PROJECT REPORT

CLASSIFICATION OF CUSTOMERS FOR INSURANCE MARKETING CAMPAIGN

GROUP 22

MAHAVIR KATHED (002749678)
Mobile No.: (617) 860 9733
kathed.m@northeastern.edu

MANDAR JADHAV (002784429)
Mobile No.: (857) 799 0479
jadhav.man@northeastern.edu

INTRODUCTION

Customer acquisition is a process where a customer is targeted to purchase a product or a service as per their need and capacity. In this project, we will be focusing on customer data from a bank. Besides usual transactional services, banks also provide car insurance services. This bank organizes regular campaigns to attract new clients. Here we are interested in which existing customer is planning to purchase a car or already has one. It is very important for an organization to plan its campaign to focus on potential consumers who have a high probability of getting converted. This is an initial step and crucial step for any firm to reach out to people to avail of their services, as finding a new customer is way more expensive than keeping an existing one.

PROBLEM

Many banks provide services like life insurance, car insurance, housing insurance, etc. to diversify their source of income. Such industries require detailed analysis of their potential customers so they can optimize their time and resources to target only those who are willing to invest in the service. Using data mining techniques, we tried to develop different models that tell us what segment of people to aim for to achieve the goals achieved. We analyzed bank customer data to predict the behavior of people to the cold calls made and derive the conclusion from training the model.

GOAL OF THE PROJECT

The main aim of this project is to classify a set of people whom the bank can pitch its campaign for car insurance. For this purpose, we are trying to understand what characteristics can lead a customer to purchase car insurance. Every customer is different and has various expectations from insurance, so it is quite challenging to get to a conclusion, but there are some attributes in the dataset that can help us achieve our goal.

APPROACH USED

To start with, we performed Exploratory Data Analysis on the data to get an idea of various trends of attributes for respective classes of customers. After EDA, we conducted Feature Engineering to eliminate irrelevant parameters present in the dataset.

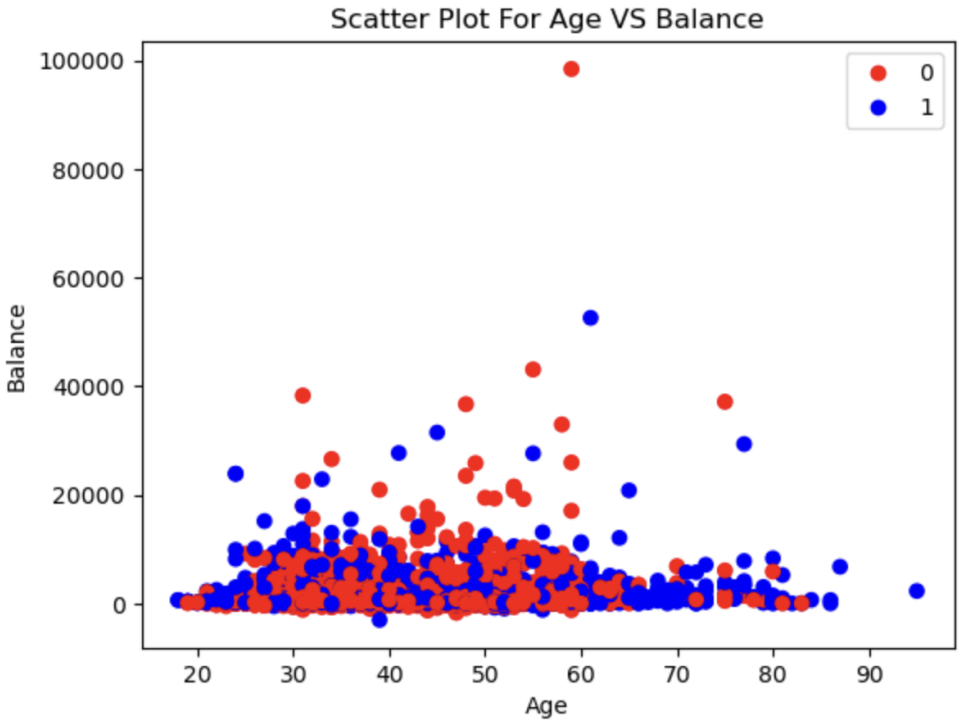
Finally, we used different models like KNN, Naive Bayes, Neural Network, CART, Random Forest, Logistic Regression, etc to the dataset, trained and tested them to see which model performs better in which condition.

