



Course Code : AIT302  
Course Name : Statistical Learning  
Lecturer : Ms. Shamini A/P Raja Kumaran  
Academic Session : 2023/04  
Assessment Title : Final Project  
Submission Due Date : 15 June 2023

Prepared by :	Student ID	Student Name
	AIT2104245	Lim Yi Jing
	AIT2104240	Chiam Yu Wei
	EEE2004045	Cheng Hung Xu

Date Received : \_\_\_\_\_

Feedback from Lecturer:	Mark:
-------------------------	-------

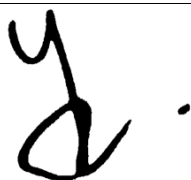

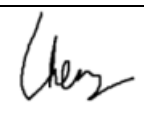
## Own Work Declaration

I/We hereby understand my/our work would be checked for plagiarism or other misconduct, and the softcopy would be saved for future comparison(s).

I/We hereby confirm that all the references or sources of citations have been correctly listed or presented and I/we clearly understand the serious consequence caused by any intentional or unintentional misconduct.

This work is not made on any work of other students (past or present), and it has not been submitted to any other courses or institutions before.

Signature:

Name	Signature
Lim Yi Jing	
Chiam Yu Wei	
Cheng Hung Xu	

Date: 5/6/2023

## Work Segregation

No.	Topic	Sub-topic	Contributors
1.	Introduction	-	Cheng Hung Xu
2.	Literature Review	Journal Review	Lim Yi Jing Chiam Yu Wei Cheng Hung Xu
		Summary from all journals	Chiam Yu Wei
3.	Methodology	Descriptive Analysis	Cheng Hung Xu
		Outliers Removal	Cheng Hung Xu
		Inferential Analysis	Chiam Yu Wei
		Feature Selection	Chiam Yu Wei
		Factor Analysis and Dimensionality Reduction	Cheng Hung Xu
		Data Augmentation	Lim Yi Jing
		Evaluation Metrics	Lim Yi Jing
		Model Training	Lim Yi Jing Chiam Yu Wei Cheng Hung Xu
4.	Classification Result and Analysis	Results of each model	Lim Yi Jing Chiam Yu Wei Cheng Hung Xu
		Model Comparison	Lim Yi Jing
5.	Proposed Solution	-	Lim Yi Jing
6.	Conclusion and Future Works	-	Lim Yi Jing
7.	Code Implementation	Descriptive Analysis	Cheng Hung Xu
		Data Cleansing	Lim Yi Jing
		Inferential Analysis	Chiam Yu Wei
		Feature Selection	Chiam Yu Wei
		Factor Analysis and Dimensionality Reduction	Lim Yi Jing
		Model Implementation and Evaluation	Lim Yi Jing Chiam Yu Wei Cheng Hung Xu
		Model Comparison	Lim Yi Jing
		Proposed Solution	Lim Yi Jing

## **Table of Contents**

### **1.0 Introduction**

### **2.0 Literature review**

1. Literature Review
2. Overall Conclusion for Literature Review

### **3.0 Experiment Methodology and Analysis**

#### **3.0.1 Data Preprocessing**

- i. Descriptive Analysis
- ii. Outliers Removal and Data Cleansing
- iii. Inferential Analysis
- iv. Feature Selection
- v. Exploratory Factor Analysis and Dimensionality Reduction
- vi. Data Augmentation

#### **3.0.2 Model training**

- i. Evaluation Metrics
- ii. XGBoost Classifier
- iii. KNN Classifier
- iv. Ridge Regression
- v. Decision Tree Classifier
- vi. SVC
- vii. MLP
- viii. Logistic Regression

### **4.0 Classification Result and Analysis**

1. Results and Analysis of each model
2. Models Comparison

### **5.0 Proposed Solution**

### **6.0 Future Works and Conclusion**

### **7.0 References**

## **Title: Company Bankruptcy Classification**

### **1.0 Introduction**

What does bankruptcy indicate in a company's perspective? It is a legal action the business undergoes on the occasion the business is incapable in repaying its debtors and fulfilling its obligations. Bankruptcy is a pebble thrown into a pool, the waves generated are the reverberation of the situation. Its employees, shareholders, and the economy will have to endure the aftermath caused by the bankruptcy of the company whereas the scale is determined by the size and position of the company in its industry. With better understanding on factors and indicators of company bankruptcy, prediction with high precision can be done on the likelihood of bankruptcy benefits investors, lawmakers and officials.

Statistical learning provides in depth evaluation on bankruptcy data and procures the variables which has a strong correlation with company bankruptcy risk. The process involves thorough inspection on return-on-assets, operation gross profit margin etc. By implementing statistical methods such as regression trees, support vector machines, logistic regression, accurate prediction model can be generated as the data is analysed properly.

In this project, we will emphasize on company bankruptcy and statistical models suitable in analysing company bankruptcy data. By reviewing journals from other researchers and combining with our findings, we will identify the variables commonly used in recognising the bankruptcy risk. Besides that, we will create several classification models to predict likelihood of a company facing bankruptcy. We will also discuss on limitation of our models in company bankruptcy prediction. In general, this project is to analyse the dataset, implement statistical models and make classification models to predict the risk of company bankruptcy.

### **2.0 Literature Review**

There are several techniques proposed by various researchers used in financial field such as predicting company bankruptcy.

In the paper of “Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches” by Mu-Yen Chen. The author has done a lot of effort in data pre-processing such as variable selection. First, they standardize the data into range of [0,1] and ran a factor analysis to test whether the differences between each variable were significant to reduce data dimensionality. Its goal is to minimize the complexity of the components by increasing large loadings and decreasing small loadings within each component.

Next, the paper of “Statistical Methods for Bankruptcy Forecasting” by Rober A. Collins and Richard D. Green examines and evaluates the assumptions and properties of various statistical models, including Multiple Discriminant Analysis (MDA), Linear Probability Model (LPM), Quadratic Discriminant Analysis (QDA), Ordinary Least Squares (OLS), and Logistic Regression (logit). Based on the results, LOGIT model has the best accuracy among all the models. This is because LOGIT model has the ability to handle non-linear relationships between variables and account for interactions between them. The LOGIT model also provides a probabilistic interpretation of its results, which can be useful in decision-making process.

The following journal reviewed by us is “Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey” by Boyacioglu et al. We will focus on data preprocessing method from this journal.

They had carried out normalization and variable reduction to improve the results. By comparing the means of both failed and non-failed banks with the independent sample t-test, the ratio which shows less difference were excluded from the dataset. This action may enable the failed and non-failed banks to be differentiated easier (Boyacioglu et al., 2009). As we know that uncorrelated variables are needed for discriminant analysis, the authors used Pearson Correlation coefficients to check the existence of high multicollinearity among variables. While the answer was positive, they applied factor analysis to get the covariance relationships in term of factors extracted. They performed Bartlett’s Test of Sphericity and KMO to ensure the factor analysis method was appropriate. Besides, PCA was used by the authors in the experiment. After standardization, two bases took the role of choosing several factors.

The first one is that the cumulative variance should exceed 70%, followed by the factors having eigenvalues greater than 1 (Boyacioglu et al., 2009). Based on their findings, CA and LRA seemed to perform well among statistical methods. More data pre-processing and methods of choosing the better factors could be explored to further improve their performance.

In ‘Data Analytic approach for bankruptcy prediction’, the researchers managed to group the input variables into 7 categories which are, cash flow ratios, growth ratios, leverage ratios, liquidity ratios, operational ratios, profitability ratios, solvency ratios, and the others. The researchers added zero values for missing data and winsorize the dataset which limits extreme values which helps to reduce the effect of outliers.

Feature importance in the dataset is obtained through the prediction models.

XGBoost, lightBGM and random forest concluded that variable f193035 which is the cash ratio, a liquidity measure of a company ability to repay its short-term debt using only cash and cash equivalent is the most important variable in terms of predicting bankruptcy risk. With the combination of feature importance and logistic regression model, the top ten important features and its respective weights could be obtained. Positive weight in the logistic regression represents that the risk of the company going bankrupt increases as the specific feature with positive weight increases, whereas for features with negative weight increases indicates the risk of bankruptcy decreases.

## **Summary of Literature Review**

We have done a simple summary regarding the related work literature review we studied.

First let us dive into data analysis part. There are various alternatives for data analysis and excellent models and can be apply to the company bankruptcy prediction problem. Statistics analysis such as inferential statistics and factor analysis are important for us to understand the data, and prepare, and pre-process the data before fitting it into machine learning and statistical learning model. Factor Analysis enable us to perform dimensionality reduction and feature selection on our dataset, by transforming observed variable to latent variable. Moreover, a balanced dataset between healthy and bankruptcy company is also a key to good classification result.

