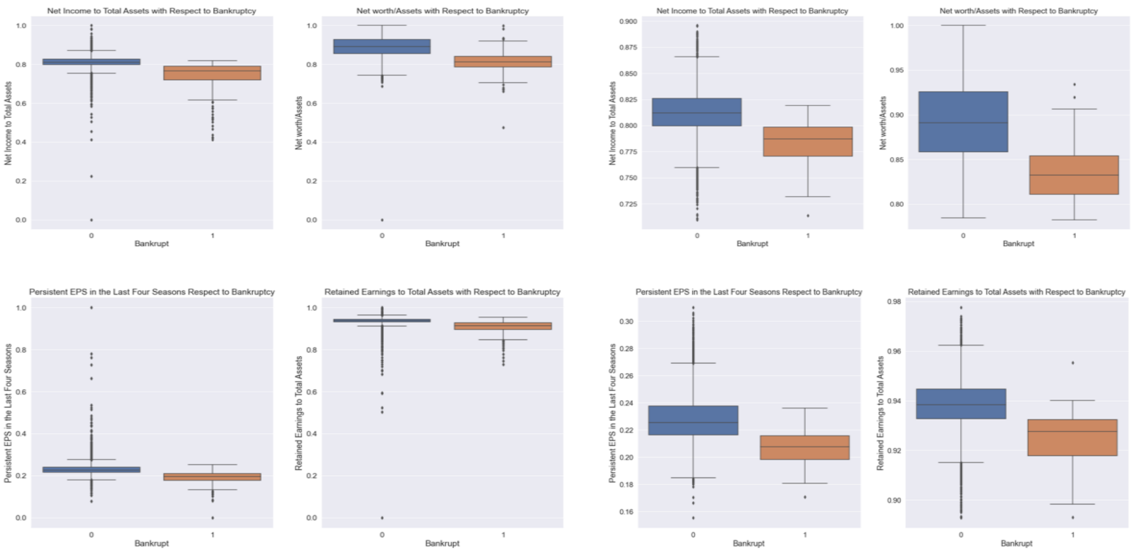


Fig 5. Distribution of the Top 4 negatively correlated variables with and without Outliers.

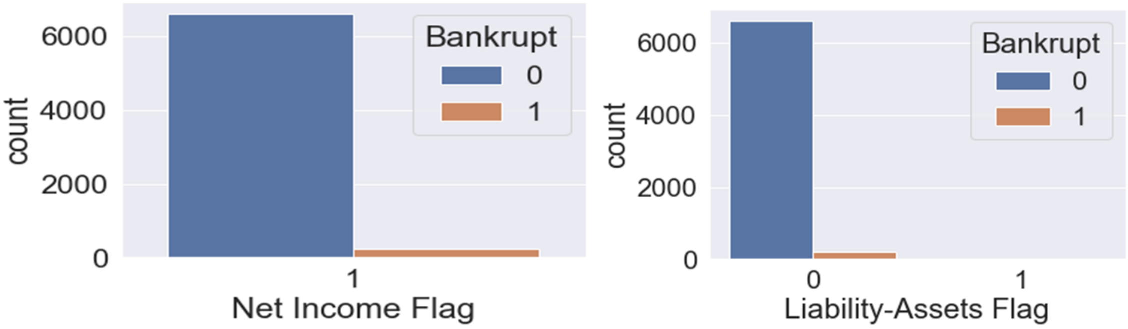


Fig 6. Count plot of the 2 categorical variables.

Figure 6 shows the distribution of the 2 categorical variables ‘Net Income Flag’ and ‘Liability-Assets Flag’. We can observe from the plot that the ‘Net Income Flag’ variable consists of only ‘1’ for all the observations, which makes it not useful for bankruptcy prediction. Out of all the companies with the ‘Liability-Assets Flag’ variable with Flag 1, 80% of the companies have gone bankrupt. This makes this variable useful for the bankruptcy prediction. Out of all the 95 independent variables in the dataset, we removed the variables which have less than 0.1 correlation with the target variable and those that have greater than 0.8 correlation among the independent variables to eliminate the redundant and highly correlated variables. This process resulted in 14 independent variables to feed into the model for the bankruptcy prediction.