ATTRIBUTES IN THE DATASET

Feature	Description	Example
Id	Unique ID number. Predictions file should contain this feature. Age of the client	'1"...."5000''
Age	Age of the client	
Job	Job of the client.	"admin.", "blue-collar", etc.
Marital Status	Marital status of the client	divorced, "married", "single"
Education	Education level of the client	primary, "secondary", etc.
Default	Has credit in default?	"yes" - l, "no" -0
Balance	Average yearly balance, in USD	
HH Insurance	Is household insured	"yes" - l, "no" -0
Car Loan	Has the client a car loan	"yes" - l, "no" -0
Communications	Contact communication type	cellular, "telephone", "NA
Last Contact Month	Month of the last contact	‘’Jan’’, ‘’Fe’’, etc.
Last Contact Day	Day of the last contact	
Call Start	Start time of the last call (HH:MM:SS)	12:43:15
Call End	End time of the last call (HH:MM:SS)	12:43:15
No Of Contacts	Number of contacts performed during this campaign for this client for this client	
Days Passed	Number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)	
Previous Attempts	Number of contacts performed before this campaign and for this client	
Outcome	Outcome of the previous marketing campaign	"failure", "other", "success", "NA"
Car Insurance	Has the client subscribed a Car Insurance?	"yes" - l, "no" -0

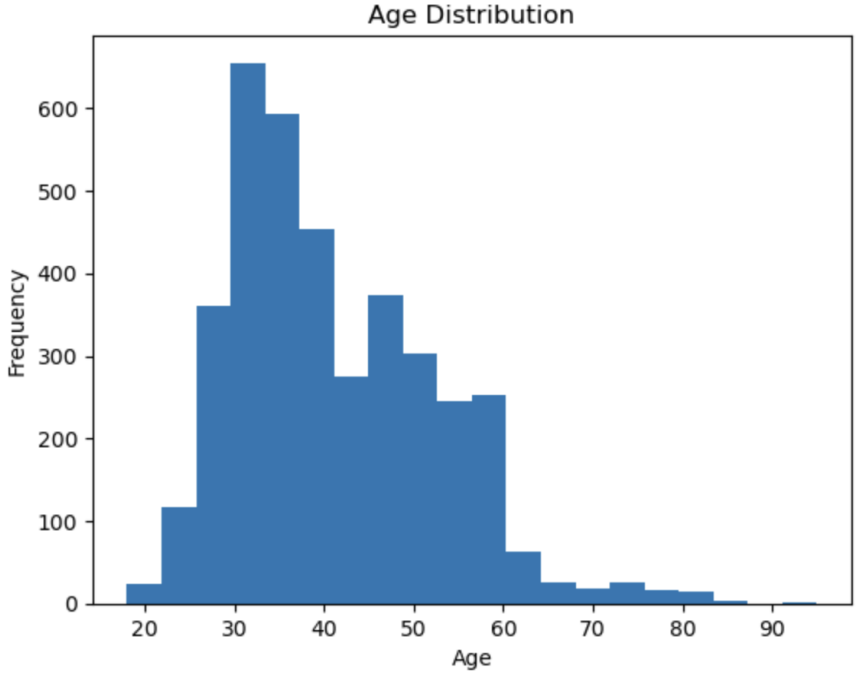
DATA EXPLORATION & VISUALIZATION

- Customer Age VS Account Balance



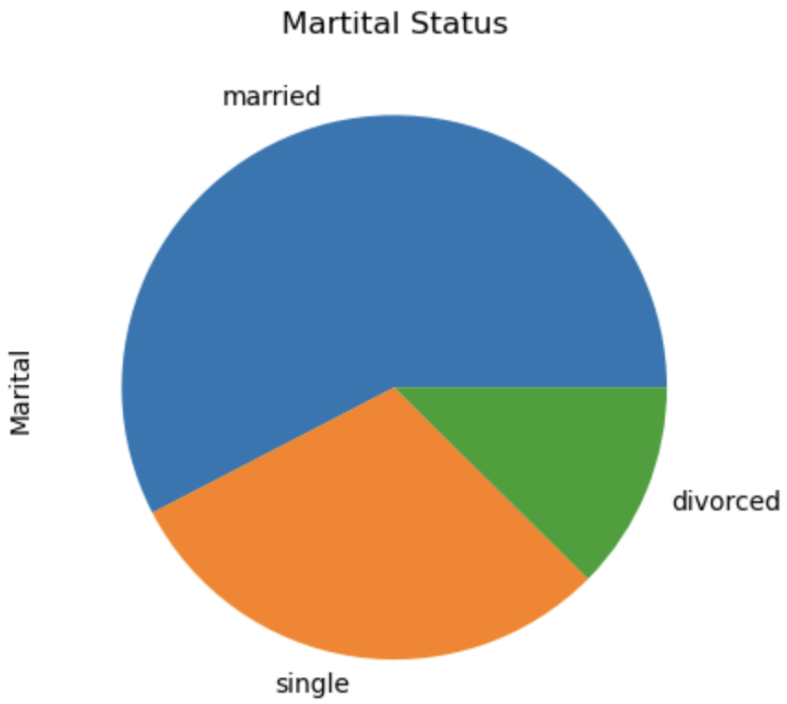
As the linear separation of data is not possible, we have to take other attributes to train the model.

