CSE422 Lab Project Report

Name: Mihir Das
ID: 22299480

1. Introduction

Our project basically aims to predict whether a customer will subscribe to a term deposit based on demographic data and historical marketing. The objective is by using binary classification models correctly predicting the outcome. To do so we have used exploratory data analysis (EDA), correlation testing and machine learning algorithms to find meaningful patterns and to achieve a good accuracy score.

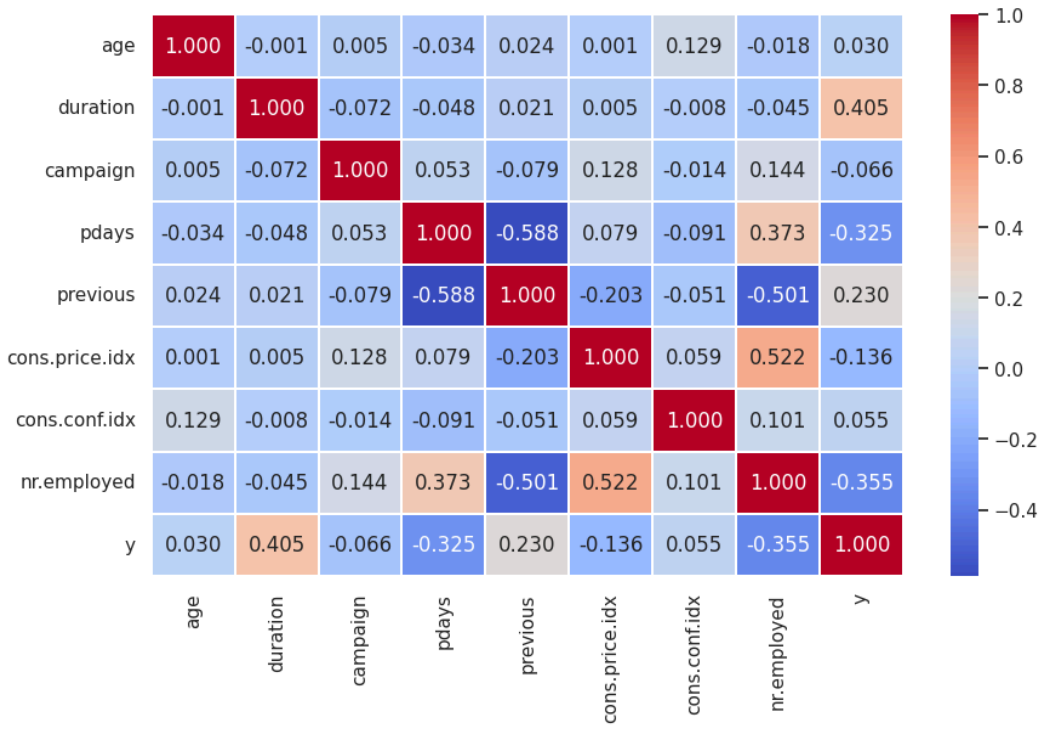
2. Dataset description

- **Total features:** 21
- **Target Column:** y (binary classification: “yes” or “no”)
- **Total Records:** 41,188
- **Type of Problem:**Classification,as we predict a binary outcome.
- **Features types:**

  Numerical(10):
  age,duration.campaign,pdays,previous,emp.var.rate,cons.price.idx,cons.conf.idx, euribor3m,nr.employed.