Next, let us summarize the models used by existing works. Most of the models are statistical and machine learning approaches. The statistical methods reviewed by us include LDA, MDA, QDA, LOGIT, and more. There are examples of machine learning techniques such as Neural Network, Support Vector Machine, Decision Tree and others in solving the bankruptcy prediction problem. In general, we found that machine learning approaches often getting better performance compared to statistical techniques. Hence, it could be a wise choice to apply distinct models and compare their performance to obtain the model that best predicts our dataset.

### **3.0 Experiment Methodology and Analysis**

In this section, we will explain the experiment we did on the dataset. Also, we will analyse the experimental result in this section (except model classification result, this will be explained in the next section).

#### **1. Data Preprocessing**

##### **1.1 Descriptive Analysis**

Descriptive analysis is the manipulation of data for it to be in an easier understandable manner. We checked the data for the existence of null values and obtained important data from each variable for instance, mean, median, mode etc. Besides that, we also checked for data type for each variable and found out only 'Bankrupt ?', 'Liability – Assets Flag' and 'net income flag' are categorical variables whereas others are continuous variables. We also made use of histogram plotting to assist us in observing the relationship between variables.

The 'Net Income Flag' variable is removed due to it only has a unique value and does not contribute to the prediction process. The imbalance of data is also observed, it will affect the predictions of the data, more work will be done to prevent this from happening. Outliers are observed through boxplot and the removal of them will be done in the next step. We can observe that cash/total asset has a high left skewed distribution of data, whereas 'net income to total asset', 'net worth/assets' and 'retain earnings to total asset' has a high right skewed distribution through density plot.



Moreover, we obtain the highest 5 positive and negative correlations with the target variable 'Bankrupt ?', heatmap and scatterplot are plotted to understand the relationships between variables through visualization.

## **1.2 Outlier Removal**

From the boxplot visualization, we can see that there are a lot of outliers in this dataset. Thus, we decided to remove outliers from our dataset.

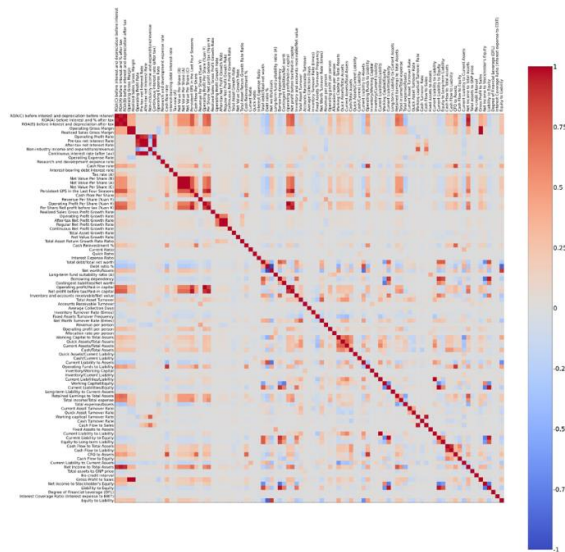
Outliers are extremely high or low data points relative to the rest of the datapoints within the dataset. This step is crucial as the existence of these outliers will cause the prediction model to be inaccurate and skew the result of any hypothesis tests. We use the Interquartile range method (IQR) to deal with the outliers for each variable. We are interested in the middle group of the population thus data within 25 and 75 percentile of the population is captured. The number of instances reduced from 6819 to 6270 after the removal of outliers. The remaining instances are pre-processed using MixMaxScaler to preserve the distribution of the data.

## **1.3 Inferential Analysis**

Inferential analysis is the method of describing or determining some information regarding the population of interest in term of sample. The term population means the whole set of instances including objects and measurement obtained from the instances of interest while the sample carries the meaning of a part of the population. We separated the binary and numeric variables for the purpose of performing different inferential analysis on them.

We first performed a correlation test on those 93 numeric variables to find out the relationship between two different variables. We used the standard Pearson correlation to analyze our variables as most of our variables were quantitative. Besides, we had removed outliers in the previous section hence making Pearson correlation coefficient suited our data. It gave us the result between the range of -1 and 1 indicating the correlation strength among the two variables. Negative coefficient value implies that the variable has negative correlation with another. When one variable changes, the other one will also change but in a different direction. The inverse of this can be applied to positive coefficient value such that one variable will

change in the same direction along with another variable. Zero value of coefficient means no correlation between the two variables.



The result shows the Pearson correlation coefficient for all numeric variables.

We can observe that the squares in red indicated a positive correlation while the squares in blue indicated a negative correlation. The gray squares indicated no correlation among variables.

	Attribute	Bankrupt	Non-bankrupt	t-statistics	p-value
0	ROA(C) before interest and depreciation before...	0.418793	0.508188	-16.103327	0.0
1	ROA(A) before interest and % after tax	0.457165	0.56205	-14.262213	0.0
2	ROA(B) before interest and depreciation after...	0.46183	0.556748	-15.130888	0.0
3	Operating Gross Margin	0.292464	0.308261	-8.970137	0.0
4	Realized Sales Gross Margin	0.292547	0.30822	-8.907045	0.0
...	...	...	...	...	...
89	Net Income to Stockholder's Equity	0.825909	0.840887	-2.905851	0.004042
90	Liability to Equity	0.293748	0.280328	2.88059	0.004367
91	Degree of Financial Leverage (DFL)	0.028458	0.027575	0.775857	0.438622
92	Interest Coverage Ratio (Interest expense to ...	0.564958	0.565388	-0.975886	0.329877
93	Equity to Liability	0.21467	0.39818	-29.108408	0.0

The next inferential analysis we did was an independent t-test for all 94 variables. We first divided the data into two classes of “bankrupt” and “non-bankrupt” as we would be comparing the sample t-test with the mean of these two classes. Before applying for the independent sample t-test, we tried to get the variance of each variable for the two different classes. Then, the variance ratio for a variable was calculated by taking the higher variance divided by the lower variance of the classes. The threshold value for variance ratio was set to 4. After checking, we got 41 variables which getting variance ratio higher than 4 (almost half of the attributes), hence we are going to assume that the variance of these two classes is not equal in the following t-test. After getting the results from t-test, we found that there was a total of 59 variables which had p-value lower than 0.05. Hence, we had enough evidence to reject the null hypotheses of the two classes having equal population means for these variables.

One sample proportion t-test was performed on the binary variable. It is a statistical test to compare the sample proportion and population proportion. Its null hypothesis assumes that the population proportion is equal to a specific hypothesized value. We got z-statistic value of -1106.2674 telling us that the sample proportion was far away from the null hypothesis proportion in the negative direction. The p-value of 0.0 indicates that probability of observing a sample proportion as extreme as the one we have calculated by assuming the null hypothesis is true, is essentially zero.

We can reject the null hypothesis with a high level of confidence and further making conclusion such that the sample proportion is statistically significantly different from the null hypothesis proportion. In simple words, we can be confident that the observed difference between the sample proportion and the null hypothesis proportion is not due to chance and that the sample proportion is a true reflection of the population proportion.

#### **1.4 Feature Selection**

Feature selection was implemented to reduce the number of variables and computational complexity of our models later. It is often useful in preventing overfitting of the model. We chose Lasso Regression (L1) for the regularization on the training data in a linear regression model. The regularization penalty could shrink the coefficients of less important features to zero, producing a sparse feature subset. Hyperparameter tuning on L1 was performed to get the best value of alpha for feature selection. The evaluation method used in the model was ROC AUC score. One of the reasons behind it was its robustness to imbalance dataset as we had.

The best alpha for our dataset is 0.0001 which got AUC ROC score of over 0.9149, hence it will be used in the following feature selection. After performing feature selection with L1, we got 33 features as our remaining feature. The dataset with these selected features was stored as a reduced dataset for the model training.

#### **1.5 Exploratory Factor Analysis and Dimensionality Reduction**

We had performed KMO test to measure how suited our data is for factor analysis. A KMO model score of 0.683 suggests moderate sampling adequacy. There is some

degree of shared variance among the feature variables, they are not completely independent of each other. However, the amount of shared variance is not very high. Although the KMO score is not in the ideal range of 0.8 and above, we can still proceed with factor analysis for dimensionality reduction.

Factor analysis is used to reduce the dimensionality of the dataset with large number of variables into small number of factors. Through exploratory on the hyperparameters of Factor Analyzer, we obtained that with no rotation on the data, using principal factor analysis, which assumes there are no correlation between factors provides the best variance explained. The factor axes are left in their original orientation, and the factors are uncorrelated when no rotation is applied; while principal factor analysis transformed our observed variables into smaller set of uncorrelated factors that account for the maximum amount of variance in the data, it assumes that the factors are orthogonal to each other, meaning they are uncorrelated. Based on these 2 hyperparameters obtained and the factor analysis result, we have drawn a few conclusions about our dataset:

- i. Possible uncorrelated factors. The factors extracted from the data are independent of each other and do not exhibit any significant pattern of interrelationships.
- ii. Maximum variance explained. The extracted factors are likely to capture the major sources of variability in the data. This implies that the identified factors represent the most significant dimensions underlying the observed variables.
- iii. Lack of interpretability. The factors could be less interpretable as no rotation is applied. It is difficult to give the variables meaningful labels or interpretations when their orientation is unchanged by rotation. Thus, rather than obtaining easily interpretable factors, the focus is mostly on dimensionality reduction and capturing the underlying variance.

