PROJECT REPORT ON
CLASSIFICATION MODELLING

DEPARTMENT OF MATHEMATICS AND STATISTICS

Under the guidance of Dr. Shankar Prawesh

By –

Rachna Chaurasia (191100)
Sahil Saini (191118)

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Shankar Prawesh, our academic and project mentor, for providing us with the guidance and patience needed throughout this project. Without his help, this endeavour would have remained incomplete. I would like to thank him for giving me this opportunity to explore the real-life application of Supervised Machine Learning and hence giving an idea about their significance in a Business Analyst's life.

PROBLEM STATEMENT

Use the datasets uploaded with this document to complete this assignment. There are two files: train.csv and test.csv. Use train.csv to train your prediction model and test.csv to only test the performance of the prediction model using misclassification error rate. There are twenty predictors labeled $x1,x2,....,x20$ and the dependent variable y has binary class labels (0 and 1). Your project report should contain following information.

(a) Develop a classification model that has lowest misclassification error rate on the test data. What is the training error rate for this model? Also, provide a brief explanation for the good performance of this model. (Hint: a good classification model should have test misclassification error rate below 0.20).

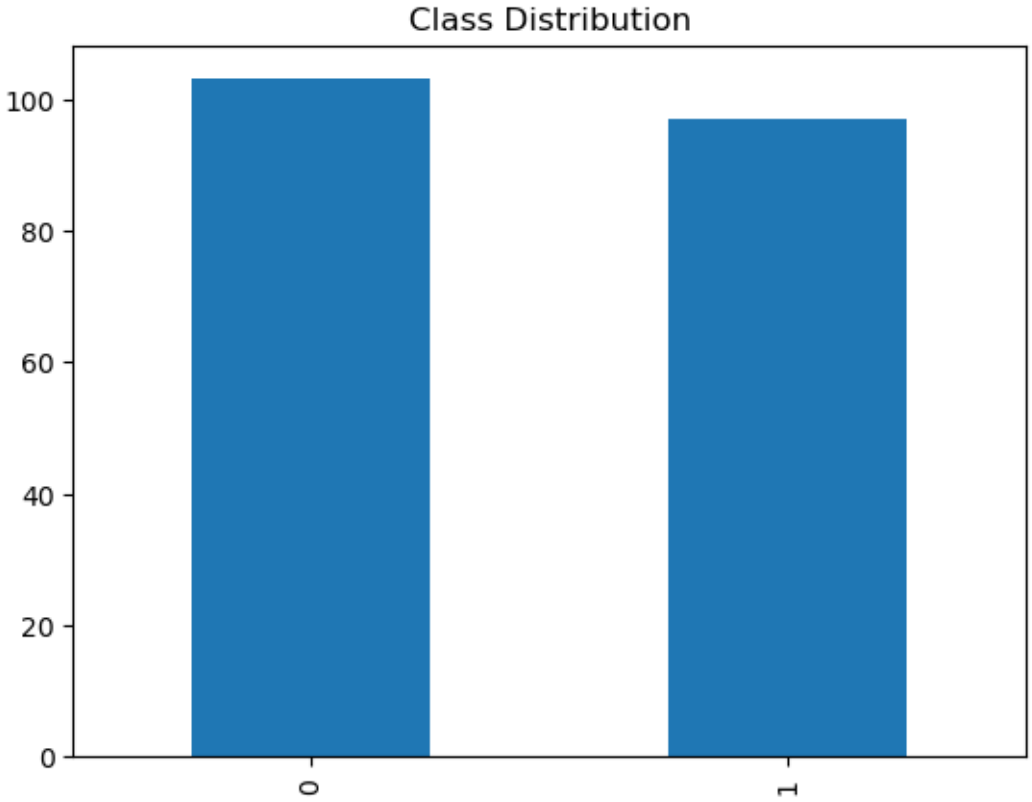
(b) Document all models you experimented with to evaluate the performance on the test data. The code used for these models should be uploaded along with your submission. The report should include a brief description of all classification models you have used to complete the assignment. If you are using a classification model that is not covered in the course, then provide an appropriate reference for this model.

(c) Generate ROC curve for the best performing classification model. How would you interpret the findings from the ROC curve?

(d) Briefly mention the contribution of each team member in the project.

METHODOLOGY

The figure below summarizes the populations of class labels in the dataset. As it can be seen easily that the dataset is perfectly balanced.