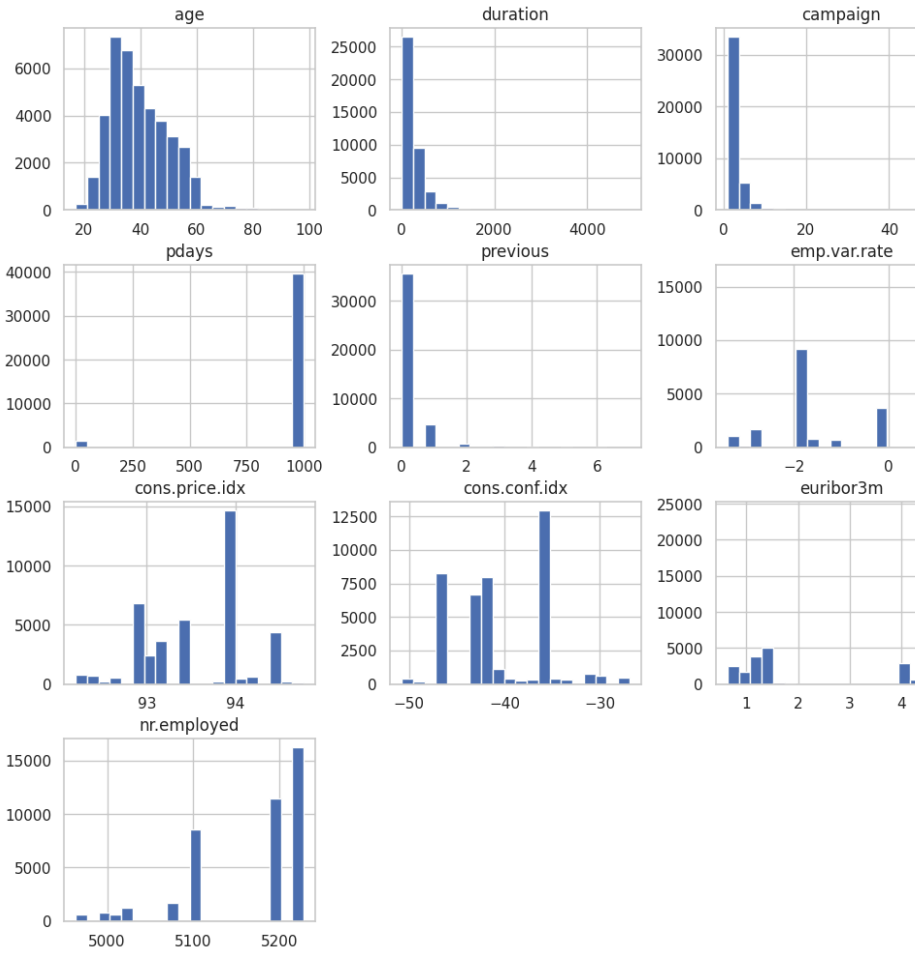
  categorical(11):
  job,marital,education,default,housing,loan,contact,month,day_of_week,outcome, y

Correlation Heatmap:



We used the seaborn heatmap to visualize the feature correlation. The strong insights include:
Duration has the highest positive correlation with the target (y:0.405)
Redundant features(emp.var.rate,euribor 3m) were dropped.
Features like pdays,euribor 3m,and nr.employed show moderate negative correlation.

Imbalance Dataset:



The y column has two classes:

- Yes: 4,640 entries(~11.3%)
- No: 36,548 entries(~88.7%)

We visualized this using a pie chart to confirm the class imbalance.This imbalance may bias model predictions toward the majority class.

3. Exploratory Data Analysis (EDA):

- Duration is the most important feature:Clients who subscribed had longer call durations.
- Boxplots showed significant differences in feature distributions grouped by y.
- Outliers were detected in duration,pdays and previous.
- Density plots showed skewness in some variables,which might impact certain ML models.

4. Dataset splitting

We have split the data into 70% training data and 30% testing data by using train_test_split() imported from sklearn.

5. Model training & testing

We have used 4 models in total.