Overall, the best hyperparameters we obtained, principal factor analysis without rotation indicates that the primary goal of us on this dataset is to identify uncorrelated factors that account for the maximum variance in the dataset. The resulting factor structure may lack interpretability but provides a basis for further exploration and analysis of the underlying patterns in the data.

After obtaining the factor loadings, Kaiser-Guttman Rule is applied which preserves factors with eigenvalues larger than 1, which means that we will keep 11 factors. These 11 factors explain more variance than any individual variable and is therefore worth keeping. Furthermore, we can identify 11 factors are worth remaining that has eigenvalues larger than 1 through scree plot.

33 features are assigned to their respective factors based on their factor loadings. Features with factor loadings lower than threshold are assigned factor 0 which does not exhibit any strong correlation with any specific factor. Next, we can observe that factor 1 has 'ROA(A) before interest and % after tax', 'Operating profit/Paid-in capital', 'Cash/Total Assets', 'Operating Funds to Liability' and 'Net Income to Total Assets' variables categorised under it. In addition, 'ROA(A) before interest and % after tax' and 'Net Income to Total Assets' variables have factor loadings higher than 0.8 within factor 1 which has the highest eigenvalue with the data. Through further research, these two variables are the most important metric to gauge the profitability of a company which is essential for our study. Overall, we interpreted that factor 1 represents a financial performance factor that encompasses variables related to profitability, asset utilization, liquidity, and financial stability. Based on our findings, these variables are all related to financial performance and profitability measures of the organizations or entities in the data.

Whereas 'Total Asset Turnover' and 'Net Worth Turnover Rate (times)' are categorised in factor 2. This suggests that there is a relationship between the efficient utilization of assets and the rate at which net worth is turned over. Companies that effectively generate revenue from their assets also tend to experience a higher rate of growth or turnover in their net worth, which is logical in the business world. Next, for factor 4 has 'Long-term fund suitability ratio (A)' and 'Fixed Assets to Assets' variables categorised under it. This factor can be explained as a measure of long-term financial stability and asset utilization. A high factor loading for these two variables indicates that a company with a higher long-term fund suitability ratio also tends to have a higher proportion of fixed assets relative to their assets.

Through dimensionality reduction based on factor analysis, we have reduced our dataset size from 33 features to 11 latent variables.

## **1.6 Data Augmentation**

During our experiment on the model training, we found out that imbalanced data will lead to poor model learning results due to several reasons such as bias towards majority class, poor generalization, insufficient learning and sensitivity to noise. Thus, we decided to perform data augmentation on our dataset.

The data augmentation techniques we applied is adaptive synthetic (ADASYN) sampling approach. The essential idea of ADASYN is to use a weighted distribution of various minority class instances according to their level of difficulty in learning, where more synthetic data is generated for minority class instances that are harder to learn compared to those minority examples that are easier to learn.

## **2. Model Training**

We have trained 7 models in our experiments in order to determine which model performed the best. The 7 models are XGBoost Classifier, KNN Classifier, Ridge Regression, Decision Tree Classifier, Support Vector Classifier, Multilayer Perceptron and Logistic Regression.

In model training, we first train our model using dataset that without data augmentation, but this gave us bad result, thus we performed data augmentation on the dataset (as mentioned above) and fit into the same models again, and we compared both model results using two different datasets.

Also, since we are dealing with imbalanced data, we will apply an oversampling method called SMOTE (Synthetic Minority Over-Sampling Technique) to the training dataset when training the models.

We also applied Stratified KFold Cross Validation Sampling on our models' training processes. This sampling techniques ensures that the class distribution in each fold is representative of the dataset, addressing the potential issue of having some folds with a significantly lower representation of certain classes.

### **2.1 Evaluation Metrics**

In our model training experiment, we will use several evaluation metrics to evaluate our model performance on our dataset.

No.	Evaluation Metrics	Explanation
1	Accuracy	Accuracy measures the overall correctness of the classification model. It gives a general idea of how well the model is performing but may not be suitable when the classes are imbalanced.
2	Precision	Precision quantifies the ability of the model to correctly predict positive instances among all instances predicted as positive.
3	Recall	Recall measures the ability of the model to correctly identify positive instances among all actual positive instances.
4	F1 Score	The F1 score is a combination of precision and recall, providing a single metric that balances both measures.
5	AUC-ROC Score	AUC-ROC provides a measure of the overall discriminatory power of the model, regardless of the chosen classification threshold.
6	Confusion Matrix	A confusion matrix is a table that summarizes the performance of a classification model by comparing predicted and true class labels. It provides a detailed breakdown of true positives, true negatives, false positives, and false negatives.

## 2.2 XGBoost Classifier (eXtreme Gradient Boosting)

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. XGBoost builds an ensemble of weak prediction models (typically decision trees) and combines them to create a strong predictive model. The final prediction is obtained by aggregating the predictions from all the individual models. We had performed cross-validation random search hyperparameter tuning while training the XGBoost classifier in order to search for the best combination of hyperparameters. The hyperparameters here included learning rate, n

estimators, evaluation metric, max depth, lambda for L2 regularization, and alpha for L1 regularization.

### **2.3 K-Nearest Neighbors (KNN Classifier)**

KNN algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. For classification using KNN, a class label is assigned on the basis of the majority vote. We had performed cross-validation and evaluation on KNN classifier during training phase. The KNN classifier is initialized without specifying any hyperparameters. This will be tuned using random search cross-validation. The hyperparameters to be tuned are number of neighbors, weighting scheme, and power parameter for the Minkowski distance metric. These hyperparameters have been optimized during cross-validation.

### **2.4 Ridge Regression**

Ridge Regression is chosen as one of our models as it is a regression method used in machine learning to lower the problem of multicollinearity and overfitting. The L2 penalty term is added to the least squares regression equation to constrain the size of the coefficients. Due to our data having multiple variables and consisting of imbalanced dataset, ridge regression model is chosen to solve this problem. The hyperparameters that require tuning are alpha values and types of solvers used for this regression. Although this model outputs continuous values, we round up the output to obtain binary values. We performed random search to obtain the best hyperparameters for the model.

### **2.5 Decision Tree Classifier**

Decision tree classifier is a supervised learning technique. It is a classification model which uses leaves to represent class labels and branches as features. This model has its perks as we can understand which features are used to classify the data to its class. The hyperparameter which requires tuning is criterion, splitter and max depth. We performed a random search to obtain the best hyperparameters for the model.

### **2.6 Support Vector Classifier (SVC)**



Support Vector Classification (SVC) was chosen as one of our models since it was originally designed for binary classification as our problem. The difference between this supervised machine learning algorithm and Support Vector Machine (SVM) is that it classifies data linearly instead of using a non-linear approach. It works by mapping datapoints into high-dimensional space to carry out linear separation easily, following by finding an optimal hyperplane that separates two classes. The term optimal means it aims to maximize the margin between the hyperplane and the nearest data points in high-dimensional space as well as minimizes the expected generalization error. We decided to choose Radial Basis Function (RBF) as the kernel function due to a few reasons. Firstly, it could capture non-linear relationship of our data as shown before, Besides, RBF kernel shows flexibility by adapting well to variety of data distributions due to the existence of hyperparameter “gamma” which known as kernel coefficient.

We performed grid search hyperparameter tuning on regularization parameter and gamma for a few values to obtain the best hyperparameter values to train the SVC model using reduced dataset.

## **2.7 Multi-layer Perceptron (MLP)**

Multi-layer Perceptron classifier was another model implemented on our dataset. It is a feedforward neural network that produces output based on the input taken. It has one input layer, one or more hidden layers and one output layer. This supervised neural network uses backpropagation in the training phase to update the weights.

As usual, we performed hyperparameters tuning on the type of optimizers, strength of L2 regularization term and learning rate for weight updates would be performed to get the best hyperparameters in MLP for our dataset. The activation function was set to be Rectified Linear Unit (ReLU) in the MLP model. As the piecewise linear function of ReLU could be essential in training to capture the non-linear relationship among the variables in our dataset.