3) Feature Selection and Modeling Methods

3.1 Data Transformation

Based on the EDA performed above, we applied log transformation on the 14 variables that we ended up selecting in the end. To standardize all the features to the same scale we then applied Standard Scaler to our final dataset. The aim of stacking them together is to standardize all the features following the feature generation process. We also applied techniques such as SMOTE (Synthetic Minority Oversampling Technique) and Oversampling to deal with the class imbalance problem in our dataset.

3.2 Feature Selection

We started with training our machine learning models on the entire dataset with 95 independent variables. We achieved a Recall score of 80% when we used all the columns to predict our dependent variable. In order to remove some noise from the dataset, we then removed features that had a very low correlation (<0.1) with the dependent variable. We also removed features that were highly correlated with each other (>0.8). We compared the absolute correlation values [since range of correlation is (-1,1)] with 0.1 and 0.8, We ended up with 14 features in our dataset after applying those filters. We chose this as our final dataset. We choose to report the metrics observed after applying oversampling on the dataset since this method gave us better results over SMOTE.

3.3 Train and Test dataset Preparation

We split the dataset into train and test based on a 90:10 split at random. We further split our training data into training and validation datasets on a 70:30 split at random.

3.4 Classification Models

We developed a Machine Learning Function to practice code reusability where one would pass in variables like ModelName, the training dataset, the actual model, the parameters to be used for Grid Search, and the value of K in K-fold cross-validation.

3.4.1 Logistic Regression

It is a classification model that's simple to implement and delivers excellent results with linearly separable classes. The logistic regression model does not classify data; instead, it models the likelihood of output in terms of input. It can, however, be used to build a classifier by simple cutoff-based rules. We found that the best performance for the model was observed with our final dataset (14 features) with a Recall value of 0.91 whereas we observed a recall of 0.8 when we used the entire dataset.

3.4.2 K-Nearest Neighbors

K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points that are closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of ‘K’ training data and which class holds the highest probability will be selected. We observed recall being 0.43 which was way low than what we observed for logistic regression.

3.4.3 Random Forest

Random Forest makes use of ensemble learning, which is a technique that combines many weaker classifiers to solve complex problems. It is made up of a large number of decision trees. The random forest algorithm's 'forest' is trained using bagging or bootstrap aggregation.

The outcome is determined by the algorithm based on the predictions of the decision trees. It predicts by averaging the output of various trees. We found recall to be 0.54 on our final dataset.

3.4.4 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is used as a tool for classification, dimension reduction, and data visualization. It has been around for quite some time now. Despite its simplicity, LDA often produces robust, decent, and interpretable classification results. When tackling real-world classification problems, LDA is often the benchmarking method before other more complicated and flexible ones are employed. We found that LDA gives us a recall score of 0.91 when bankruptcy= 1 or True on our final dataset.

3.4.5 Quadratic Discriminant Analysis (QDA)

An extension of linear discriminant analysis is quadratic discriminant analysis, often referred to as QDA. This method is similar to LDA and also assumes that the observations from each class are normally distributed, but it does not assume that each class shares the same covariance matrix. Instead, QDA assumes that each class has its own covariance matrix. We observed a recall score of 0.77 on the validation dataset for this model.

3.4.6 XG Boost

The XGBoost ensemble model is a gradient boosting ensemble model based on decision trees. Gradient boosting attempts to predict a target variable by combining the estimates of several simpler and weaker models. Some of the features of XGBoost are as follows: Parallel tree structure, pruning trees with a depth-first approach, and regularization is done to avoid overfitting. Best hyperparameters found by grid search were lambda=1, learning_rate=0.100000001, max_depth=9, n_estimators=100 and gamma=0. We found Recall to be 0.43 on the validation dataset

3.4.7 CatBoost

CatBoost is an algorithm for gradient boosting on decision trees. It is developed by Yandex researchers and engineers and is used for search, recommendation systems, personal assistants, self-driving cars, weather prediction, and many other tasks at Yandex and in other companies, including CERN, Cloudflare, and Careem taxi. It is open-source and can be used by anyone. It is similar to XGBoost, but it is over 3 times faster when training the model. We observed Recall on the validation dataset to be 0.93 which was better than XGBoost