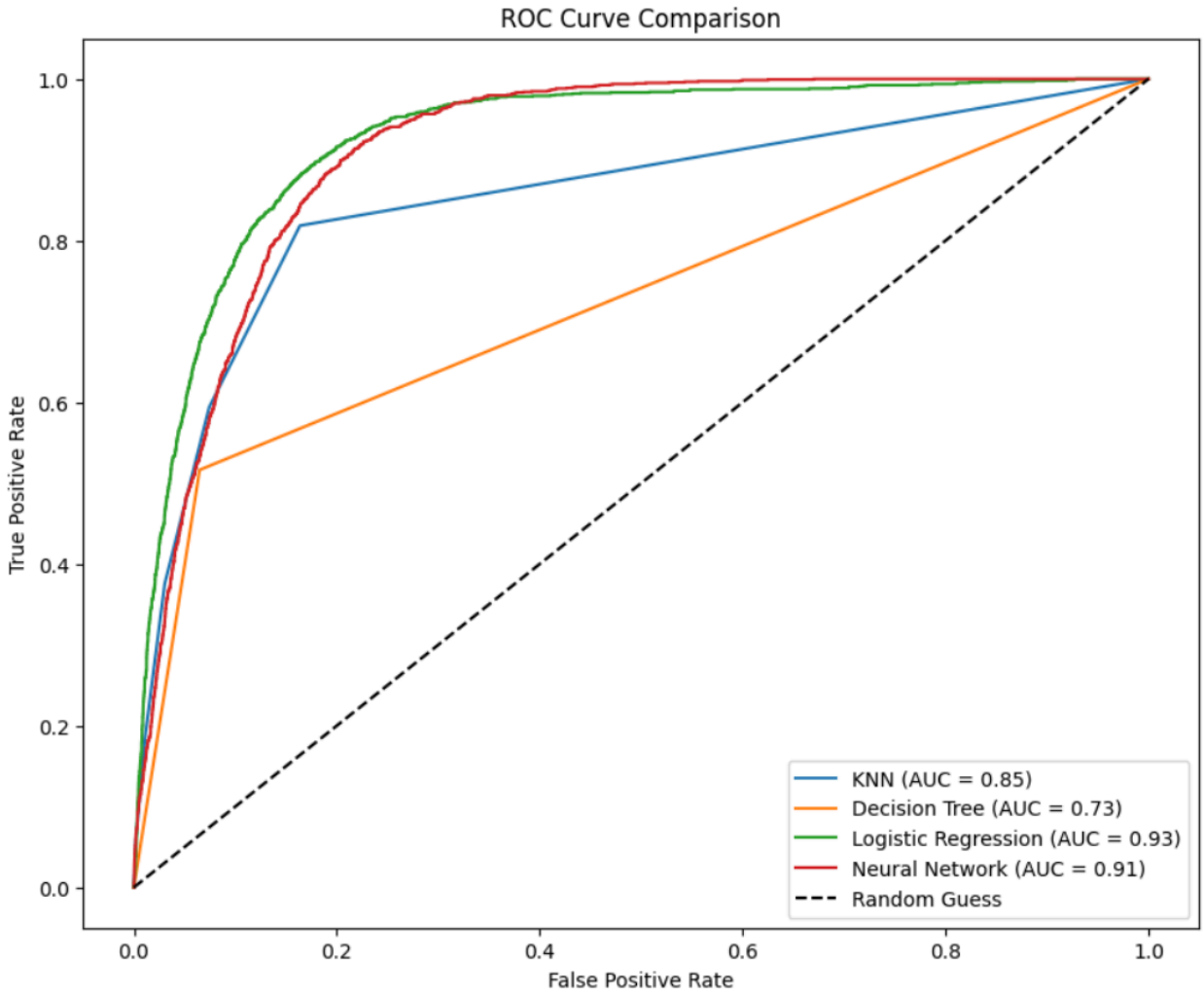
1. Logistic Regression
2. KNN
3. Decision Tree
4. Neural Network (must)

We trained the data (X_train and y_train) in models one by one and then predicted the X_test. Then plotted the ROC curve, AUC score and then the confusion matrix.

6. Model selection/Comparison analysis

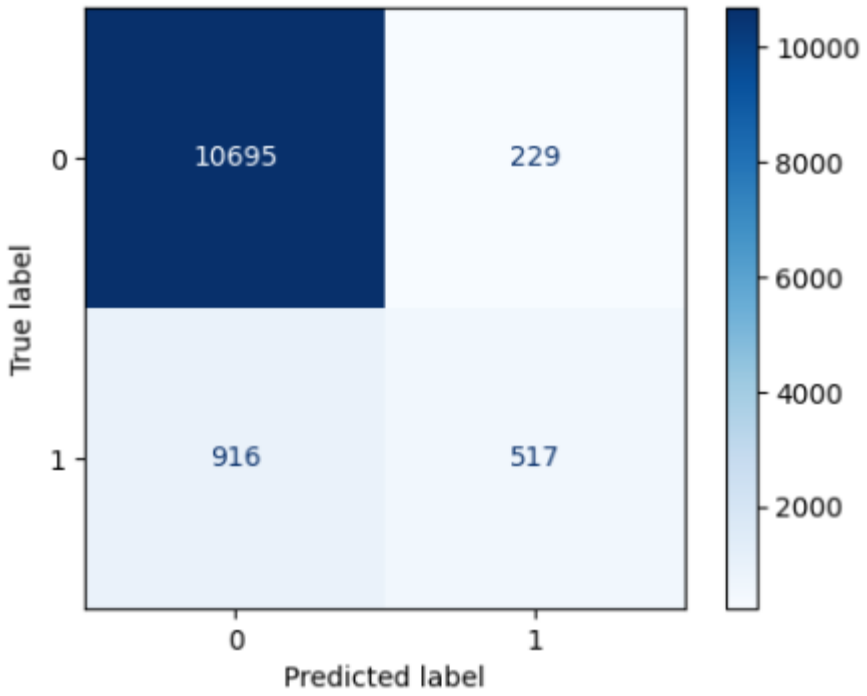
Determining the performance of the models. Since our dataset is perfect for binary classification models and also imbalanced, ROC and AUC curves evaluate the performance of the models. Another great way to evaluate the performance is through a confusion matrix. It basically compares between actual true labels and predicted labels. From which we can see how many are correct and how many are wrong. True labels are on the left and predicted labels on the bottom side which in both cases have two classes named ‘1’, ‘0’ meaning ‘yes’ and ‘no’.