We will now fit some classification model listed below to get which classifier is giving better accuracy or we can say less misclassification error-

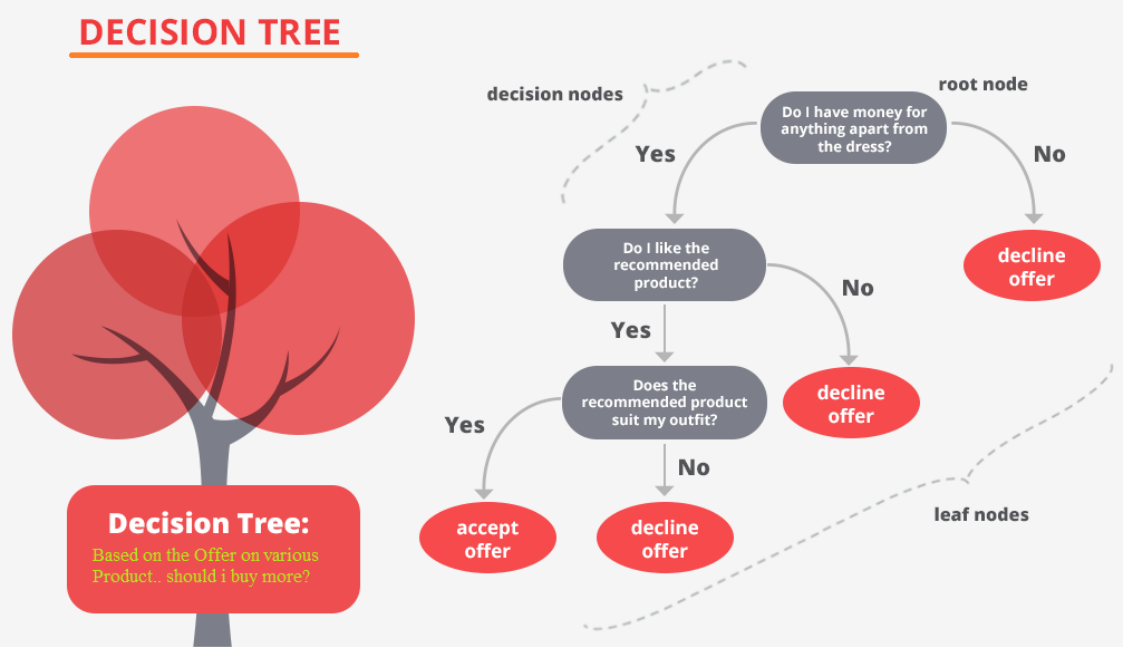
- Decision Tree
- Random Forest
- Logistic Regression
- K nearest neighbour

Let's discuss them all one-by-one and see what fits the data well.

Decision Tree-

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets. These metrics are applied to each candidate subset, and the resulting values are combined (e.g., averaged or majority voting) to provide a measure of the quality of the split. There are two metrics in Decision Tree namely gene impurity and information gain.



Results and Conclusion

Results using Gini Index:

```
Confusion Matrix:
 [[327 164]
 [157 352]]
Accuracy: 67.9
Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.67      0.67       491
           1       0.68      0.69      0.69       509

    accuracy                           0.68      1000
   macro avg       0.68      0.68      0.68      1000
weighted avg       0.68      0.68      0.68      1000

Outcome values :
 352 157 164 327
Misclassification rate :
 0.321
```

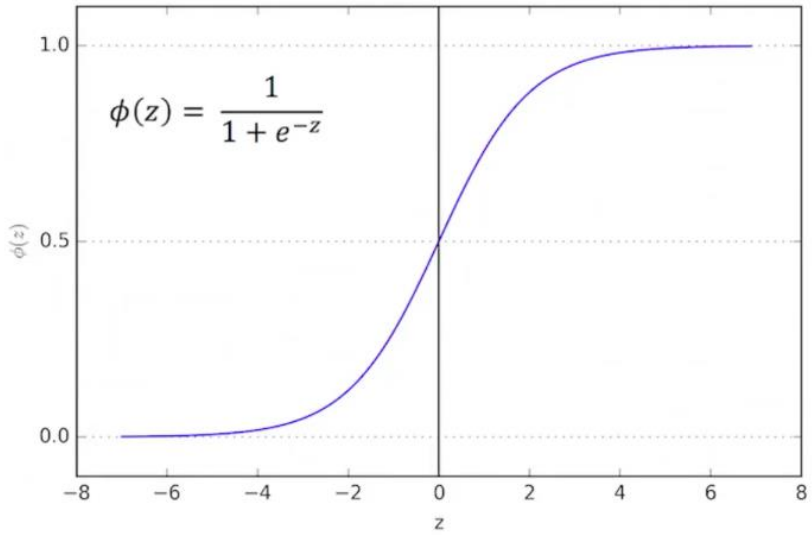
Results using Entropy:

```
Confusion Matrix:
 [[380 111]
 [239 270]]
Accuracy: 65.0
Classification Report:
              precision    recall  f1-score   support

           0       0.61      0.77      0.68       491
           1       0.71      0.53      0.61       509

    accuracy                           0.65      1000
   macro avg       0.66      0.65      0.65      1000
weighted avg       0.66      0.65      0.65      1000

Outcome values :
 270 239 111 380
Misclassification rate :
 0.35
```