3.4.8 Support Vector Classifier

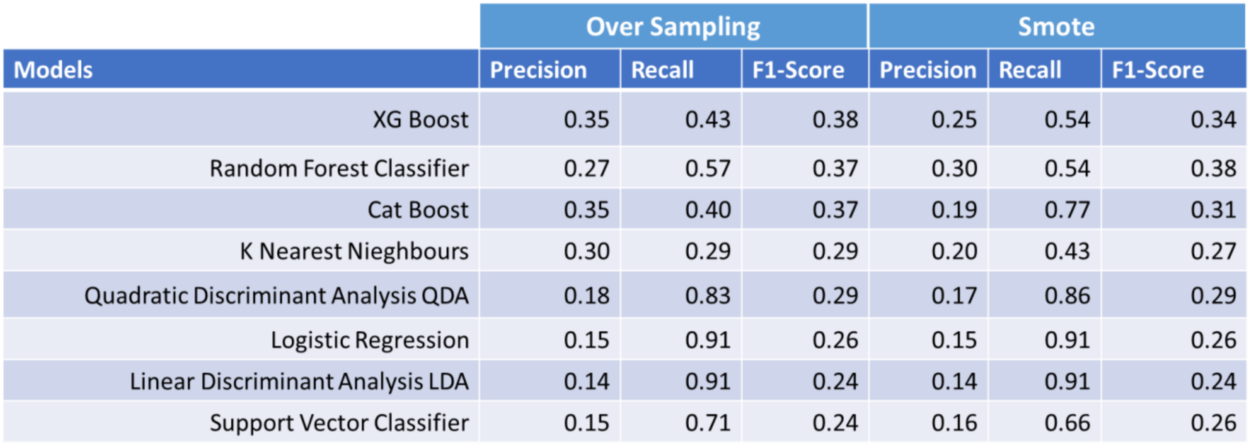
Support vector machines (SVMs) are powerful, yet flexible supervised machine learning methods used for classification, regression, and outliers' detection. SVMs are very efficient in high-dimensional spaces and generally are used in classification problems. SVMs are popular and memory-efficient because they use a subset of training points in the decision function. The main goal of SVMs is to divide the datasets into a number of classes in order to find a maximum marginal hyperplane (MMH) which can be done in the following two steps:

- Support Vector Machines will first generate hyperplanes iteratively that separate the classes in the best way.

- After that, it will choose the hyperplane that segregates the classes correctly. We observed a recall score of 0.66 on the validation dataset with 14 features.

4) Results:

4.1 Comparative Analysis



	Over Sampling			Smote		
Models	Precision	Recall	F1-Score	Precision	Recall	F1-Score
XG Boost	0.35	0.43	0.38	0.25	0.54	0.34
Random Forest Classifier	0.27	0.57	0.37	0.30	0.54	0.38
Cat Boost	0.35	0.40	0.37	0.19	0.77	0.31
K Nearest Nieghbours	0.30	0.29	0.29	0.20	0.43	0.27
Quadratic Discriminant Analysis QDA	0.18	0.83	0.29	0.17	0.86	0.29
Logistic Regression	0.15	0.91	0.26	0.15	0.91	0.26
Linear Discriminant Analysis LDA	0.14	0.91	0.24	0.14	0.91	0.24
Support Vector Classifier	0.15	0.71	0.24	0.16	0.66	0.26

Fig 7. Metric Scores of the models for Over Sampling and SMOTE

From figure 7, we can observe that we achieved the best results on the validation data using Logistic Regression. While the worst performing model was K-nearest neighbors. Logistic regression despite being a simple model performed the best and it is the fastest of all the eight models we employed for classification. It's often observed that with real-world datasets, simple models such as Logistic regression often overperform complex deep learning models and gradient boosting models such as XGBoost and CatBoost. We have observed this in our dataset.