## **2.8 Logistic Regression (LR)**

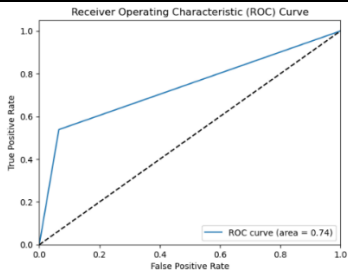
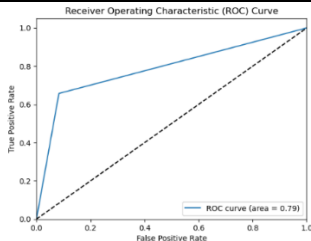
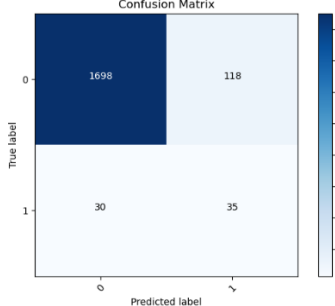
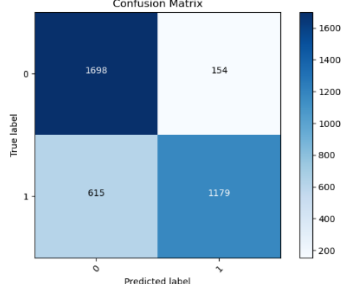
Logistic Regression (LR) was the following model used. It is a machine learning algorithm that studies the relationship between categorical dependent variables and

multiple independent variables. It uses sigmoid which maps prediction into probabilities as the cost function.

In the hyperparameter tuning part, we trained the model using different norm for penalty and “C”, the inverse of regularization strength. The penalties involved were L1, L2 and elasticnet. “Saga” was chosen as the solver in the model training. The reason behind it was to improve generalization by getting better estimated gradients to update the model.

## 4.0 Classification Results and Analysis

### 1. XGBoost

No.	Reduced Dataset	Augmented Transformed Dataset
1	Test Accuracy: 0.9213184476342371 Test Precision: 0.22875816993464052 Test Recall: 0.5384615384615384 Test F1-score: 0.3211009174311927 Test AUC: 0.7367417824466281	Test Accuracy: 0.7890839275918815 Test Precision: 0.8844711177794449 Test Recall: 0.657190635451505 Test F1-score: 0.754077390470099 Test AUC: 0.7870186438596618
2	 <p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>ROC curve (area = 0.74)</p>	 <p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>ROC curve (area = 0.79)</p>
3	 <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <p>0 1</p> <p>0 1</p> <p>1698 118 30 35</p>	 <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p> <p>0 1</p> <p>0 1</p> <p>1698 154 615 1179</p>

The table above shown the performance results of XGBoost on both reduced dataset and augmented transformed dataset. We can see that there is a significant

difference performance result between both of them. At first, XGBoost performed badly on the reduced dataset. The model has relatively high accuracy but relatively low precision, recall and f1-score. Precision rate of 0.229 indicates that when the model predicts an instance as positive, it is only correct about 22.9% of the time; recall rate of 0.538 indicates that the model captures only around 53.8% of actual positive instances. Overall, the model struggles to correctly identify positive instances, resulting in low precision and recall.

Thus, we came out with a hypothesis that this is caused by the imbalanced data, Thus, we performed data augmentation and fitted the augmented data into the model again. And, the result has improved significantly, with a f1 score of 0.7540. The precision is relatively high, indicating that when the model predicts an instance as positive, it is usually correct; however, the recall is moderate, indicating that the model captures a significant portion but not all of the actual positive instances. Overall, XGBoost on augmented transformed dataset has improved significantly compared to reduced dataset.

Therefore, we came out with a conclusion that XGBoost cannot deal with imbalanced data well, we need to have a balanced data in order to use XGBoost well.

Here are some reasons why XGBoost does not inherently handle imbalanced data based on our study:

1. Biased Gradient Updates

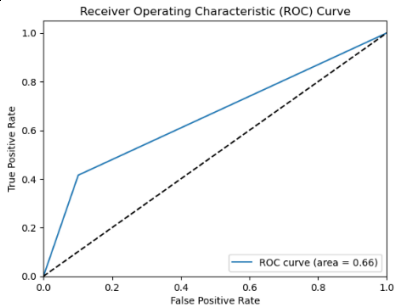
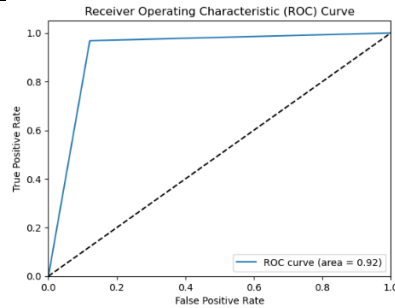
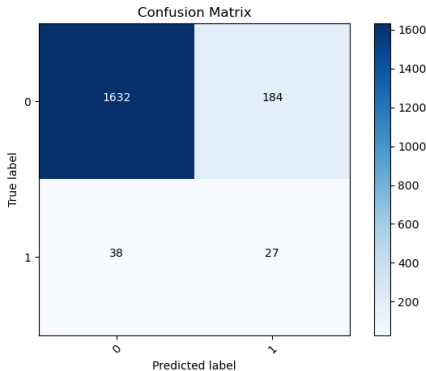
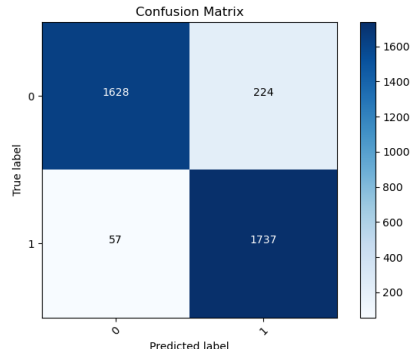
XGBoost is a gradient boosting algorithm that optimizes a differentiable loss function by iteratively adding weak learners to the ensemble. In the case of imbalanced data, the gradient updates can be biased towards the majority class, as it focuses more on reducing the overall loss without properly addressing the imbalance.

2. Skewed Decision Threshold

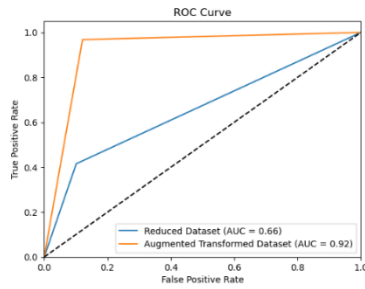
In balanced data, the threshold used to classify instances is typically set as 0.5. However, in the case of imbalanced data, this threshold may not be appropriate.

A skewed decision threshold can cause the model to favor the majority class, leading to biased classifications.

## 2. KNN

No.	Reduced Dataset	Augmented Transformed Dataset
1	Test Accuracy: 0.8819776714513556 Test Precision: 0.12796208530805686 Test Recall: 0.4153846153846154 Test F1-score: 0.1956521739130435 Test AUC: 0.6570315147407658	Test Accuracy: 0.9229292375205704 Test Precision: 0.8857725650178481 Test Recall: 0.9682274247491639 Test F1-score: 0.9251664447403463 Test AUC: 0.923638550387541
2	 <p>Receiver Operating Characteristic (ROC) Curve for the Reduced Dataset. The plot shows True Positive Rate vs False Positive Rate. The ROC curve (blue line) is significantly below the diagonal (dashed line), indicating poor performance. The area under the curve is 0.66.</p>	 <p>Receiver Operating Characteristic (ROC) Curve for the Augmented Transformed Dataset. The plot shows True Positive Rate vs False Positive Rate. The ROC curve (blue line) is very close to the top-left corner, indicating excellent performance. The area under the curve is 0.92.</p>
3	 <p>Confusion Matrix for the Reduced Dataset. The matrix shows True label (0, 1) vs Predicted label (0, 1). The counts are: True Positive (1632), True Negative (184), False Positive (38), and False Negative (27). The color scale ranges from 0 to 1600.</p>	 <p>Confusion Matrix for the Augmented Transformed Dataset. The matrix shows True label (0, 1) vs Predicted label (0, 1). The counts are: True Positive (1628), True Negative (224), False Positive (57), and False Negative (1737). The color scale ranges from 0 to 1600.</p>

Similar to XGBoost, KNN classifier performed way better on augmented transformed dataset than reduced dataset. The KNN performed very well on the augmented dataset. The accuracy rate of 0.92 suggests that the model has a high overall classification rate, and f1 score of 0.92 indicates that when the models predicts an instances as positive, it is usually correct. Overall, KNN performed well in correctly classifying both positive and negative instances, with a high accuracy and f1-score on test dataset.



The difference of performance results between reduced and augmented dataset can further be visualized using the roc-curve on the side.

The reason why KNN does not inherently handle imbalanced data is because of majority class dominance and distance metric sensitivity.

Thus, we further proved that data augmentation is important in our case, where our dataset is highly imbalanced.