- Distribution of customers as per their Age.



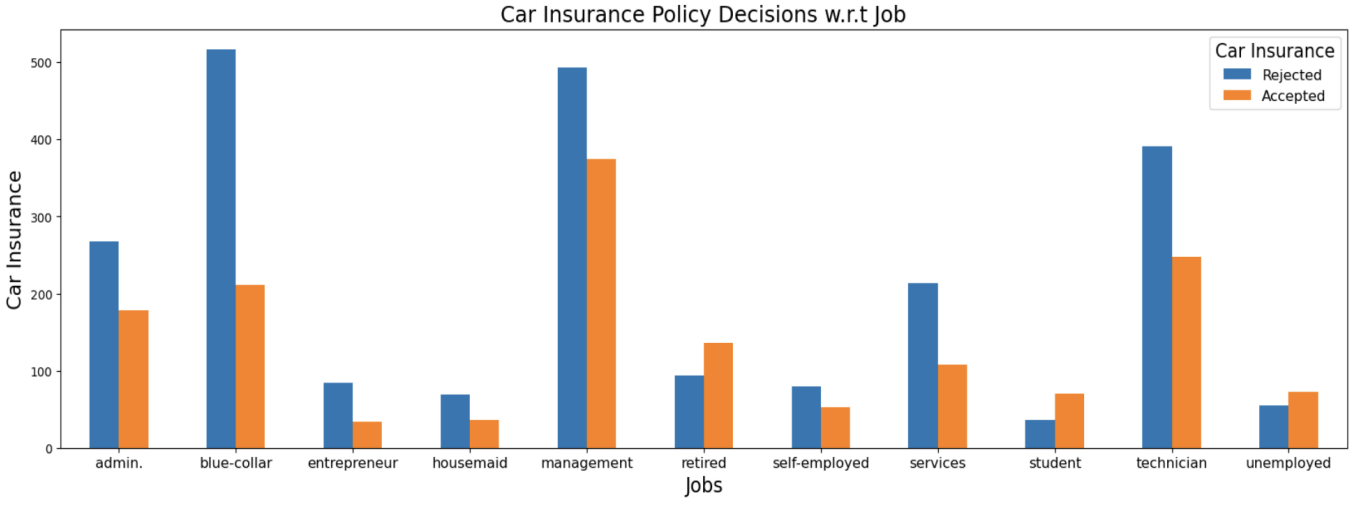
This bank has a majority of customers between the Age range of 25 to 30, that is Working Class.

- Marital Status of Customers.



More than 50% of Customers are married, so they might consider opting for car insurance.