4.2 Feature Importance

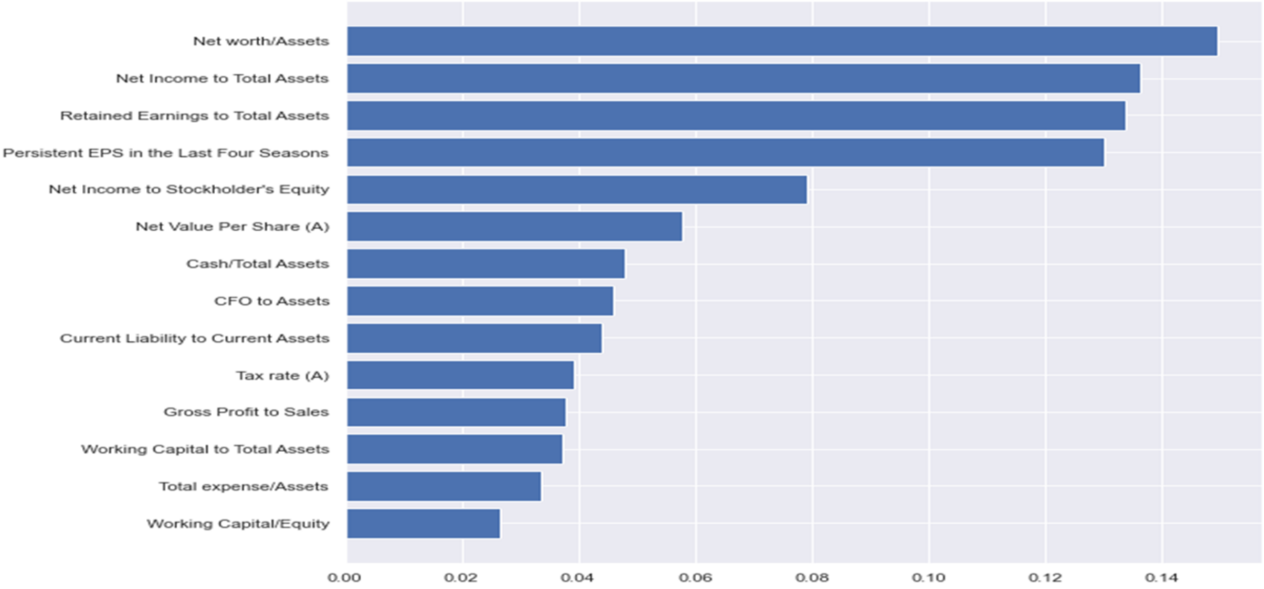


Fig 8. Important Features from Random Forest Model

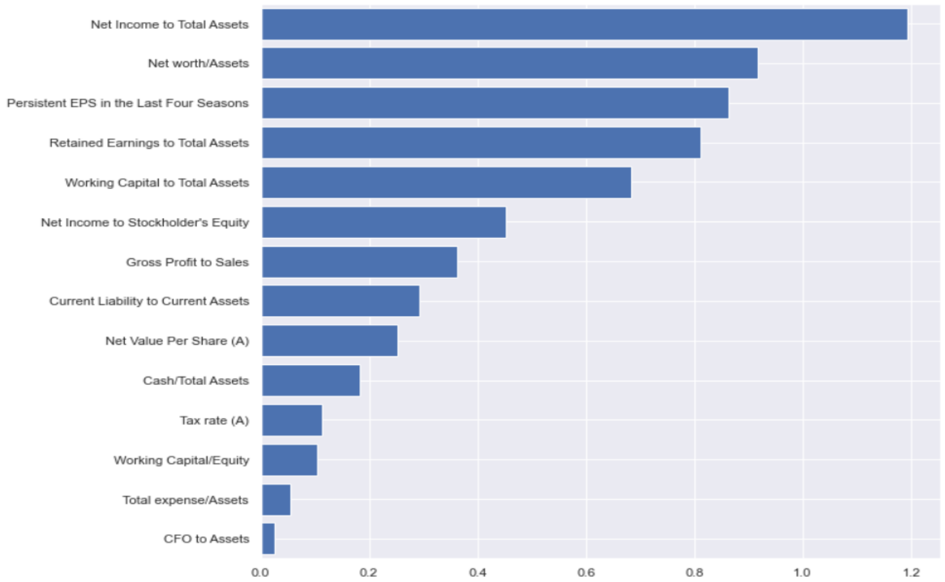


Fig 9. Important features from Logistic Regression Model

Figure 9 shows the absolute coefficient values of the logistic regression model. We can observe that both figures 8 and 9 have similar features that assisted the model in predicting our target variable efficiently.

5) Discussion:

Results demonstrate which are the most defining factors to play an important role in issuing credit to the unbanked population who are taken advantage of by most of the money lenders. This Analysis and Modeling can help us predict potential bankruptcies better.

6) Future Enhancements:

1. Try ensemble algorithms which are a mix of logistic regression and LDA for the data to further improve the performance.
2. Try more models like deep neural networks for tabular data to see if we can get a better performance.

7) Statement of Contributions:

- **Farhan Chughtai**: Performed Data Preprocessing, Exploratory Data Analysis, Implemented Logistic Regression Algorithm, and Support Vector Classifiers. Worked on Presentation and Report.