### 3. Ridge Regression

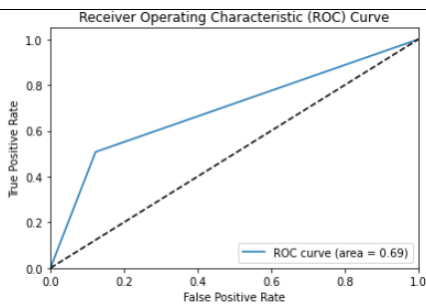
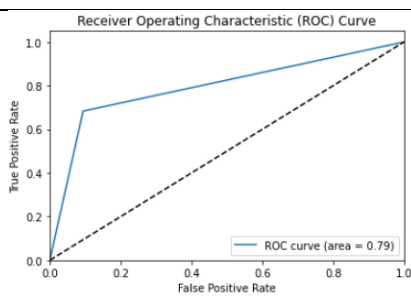
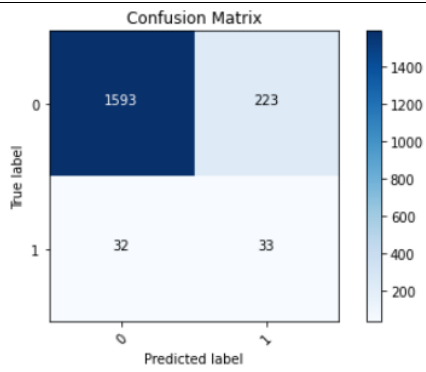
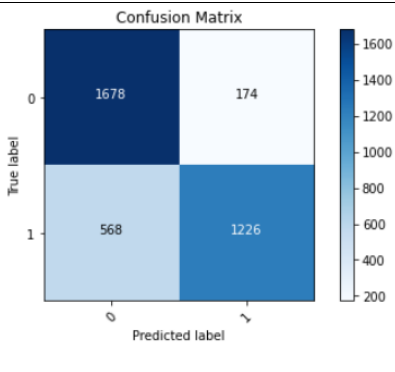
No.	Reduced Dataset	Augmented Transformed Dataset
1	Parameters: Ridge(alpha=0.20090000000000002) Test Accuracy: 0.8958001063264222 Test Precision: 0.8958001063264222 Test Recall: 0.8958001063264222 Test F1-score: 0.8958001063264222 Test AUC: 0.7894739071501186	Parameters: Ridge(alpha=5.496700000000001, solver='saga') Test Accuracy: 0.8173340647284696 Test Precision: 0.8173340647284696 Test Recall: 0.8173340647284696 Test F1-score: 0.8173340647284697 Test AUC: 0.8302189503769464
2	<p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>ROC curve (area = 0.79)</p>	<p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>ROC curve (area = 0.83)</p>
3	<p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p>	<p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p>

After performing hyperparameter tuning, we got the best validation accuracy of 0.8958 with alpha value of 0.2009 as shown at the table above. Hence, we trained the Ridge Regression model using these values of hyperparameter.

The above metrics are the results obtained from training the Ridge Regression model. Although the model is not designed for classification, and by rounding up predicted labels the model will predict out of range labels. Nevertheless, by studying the ROC curve, this model is able to have high performance.

The model is further improved when it is trained on the augmented dataset by observing its ROC curve. As for the confusion matrix is due to the increase in the 1 label, thus the model predicts a wider range of labels.

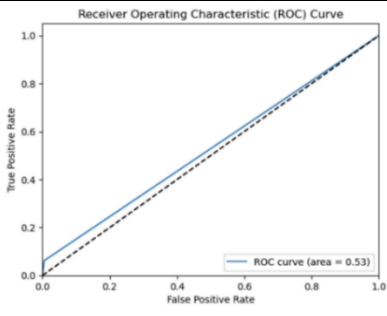
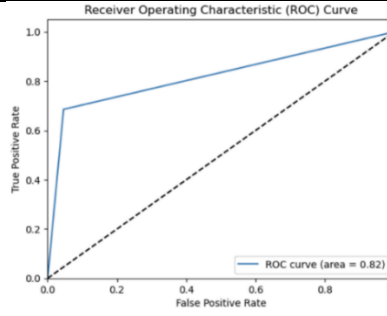
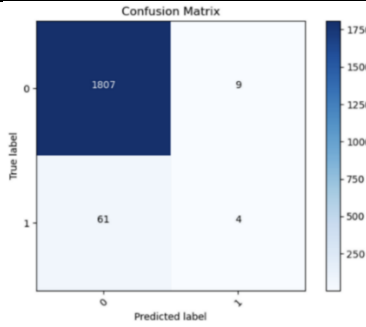
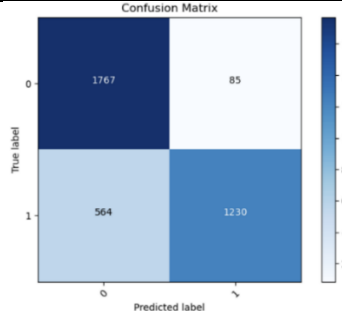
#### 4. Decision Tree Classifier

No.	Reduced Dataset	Augmented Transformed Dataset
1	Parameters: DecisionTreeClassifier(max_depth=4) Test Accuracy: 0.8644338118022329 Test Precision: 0.12890625 Test Recall: 0.5076923076923077 Test F1-score: 0.20560747663551404 Test AUC: 0.6924474754320569	Parameters: DecisionTreeClassifier(max_depth=3) Test Accuracy: 0.7964893033461328 Test Precision: 0.8757142857142857 Test Recall: 0.6833890746934225 Test F1-score: 0.7676894176581089 Test AUC: 0.7947182954460632
2	 <p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate vs False Positive Rate</p> <p>ROC curve (area = 0.69)</p>	 <p>Receiver Operating Characteristic (ROC) Curve</p> <p>True Positive Rate vs False Positive Rate</p> <p>ROC curve (area = 0.79)</p>
3	 <p>Confusion Matrix</p> <p>True label vs Predicted label</p> <p>Values: (0,0)=1593, (0,1)=223, (1,0)=32, (1,1)=33</p>	 <p>Confusion Matrix</p> <p>True label vs Predicted label</p> <p>Values: (0,0)=1678, (0,1)=174, (1,0)=568, (1,1)=1226</p>

After performing hyperparameter tuning, we got the best validation accuracy of 0.86443 with a max depth of 4 as shown at the table above. Hence, we trained the Decision Tree model using these values of hyperparameter.

The above metrics are the results obtained from training the Decision Tree model. We can observe the model has a high accuracy but a low precision. The precision value of 0.12 shows that the model is unable to generate good positive identification. Whereas the F1 score also implies that the model is suffering due to the imbalance nature of the dataset. After augmenting the dataset, the performance of the model shot up as precision rose to 0.875 which implies the model can correctly predict positive identification. Besides that, the ROC graph implies that the model is performing well in the prediction task.

## 5. SVC

No	Reduced Dataset	Augmented Transformed Dataset
1	Best Parameters: Regularization Parameter: 10, Gamma: Test Accuracy: 0.9627857522594365 Test Precision: 0.3076923076923077 Test Recall: 0.06153846153846154 Test F1-score: 0.10256410256410257 Test AUC: 0.5282912572009488	Best Parameters: Regularization Parameter: 10, Gamma: Test Accuracy: 0.821996708721887 Test Precision: 0.935361216730038 Test Recall: 0.68561872909699 Test F1-score: 0.7912512061756193 Test AUC: 0.8198612004016267
2	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.53)	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.82)
3	 Confusion Matrix True label vs Predicted label Values: (0,0)=1807, (0,1)=9, (1,0)=61, (1,1)=4	 Confusion Matrix True label vs Predicted label Values: (0,0)=1767, (0,1)=85, (1,0)=564, (1,1)=1230

After performing hyperparameter tuning, we got the best validation accuracy of 0.9521 with regularization parameter of 10 and gamma of 10 as shown at the table above. Hence, we trained the SVC model using these values of hyperparameter.

The above metrics are the results obtained from training the SVC model. Although it got a high accuracy of over 0.9627, we knew that it misleads us by looking at other metrics along with the ROC curve and confusion matrix. The low precision and recall gave us the information of low proportion of positive prediction were correct and low true positive rate. The low F1-score gave us insight of the model was unable to make correct prediction. The AUC score was interpreting the result more correctly than accuracy in our dataset. It was having value of only 0.5283 which indicates the low overall performance of the model.

The confusion matrix summarized the problem faced. As we can observe, most of the correctly classified instances were from the class '0' while the correctly classified instances for class '1' were the minority. Hence, we performed data augmentation using ADASYN on the dataset and trained the model again. The following images showed the results obtained from augmented transform data.