- People having insurance VS their qualification

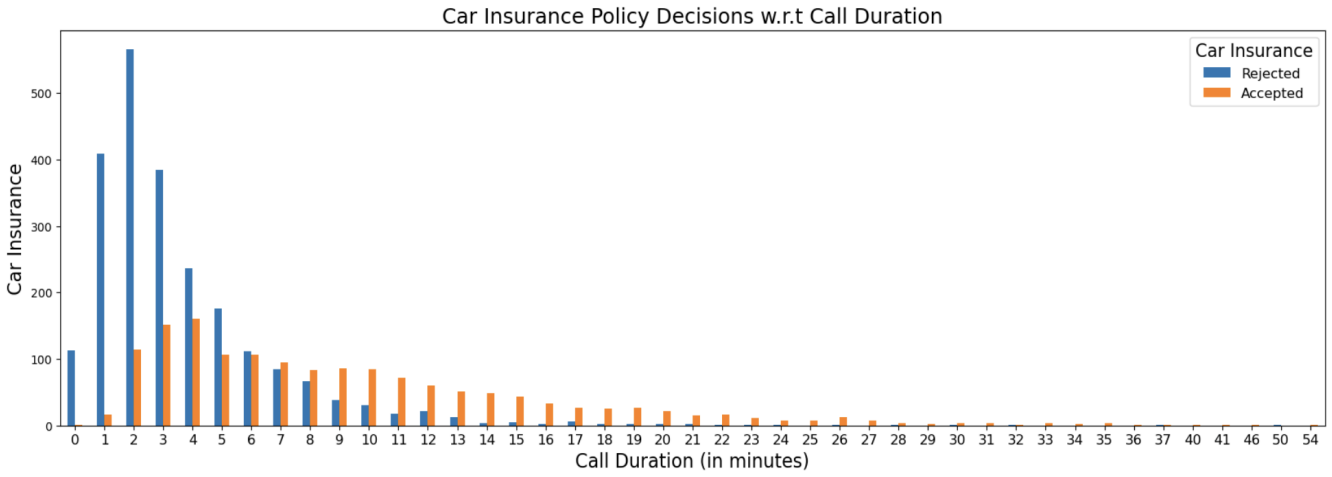


Here, we observed that most people who accepted the insurance proposal are decently qualified.

- People who bought Car Insurance & their Educational Qualification.

Car Insurance	0	1	Percentage Enrolled
Education			
primary	363	194	34.83
secondary	1256	727	36.66
tertiary	680	600	46.88

- People Opt for Insurance according to the time spent on call.



In this chart, we understood that as the call duration increases, the number of counts for Car Insurance accepted also increases.

Model Selection

As our project is about classification, we consider the below models.
i) K-Nearest Neighbour (KNN)
ii) Naive- Bayes
iii) Classification And Regression Tree (CART)
iv) Random Forest
v) Logistic Regression
vi) Neural Network
vii) Linear Discriminant Analysis (LDA)

However, in our dataset, we have both numerical and categorical variables. As KNN, Logistic Regression can handle only numerical variables, all categorical variables have to

assign numerical labels by using the module LabelEncoder in Sklearn Library, so we can apply all the models on the modified data frame. We can create m-1 dummies for m categories for each variable as well.

Applying every model is as below,

1) K-Nearest Neighbour (KNN)
   KNN chooses the nearest neighbor and assigns the responsibility of the nearest point to the test point. Before applying the KNN, all numerical values corresponding to numerical columns (Age, balance) columns are normalized.
   For K= 1, got training accuracy was 100%, and the testing accuracy was 73.16%, which means model overfits. After calculating the training_accuracy, testing_accuracy, and F1_score for changing the values of K from 1 to 14, we get below values

	K_values	Training_accuracy	Testing_accuracy	F1_score
10	11	0.791012	0.790576	0.710669
6	7	0.811955	0.787958	0.713781
9	10	0.788831	0.785995	0.688868
12	13	0.792757	0.785995	0.699725
8	9	0.801920	0.785340	0.706093
11	12	0.782286	0.785340	0.684615
13	14	0.781850	0.783377	0.681424
7	8	0.797120	0.780759	0.680038
4	5	0.814572	0.775524	0.695111
5	6	0.802356	0.763743	0.649174
2	3	0.856021	0.762435	0.679045
3	4	0.818935	0.751963	0.622134
1	2	0.854712	0.739529	0.581053
0	1	1.000000	0.731675	0.652542

After arranging the data in reverse order of Testing_accuracy, we found that for K=11, testing accuracy, and F1_score is highest i.e. 79.05% and 71.06% and there is no significant difference between training and testing accuracy. So, K=11 is the best fit.