- **Sai Vineeth Kaza**: Performed Feature Selection and Engineering, Implemented XGBoost and CatBoost Algorithm. Worked on Presentation and Report.

- **Pratyusha Parashar**: Performed Data Preprocessing, Exploratory Data Analysis, and implemented LDA and QDA. Worked on Presentation and Report.

- **Nithya Balachandiran**: Performed Feature Selection, Implemented Random Forest Classifier. Worked on Presentation and Report.

8) References:

[1] Dataset Source, https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction

[2] Exploratory Data Analysis, https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

[3] Logistic Regression, https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4] XGBoost Algorithm, https://towardsdatascience.com/xgboost-python-example-42777d01001e

[7] Random Forest Classifier, https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956

[8] Metrics for Model Evaluation, https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b
https://medium.com/analytics-vidhya/precision-recall-tradeoff-for-real-world-use-cases-c6de4fabbcd0

[9] Applying Standard Scaler to Dataset, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[10] GridSearchCV, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[11] Applying both Log Scale and Standard Scaler to the dataset, https://stats.stackexchange.com/questions/483187/difference-between-log-transformation-and-standardization

[12] Support Vector Classifier, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[13] CatBoost Classifier, https://catboost.ai/en/docs/concepts/python-usages-examples

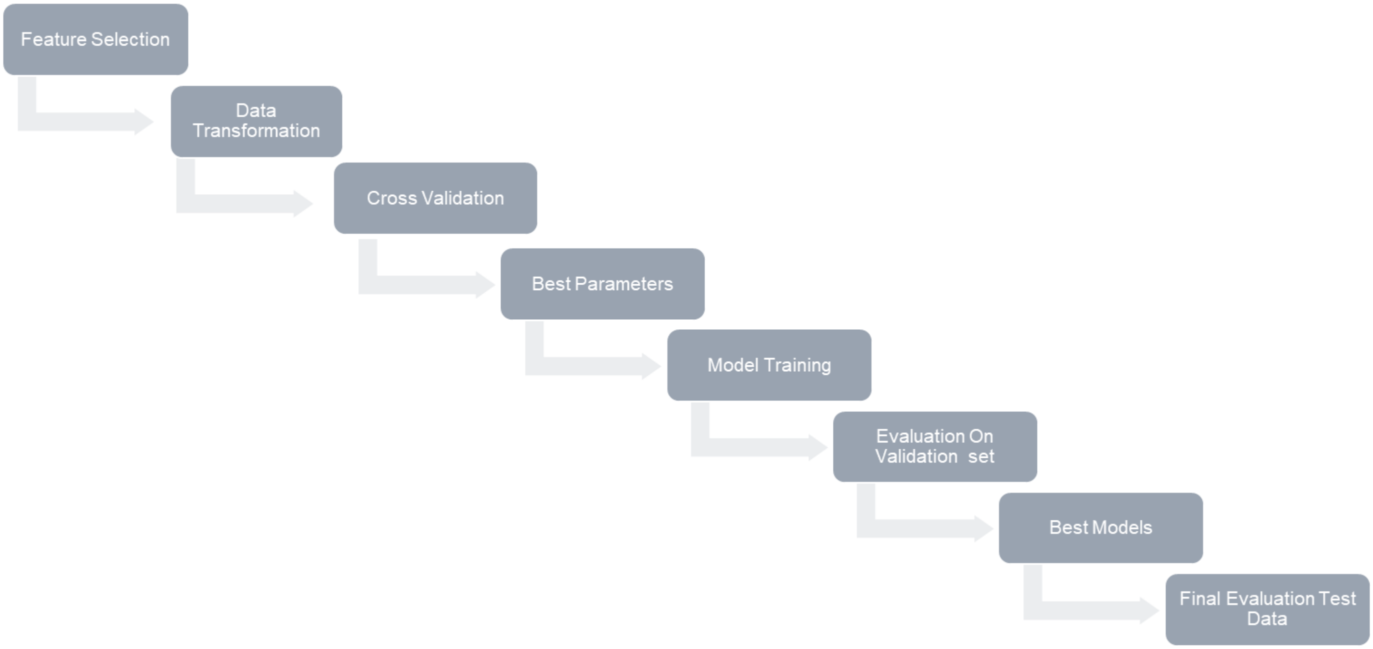
[14] Linear Discriminant Analysis and Quadratic Discriminant Analysis, https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/