ROC curve and AUC: (performance)

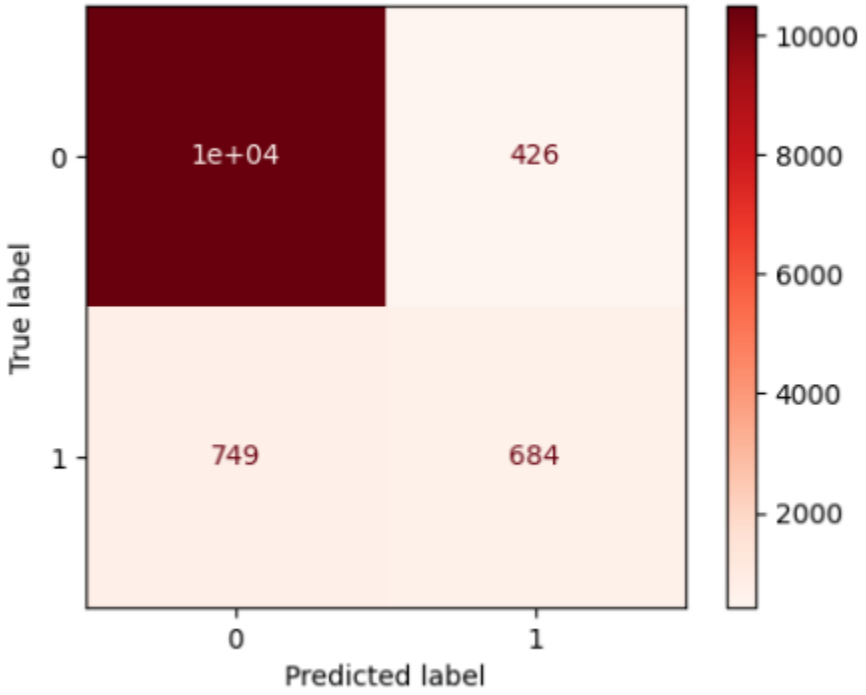


Confusion Matrix: (performance)

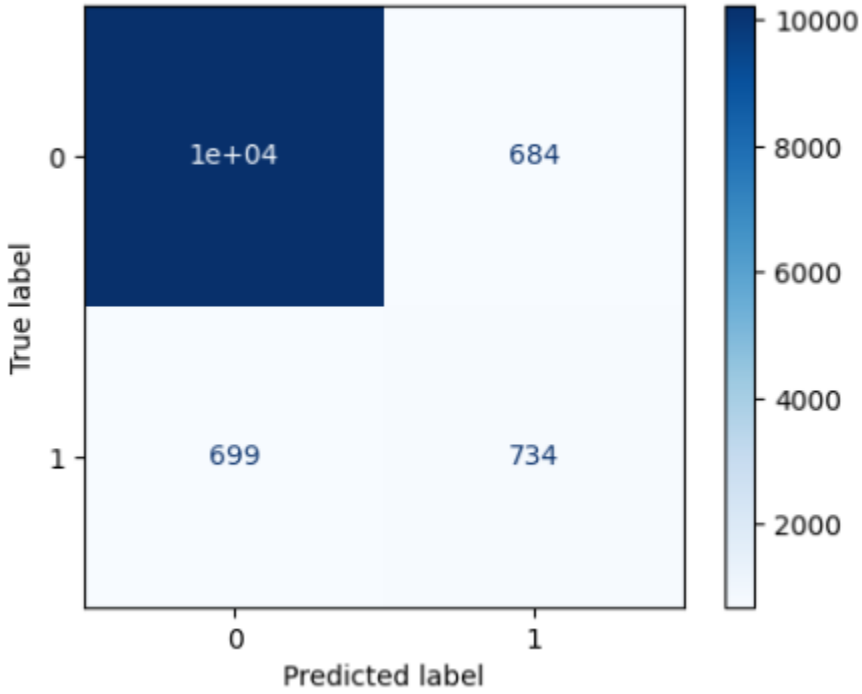
- Logistic Regression:


10695
229
916
517
0
1
True label
0
1
Predicted label
10000
8000
6000
4000
2000

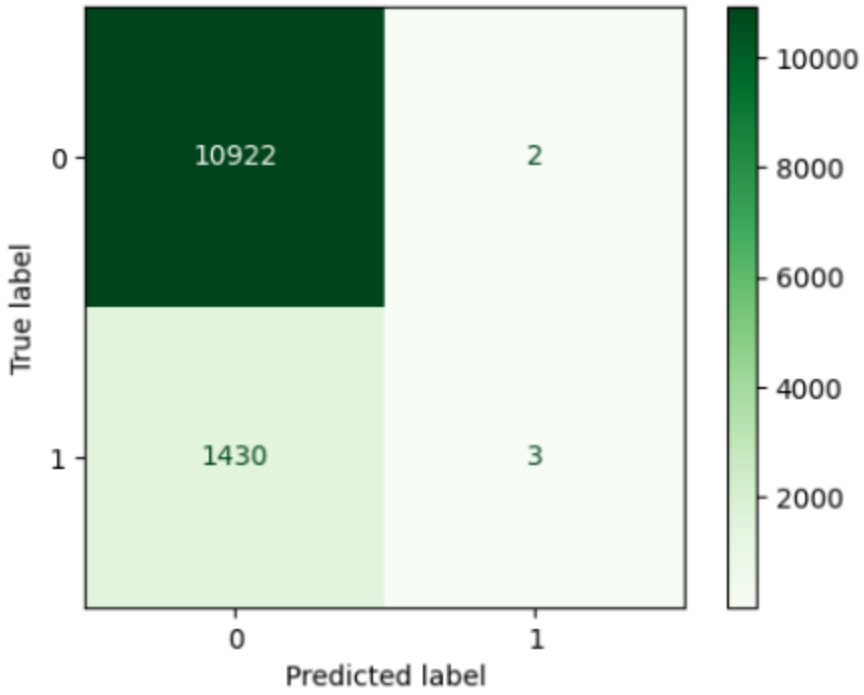
- KNN:


1e+04
426
749
684
0
1
True label
0
1
Predicted label
10000
8000
6000
4000
2000

- Decision Tree:



- Neural Network: (MLPClassifier)