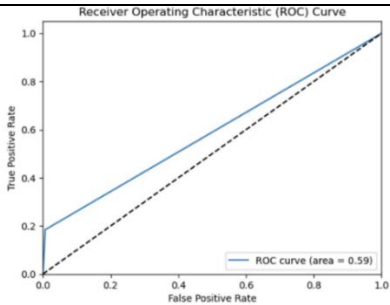
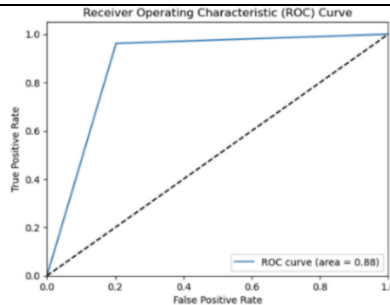
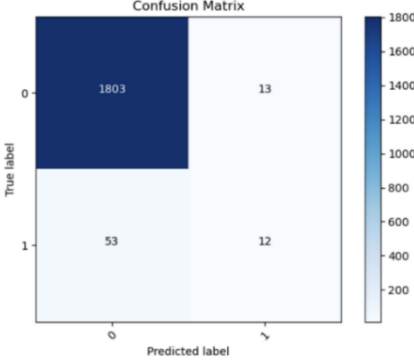
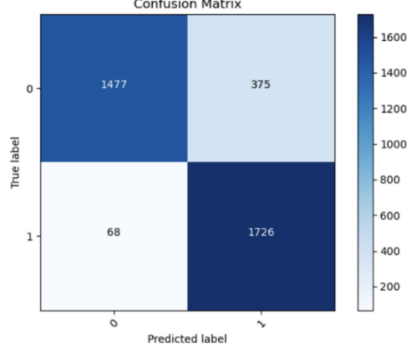
The results seemed to be much better and more reasonable after data augmentation. The SVC model was getting much higher precision, recall, F1-score and AUC score compared to before. We can conclude that the model had the ability to classify the data in a more reasonable way with acceptable accuracy.

We can observe from the ROC curve such that the model exhibited better discriminant power to classify two classes. As the data distribution became more balanced, most of the correctly classified instances laid on the true positive and true negative entries. This indicates that SVC struggles in handling imbalance data. One of the possible reasons is the undesirable margin maximization on majority class. This is due to SVC aiming to maximize the margin between decision boundary and support vectors, hence the imbalance data may cause SVC to prioritize the majority class margin and neglect the



minority class. The next reason could be the decision boundary biased toward majority class. This happens because one of the classes has far lesser samples hence it would be falsely classified as another class. The last reason we thought would be the similar misclassification cost for the two classes. This results in the effect of misclassifying the minority class could be more significant than the minority class, leading to bad performance. In conclusion, the use of data augmentation techniques such as ADASYN improved the performance of SVC when dealing with imbalance dataset.

## 6. MLP

No	Reduced Dataset	Augmented Transformed Dataset
1	Best Parameters: Solver: adam, Learning Rate: 0.005, Alpha: 0. Test Accuracy: 0.9649122807017544 Test Precision: 0.48 Test Recall: 0.18461538461538463 Test F1-score: 0.2666666666666667 Test AUC: 0.5887283971535073	Best Parameters: Solver: adam, Learning Rate: 0.01, Alpha: 0. Test Accuracy: 0.8784969829950631 Test Precision: 0.8215135649690624 Test Recall: 0.9620958751393534 Test F1-score: 0.8862644415917843 Test AUC: 0.8798060369217285
2	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.59)	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.88)
3	 Confusion Matrix True label vs Predicted label Matrix values: (0,0)=1803, (0,1)=13, (1,0)=53, (1,1)=12	 Confusion Matrix True label vs Predicted label Matrix values: (0,0)=1477, (0,1)=375, (1,0)=68, (1,1)=1726

After getting the best value of hyperparameters, the MLP model was trained using them and here came the results below. MLP model trained using ADAM as optimization algorithm, learning rate of 0.005 and alpha of 0.0001 was

giving terrible results on the reduced dataset. It produced the test accuracy of around 0.9649 which was a misleading metrics. The precision value of 0.48 was telling us that only about half of the positive predictions were correct and the recall value of about 0.1846 giving us an insight of a low proportion of true positives were predicted correctly. When both the precision and recall were not getting good results, neither the F1-score as it was calculated based on these two metrics. It summarized the model performance as poor by giving the value of around 0.2667. As we were dealing with imbalance dataset, the AUC score provided more reliable result than accuracy to evaluate the model. It had only 0.5887, indicating the poor performance of MLP classifier on the reduced dataset.

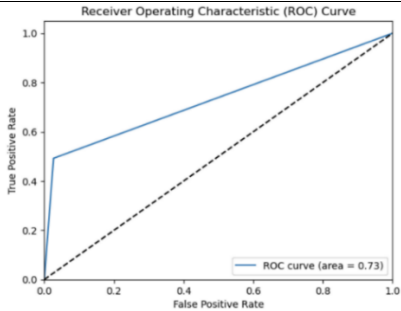
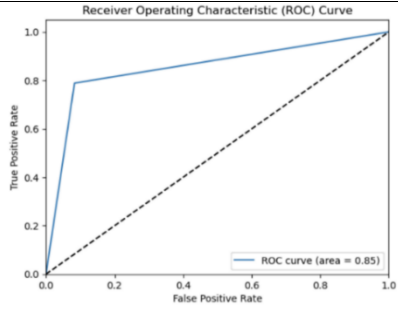
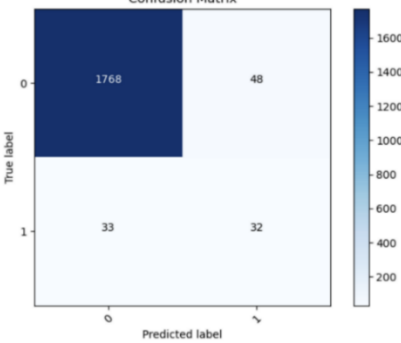
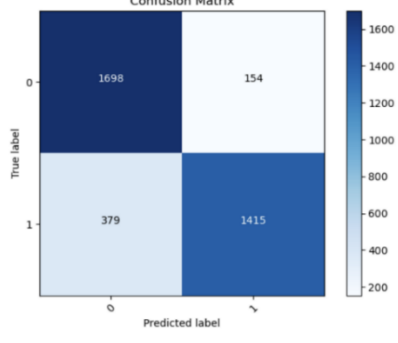
The ROC curve below was giving us an insight of poor performance of MLP classifier for the reduced dataset by illustrating the low value of AUC. The confusion matrix below clearly shows that the model biased toward the majority class, such that most of the correctly classified instances were from class "0". From here, we can observe that SMOTE alone was not enough to eliminate the effect of imbalance dataset. Hence, data augmentation using ADASYN would be carried out to be tried on the training section of the MLP model again.

The results obtained from augmented dataset were much better. Not only having acceptable accuracy and AUC score, the precision and recall were getting significantly higher values than model without using augmented dataset. The same applied to F1-score which derived from the precision and recall.

When we looked at the ROC curve, it was closer to the top left corner hence indicating the higher discriminant ability of the MLP classifier to classify the two classes. Followed by the confusion matrix which indicates the better overall performance of MLP model using augmented dataset as the amount of correctly classified instances became more balanced, solving the undesired majority and minority class scenario. In conclusion, the alternatives of treating imbalance dataset had improved the overall performance of MLP classifier, at the same time telling us the inability of MLP in dealing those type of

imbalance dataset. There were a few possible reasons, one of them would be due to the data shortage in minority class. When this happens, the MLP could be facing difficulties in learning the essential patterns and generalization, causing low model performance. Besides, MLP was encountering biased learning toward majority class. Similar outcomes produced by this issue as compared to data shortage. Finally, the similar misclassification costs from minority and majority cause more serious negative effects when misclassifying the minority class. To conclude, the use of data augmentation technique solved the inability of MLP in training imbalance dataset.

## 7. Logistic Regression

No	Reduced Dataset	Augmented Transformed Dataset
1	Best Parameters: Penalty: 11, Inverse of regularization strength: 0.1 Test Accuracy: 0.9569377990430622 Test Precision: 0.4 Test Recall: 0.49230769230769234 Test F1-score: 0.44137931034482764 Test AUC: 0.7329379871230092	Best Parameters: Penalty: 11, Inverse of regularization strength: 1.0 Test Accuracy: 0.853812397147559 Test Precision: 0.9018483110261313 Test Recall: 0.7887402452619844 Test F1-score: 0.8415105560511449 Test AUC: 0.8527934487649014
2	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.73)	 Receiver Operating Characteristic (ROC) Curve True Positive Rate vs False Positive Rate ROC curve (area = 0.85)
3	 Confusion Matrix True label vs Predicted label Matrix values: (0,0)=1768, (0,1)=48, (1,0)=33, (1,1)=32	 Confusion Matrix True label vs Predicted label Matrix values: (0,0)=1698, (0,1)=154, (1,0)=379, (1,1)=1415

The results using the best hyperparameters for the reduced dataset were shown above. It faced the same problem but not as severe as SVC and MLP. We knew that the high accuracy here was misleading us to a wrong interpretation. The low F1 score caused by low values of precision and recall was telling us that less than half of the positive predictions were correct as well as a low proportion of true positives were predicted correctly. Besides, the AUC score of around 0.7329 in LR seemed to be more acceptable than SVC and MLP but still improvable.