[15] Jupyter Notebook, https://drive.google.com/file/d/10J-ki9U9AwMiKUJQHfPdlzrWKsln9MpY/view?usp=sharing

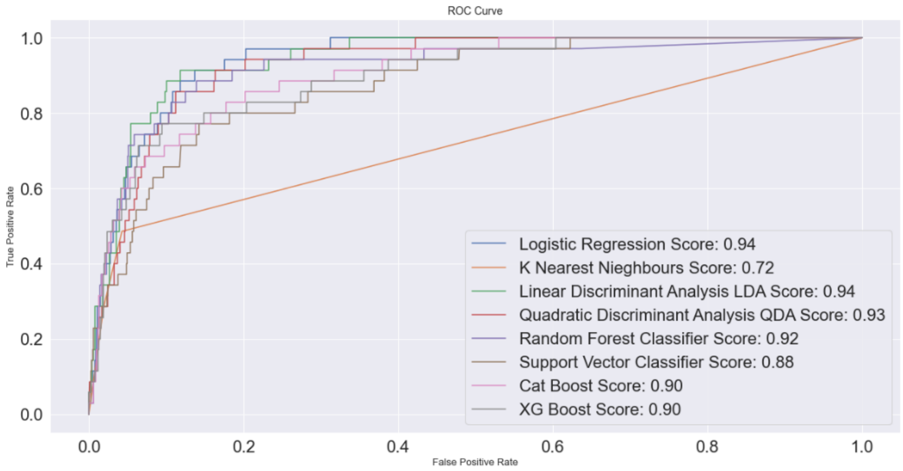
[16] Kaggle Notebook that we referenced for some EDA, https://www.kaggle.com/code/ginelledsouza/bankruptcy-analysis

9) Appendix:

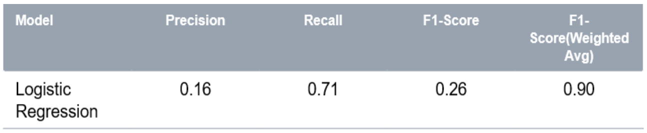
[1] Below is the Machine Learning Process flow:



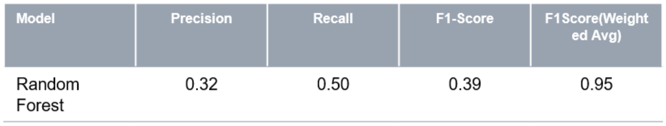
[2] Below is the ROC Curve:



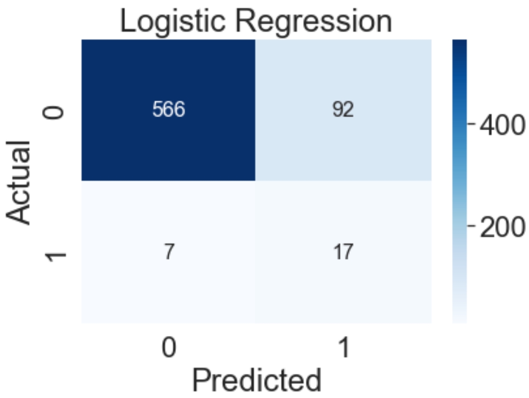
[3] Final Results on Test Data

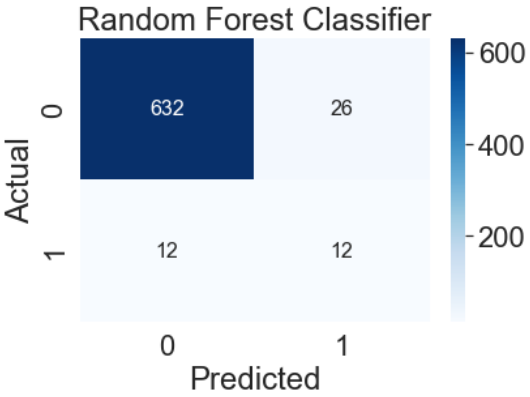


Model	Precision	Recall	F1-Score	F1-Score(Weighted Avg)
Logistic Regression	0.16	0.71	0.26	0.90



Model	Precision	Recall	F1-Score	F1Score(Weighted Avg)
Random Forest	0.32	0.50	0.39	0.95


Logistic Regression
Actual
Predicted


Random Forest Classifier
Actual
Predicted