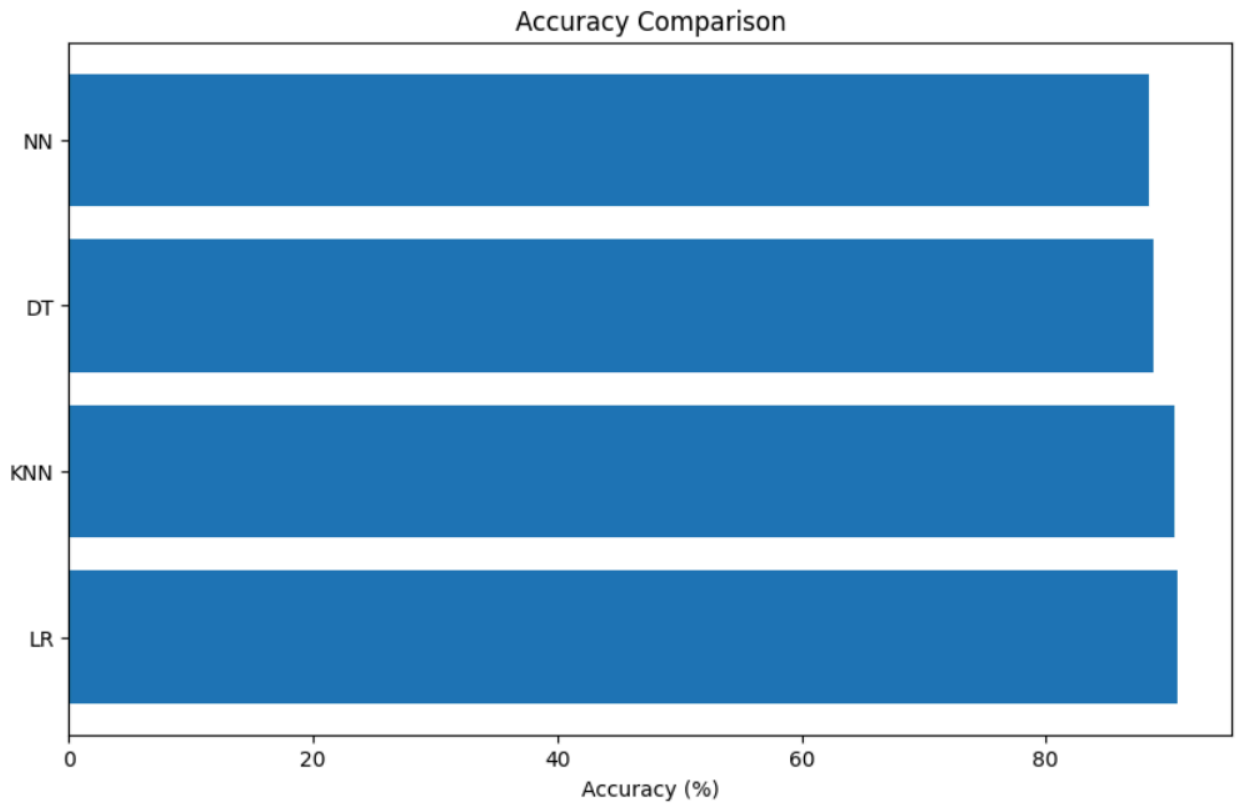


Classification Report: (summary of performance evaluation of all models)

```
Logistic Regression:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     10924
           1       0.69      0.36      0.47      1433

    accuracy                           0.91     12357
   macro avg       0.81      0.67      0.71     12357
weighted avg       0.89      0.91      0.89     12357

--------------------------------------------------
KNN:
              precision    recall  f1-score   support

           0       0.93      0.96      0.95     10924
           1       0.62      0.48      0.54      1433

    accuracy                           0.90     12357
   macro avg       0.77      0.72      0.74     12357
weighted avg       0.90      0.90      0.90     12357

--------------------------------------------------
Decision Tree:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     10924
           1       0.52      0.51      0.51      1433

    accuracy                           0.89     12357
   macro avg       0.73      0.72      0.73     12357
weighted avg       0.89      0.89      0.89     12357

--------------------------------------------------
Neural Network:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94     10924
           1       0.60      0.00      0.00      1433

    accuracy                           0.88     12357
   macro avg       0.74      0.50      0.47     12357
weighted avg       0.85      0.88      0.83     12357
```

Accuracy of all models: (accuracy rates through bar plot)

NN: Neural Network, DT: Decision Tree, KNN, LR: Linear Regression




```
Logistic Regression Accuracy: 0.9073399692481994
KNN Accuracy: 0.9049121955167112
DT Accuracy: 0.8880796309783928
NN Accuracy: 0.8841142672169621
Best accuracy: 90.734% ---> Logistic Regression
```

Neural Networks by TensorFlow:

We have also used Neural Networks by importing from tensorflow() in which there are first layer, hidden layer and final layer. Then compiled the model and used Adam optimizer. Trained the model with 10 epochs with batch size of 30 and finally plotted the loss curve. This way, the model gave us an **accuracy score of 88.41%.**
So, both ways NN performed kind of similarly.

7. Conclusion

In this project, as the outcome was a classification problem, I have implemented four classification models including Neural Network and among them, Logistic Regression has the best accuracy score.
From the ROC curve, we can see the AUC score of Linear Regression and Neural Network was really good respectively 0.93 and 0.91 which means they performed consistently. The least performed one is Decision Tree with AUC = 0.73 which indicates that this may not be the best model for this dataset. Later on we saw, decision tree has the lowest accuracy score in the bunch. The dataset doesn't have any missing value so that's a plus point. We also removed high correlation from correlation heatmap which further helped the models. So, basically EDA and Data Preprocessing are the parts for which our models got such results. Another thing could have been done is, by using SMOT, making the dataset balanced from imbalanced which may have resulted in higher accuracy. But for this project we decided not to but in future, we will definitely look into it.