The ROC curve above demonstrates the AUC score on it. The curve was further away from the down left corner but still far away from the top left corner. This indicated that the LR model had a certain discriminant ability to distinguish the two classes but was still insufficient. The following was the confusion matrix which illustrated to us that LR facing the struggles as other models regarding the imbalance dataset as the proportions of true positives and true negatives were showing large disparity. We came to a conclusion here, that SMOTE alone was not enough to deal with the imbalance dataset problem. Hence, augmented transformed data via ADASYN will be tried on the LR model training.

The results from LR model after training with augmented transformed data were getting improvement. All the metrics were getting values higher than 0.7 which seemed to be more reasonable. Both the accuracy and AUC score were getting values more than 0.85 which indicated the better overall performance of LR model. Besides, the higher precision and recall also resulted in a higher F1 score. At the same time, it gave us an insight into the model to better discriminate between positive and negative instances.

The ROC curve below showed that the curve was getting closer to top left corner, indicating better performance of the model via the use of augmented transformed data. Moving on to the confusion matrix part, the proportions of true positives and true negatives became balanced and getting higher accuracy at the same time. This was the positive effects brought by augmented transformed data to imbalance dataset. A possible reason LR struggled in imbalance dataset was due to skewed class distribution toward majority class.

When the minority class samples were too little, it could be hard for the LR model to learn about the dataset and give accurate results for the minority class.

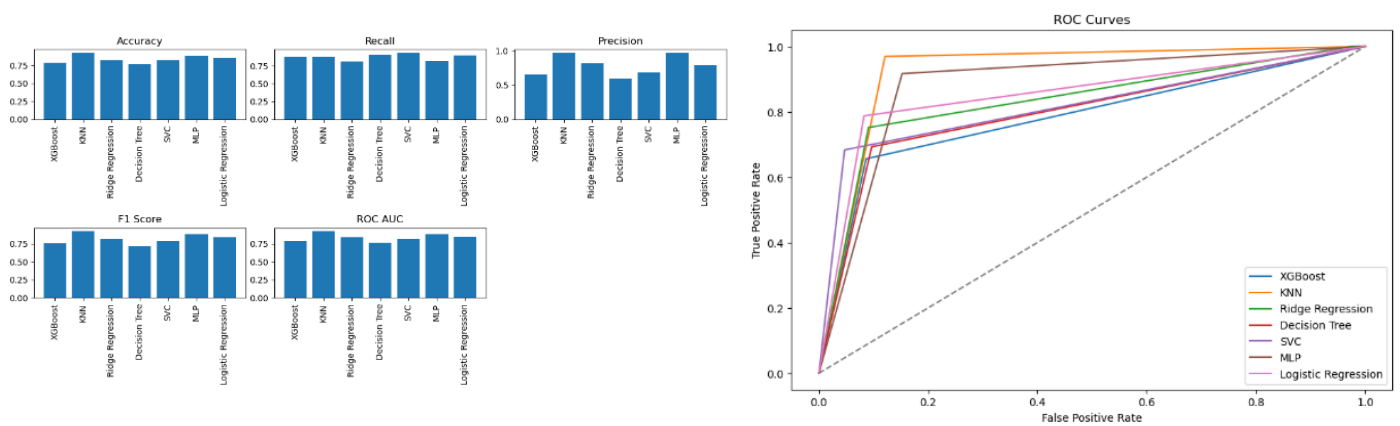
## 8. Models Comparison

Performance of all models on reduced and augmented dataset is shown below:

<b>N o.</b>	<b>Model</b>	<b>Dataset</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1- Score</b>	<b>AUC- ROC</b>
1	XGBoost	Reduced	0.921	0.229	0.538	0.321	0.737
		Augmented	0.789	0.884	0.657	0.754	0.787
2	KNN	Reduced	0.882	0.128	0.415	0.196	0.657
		Augmented	0.903	0.842	0.996	0.912	0.902
3	Ridge Regression	Reduced	0.887	0.887	0.887	0.887	0.804
		Augmented	0.854	0.854	0.854	0.854	0.859
4	Decision Tree	Reduced	0.886	0.153	0.508	0.236	0.704
		Augmented	0.8117	0.810	0.862	0.828	0.811
5	SVC	Reduced	0.963	0.308	0.062	0.103	0.528
		Augmented	0.822	0.935	0.686	0.791	0.820
6	MLP	Reduced	0.965	0.480	0.185	0.267	0.589
		Augmented	0.878	0.822	0.962	0.886	0.880
7	LR	Reduced	0.957	0.400	0.492	0.441	0.733
		Augmented	0.854	0.902	0.789	0.842	0.853

Based on the table results above, we can see that all models have improved their performance when using augmented dataset. Thus, we can conclude that to train the classification problem well, we need a balanced and well-organized dataset, which means in our case, feature selection, factor analysis, and data-augmentation is important to improve the model evaluation performances. Thus, we will be using dataset with data augmentation in our final proposed solution.

Next, we will compare the performances of each model to determine which is the best fit to our classification problem.



Based on the graphs visualization above, we could see that all model's performance has not many differences, all of them can achieve good result in the classification of bankruptcy problem.

To choose the best model, we have decided to use F1 score to determine it. This is because F1 score combines both precision and recall into one metric. It is the harmonic mean of precision and recall and is probably the most used metric for evaluating binary classification models.

Based on the result, KNN has the highest F1 score, followed by MLP, LR, Ridge Regression, SVC, XGBoost and Decision Tree. Since the F1 score of these models are close to each other, we decided to perform ensemble model of the 2 highest F1 score model: KNN and MLP in our proposed solution.

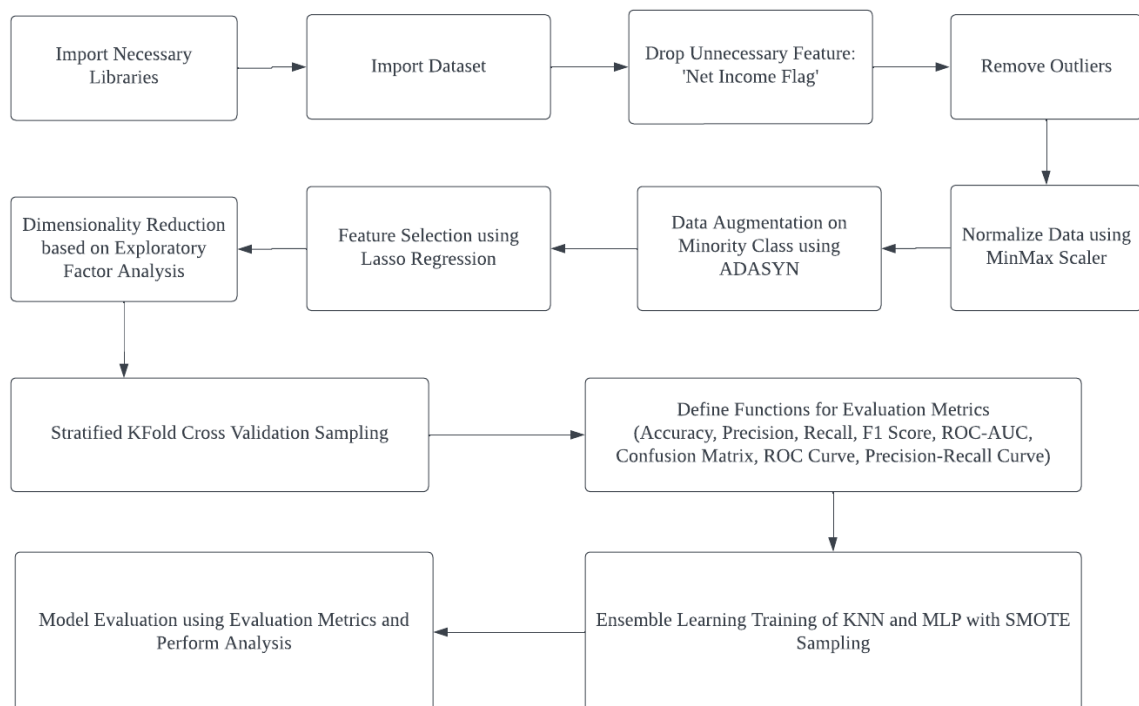
## 5.0 Proposed Solution

In this section, we will propose the final solution for our classification problem.