Logistic Regression-

Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Logistic regression is most commonly used when the data in question has binary output, so when it belongs to one class or another, or is either a 0 or 1.
This regression technique uses a sigmoid function/logistic function which has an 'S' shape when plotted. The sigmoid functions pushes the values towards the margins that are 0 and 1. It gives an output 0 if the input is a large negative number and 1 if the input is a large positive number. This is the basis of the classification algorithm that is the logistic regression.
The sigmoid function is as follows:

$$Y= 1/ (1+e\text{-}x)$$



Results and Conclusion

```
Confusion matrix :
 [[418  73]
 [ 95 414]]
Outcome values :
 414 95 73 418
Classification report :
              precision    recall  f1-score   support

           0       0.81      0.85      0.83       491
           1       0.85      0.81      0.83       509

    accuracy                           0.83      1000
   macro avg       0.83      0.83      0.83      1000
weighted avg       0.83      0.83      0.83      1000

Misclassification rate :
 0.168
```

Random forest-

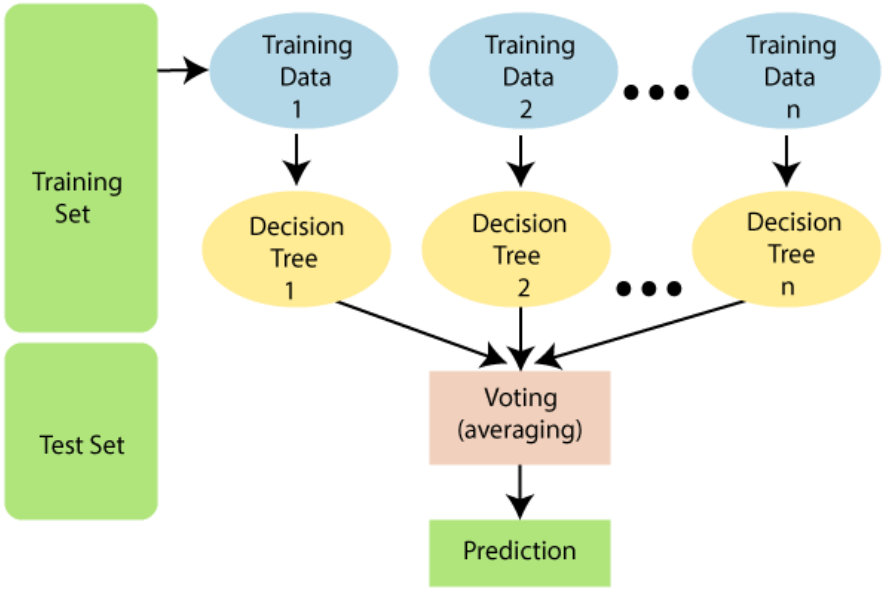
Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':



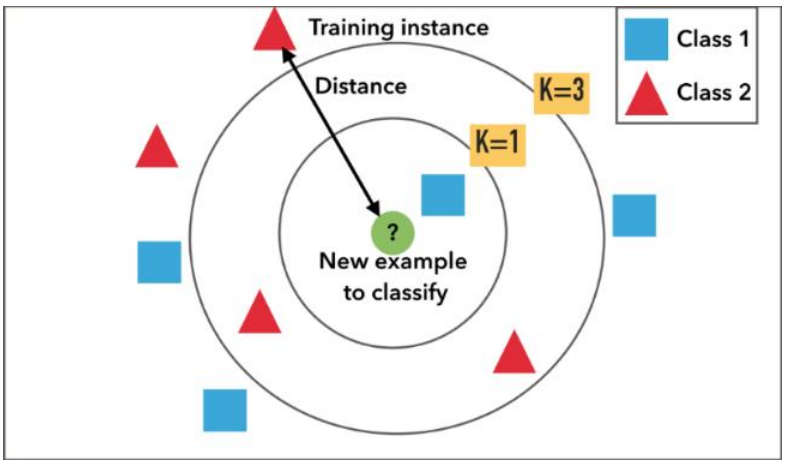
Results and Conclusion

```
Confusion Matrix:
 [[395  96]
 [145 364]]
Accuracy: 75.9
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.80      0.77       491
           1       0.79      0.72      0.75       509

    accuracy                           0.76      1000
   macro avg       0.76      0.76      0.76      1000
weighted avg       0.76      0.76      0.76      1000

Outcome values :
 364 145 96 395
Misclassification rate :
 0.241
```

K nearest neighbour-

In *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy dramatically. The neighbors are taken from a set of objects for which the class (for *k*-NN classification) or the object property value (for *k*-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.




```
Confusion Matrix:
 [[319 172]
 [164 345]]
Accuracy: 66.4
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.65      0.66       491
           1       0.67      0.68      0.67       509

    accuracy                           0.66      1000
   macro avg       0.66      0.66      0.66      1000
weighted avg       0.66      0.66      0.66      1000

Outcome values :
 345 164 172 319
Misclassification rate :
 0.336
```