The proposed solution will be divided to 4 general steps, which are:

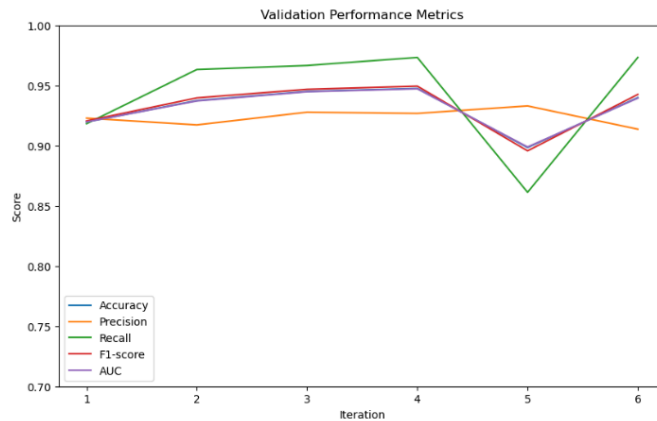
1. Import library and dataset.
2. Data preprocessing, data augmentation, feature selection and dimensionality reduction
3. Model training using ensemble learning of KNN and MLP
4. Model evaluation and result analysis

Followed is the flow chart of the solution:



## Results and Analysis of Proposed Solution

Validation Performance of Ensemble Model during training phase:



Based on the validation performance metrics graph above, we can see that the performance of every cross validation is good, most of the evaluation metrics are good, with most of the values of evaluation metrics is above 0.90, and consistent for every cross-validation step, indicating that this ensemble model learnt and trained well in our final proposed dataset.

## Result and Analysis on Test Dataset

---

```

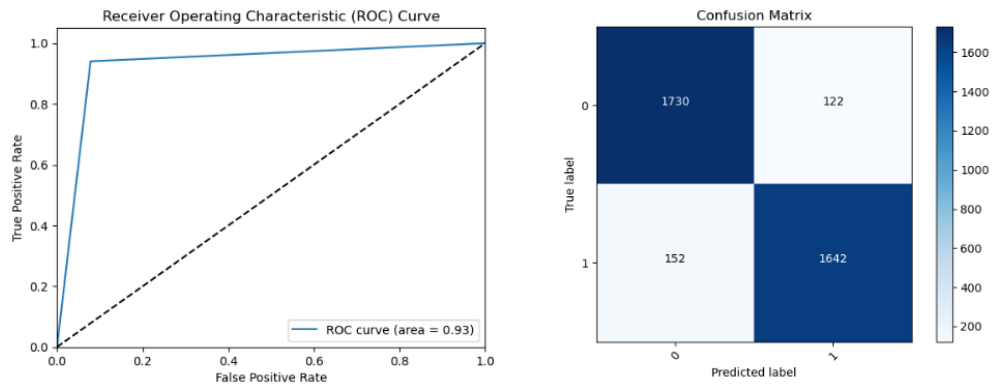
Test Dataset Performance Using Ensemble Model:
Accuracy: 0.9308831596269885
Precision: 0.9208515283842795
Recall: 0.9403567447045708
F1-score: 0.9305019305019305
ROC-AUC: 0.9310315041017454

```

Based on the evaluation metrics, the ensemble achieved strong performance on test dataset.

With a precision of 0.921, the model shows a high level of precision in identifying positive instances. This shows that the model has a low rate of false positives, minimizing the occurrence of misclassified negative instances. Next, the recall score of 0.940 demonstrates the model's ability to capture a large proportion of the actual positive instances, indicating a low rate of false negatives. This means that the model has effectively identified a substantial number of positive instances. Meanwhile the F1 score of 0.931 suggests that the model has achieved a good trade-off between precision and recall, indicating that it can reliably classify instances while minimizing both false positives and false negatives.





Based on the ROC Curve and confusion matrix of the ensemble model prediction on test dataset, we can do some interpretation:

1. True Positive: 1730 – This indicates the number of instances that were correctly classified as positive.
2. False Positive: 122 – This indicates the number of instances that were wrongly predicted as positive when they were actually negative.
3. False Negative: 152 – This signifies the number of instances that were wrongly predicted as negative when they were actually positive.
4. True Negative: 1642 – This represents the number of instances that were correctly predicted as negative.

The ROC area of 0.93 indicates that the model has a high discriminatory power and is effectively at differentiating between the two classes.

In summary, based on the confusion matrix and ROC area of 0.93, we can conclude that the model has performed well, as it has a high F1 score, with a relatively low number of false positives and false negatives.

## 6.0 Future Works and Conclusion

### Future Works

There are still a lot of methods we can implement on this topic in the future time. For the dimensionality reduction based on factor analysis we applied in this project, as we were following Kaiser-Guttman Rule to decide the retain factors, there are some

factors which no primary factor belongs to them, but we still retain those factors, this might lead to increasing of complexity and multicollinearity. Thus, in the future we might study another way of deciding which factors should we retain.

Moreover, we can perform more feature engineering on the dataset as the identification and selection of relevant features are vital in bankruptcy classification. We can develop a feature engineering method based on the knowledge and information from financial analyst, financial statements and other relevant sources.

Moreover, we can put more effort on benchmarking and evaluation of classification results. We can develop and standardize a benchmark datasets and evaluation metrics in order to facilitate the comparison of different classification models. We should also emphasize more and establishing robust benchmarks specially for company bankruptcy classification in order to assess and analyse the performance of different algorithms accurately.

## **Conclusion**

In this project, we have implemented a proposed solution to perform company bankruptcy classification. The solution is proposed after several experiments and analysis.

There are some key insights based on our studies in this project. In our project, we have come out with conclusion that:

1. Data augmentation is important for imbalanced data, as it can enhance model generalization, reduce overfitting and enhance decision boundaries.
2. Dimensionality reduction using factor analysis helps us to simplify complex dataset and improve model classification by identifying underlying latent factors.
3. Ensemble learning of several models can improve the classification result, as it can reduce bias and variance, improve generalization and more robust against noise and outliers.

## 7.0 References:

1. Boyacioglu, M. A., Kara, Y., & Baykan, Ö. K. (2009). Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of Savings Deposit Insurance Fund (SDIF) transferred banks in Turkey. *Expert Systems with Applications*, 36(2), 3355–3366. <https://doi.org/10.1016/j.eswa.2008.01.003>
2. Chen, M. Y. (2011). Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches. *Computers & Mathematics With Applications*, 62(12), 4514–4524. <https://doi.org/10.1016/j.camwa.2011.10.030>
3. Collins, R. W., & Green, R. (1982). Statistical methods for bankruptcy forecasting. *Journal of Economics and Business*, 34(4), 349–354. [https://doi.org/10.1016/0148-6195\(82\)90040-6](https://doi.org/10.1016/0148-6195(82)90040-6)
4. Son, H., Hyun, C., Phan, D., & Hwang, H. J. (2019). Data Analytic Approach for bankruptcy prediction. *Expert Systems with Applications*, 138, 112816. <https://doi.org/10.1016/j.eswa.2019.07.033>

## MARKING RUBRICS

		Score and Descriptors				
No.		Poor	Average	Excellent	Weight (%)	Mark
<b>Component 1: Project Development</b>						
		<b>0-2</b>	<b>3</b>	<b>4-5</b>	<b>5</b>	
1	Use of Data Set	No application pre-processing data. Only use the raw data.	Apply partial complete of data pre-processing.	Apply complete data pre-processing methods on data sets.		
		<b>0-2</b>	<b>3</b>	<b>4-5</b>	<b>5</b>	
1	Code Quality	Design codes that are poorly structured. Not fully functional. Not documented.	Codes are sufficiently documented and mostly functional. Satisfactorily structured.	Codes are fully functional and well structured and documented.		
		<b>0-2</b>	<b>3</b>	<b>4-5</b>	<b>5</b>	
2	Method Functionalities	Incomplete development of the method.	Complete development of method.	Completed\ enhanced the developed methods.		
		<b>0-2</b>	<b>3-5</b>	<b>6-10</b>	<b>10</b>	
3	Performance Evaluation	Lack evaluation of performance and analysis.	Partially complete performance evaluation.	Provide all analysis and performance evaluation.		
				<b>Subtotal</b>	<b>25</b>	
<b>Component 2: Project Report + Project Demonstration</b>						
		<b>0 - 6</b>	<b>7 - 10</b>	<b>11-15</b>	<b>15</b>	
1	Project Report	Poor writing quality Poor or no formatting / presentation Lack of discussion for statistical method's results.	Satisfactory writing quality, grammar and flow. Substantial content on the statistical methods.	Good writing quality, grammar and flow Well formatted and good presentation Demonstrate excellent experimental analysis of statistical methods.		
		<b>0-2</b>	<b>3</b>	<b>4-5</b>	<b>5</b>	
1	Presentation	Poor quality slides Poor time management Speech that is unclear.	Satisfactory quality slides Speech that is satisfactory and understandable.	High quality slides Good time management Speech that is clear and impactful.		
		<b>0-2</b>	<b>3</b>	<b>4-5</b>	<b>5</b>	
2	Demonstration	Poor demonstration that is unclear of implementation methods.	Satisfactory demonstration of implementation of methods.	Excellent demonstration is fully functioning and logical.		
				<b>Subtotal</b>	<b>25</b>	
				<b>Grand Total</b>	<b>50</b>	

Note to students: Please attach this appendix together with the submission of coursework