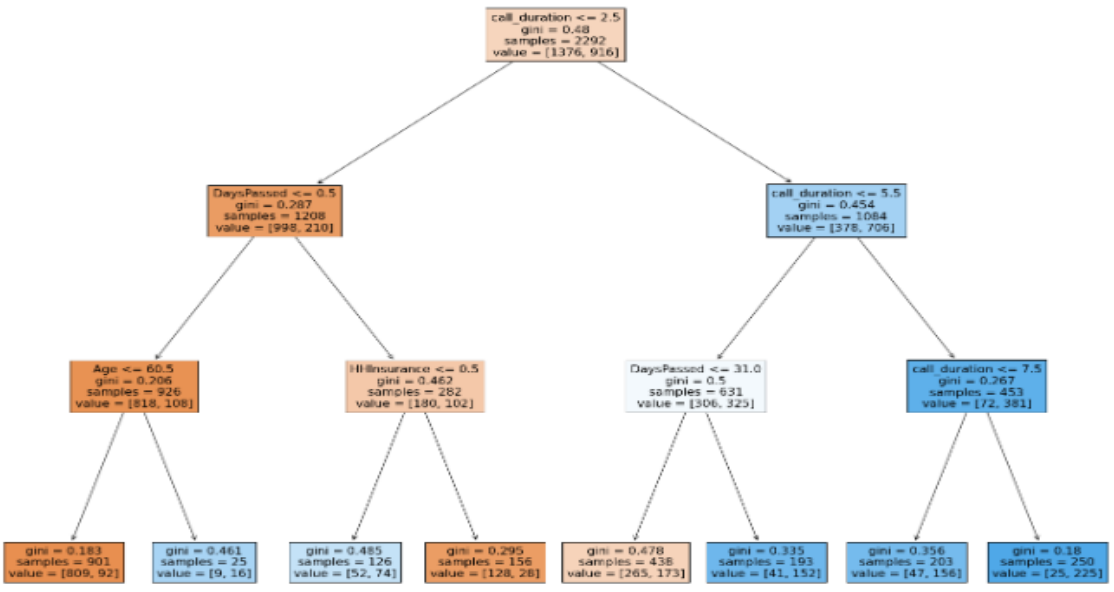
2) Naive Bayes
   Before applying the Naive Bayes, Age, and Balance, are numerical variables to be converted to categorical. So, the balance is divided by 2000 and rounded to the values. Also, Age is divided into 10 and made 10 categories.
   After applying Naive-Bayes, training and testing accuracy was 65% and 66% respectively, which is much lower.

3) CART
   Applying the CART, a fully grown tree had a maximum depth of 19 overfitted giving training accuracy of 100% and testing accuracy of 73%. So, in order to get

optimum values, we have to prune the tree. Max_depth is one of the most important hyperparameters while performing the pruning. So, model accuracy and F1 scores are calculated after training the model for different values of max_depth varying from 1 to 19.

	max_depth	Training_Acc	Testing_acc	F1_Score
0	1	0.743455	0.708115	0.660061
1	2	0.743455	0.708115	0.660061
2	3	0.796248	0.780759	0.703277
3	4	0.809773	0.776178	0.731132
4	5	0.823735	0.786649	0.741270
5	6	0.840750	0.789921	0.736237
6	7	0.860820	0.782068	0.706090
7	8	0.883944	0.783377	0.720675
8	9	0.905323	0.765707	0.701169
9	10	0.922339	0.761126	0.696592
10	11	0.941972	0.757199	0.695152
11	12	0.955934	0.736911	0.660473
12	13	0.972077	0.739529	0.666107
13	14	0.981675	0.733639	0.661116
14	15	0.991274	0.731021	0.665037
15	16	0.996510	0.719241	0.648649
16	17	0.998255	0.722513	0.653595
17	18	0.999564	0.723168	0.652424
18	19	1.000000	0.731675	0.667208

Comparing the accuracy and F1 score, we found that for max_depth = 3, there is no significant difference between the training and testing accuracy. So, max_depth =3 is chosen.



iv) Random Forest

Random forest gave the accuracy of 81% training and 79% testing accuracy.

v) Logistic Regression

Logistic regression tries to separate the data linearly. While applying the model, we consider various hyperparameters namely ‘penalty’, ‘tol’, and ‘solver’. After changing various hyperparameters we achieved training and testing accuracy of 79.93% and 77.95%.

After calculating the coefficients of every variable, we found that the most considerable parameter is call_duration and the least weighted parameter is a car loan.

	coeff
Age	0.010479
Job	0.035447
Marital	0.223417
Education	0.331586
Default	0.000000
Balance	-0.000004
HHInsurance	-1.211294
CarLoan	-0.652273
NoOfContacts	-0.112811
DaysPassed	0.002603
PrevAttempts	0.139391
call_duration	0.555977

vi) Neural Network

After fitting the model into Neural Network, we get a testing accuracy of 79.97%. In order to see if accuracy is changing with increasing the hidden layer, we considered hidden layers up to 5. Accuracy and F1 scores are as below,

	Hidden_layers	Training_acc	Teating_acc	F1_score
0	1	0.795812	0.799738	0.747941
1	2	0.737347	0.750654	0.711145
2	3	0.750873	0.753272	0.726216
3	4	0.801483	0.795812	0.748387
4	5	0.754799	0.756545	0.722802

After comparing the accuracy and F1 scores, we found that there is no increase in the accuracy score. This means, there is no complex structure in data presentation and is linearly separable. However, accuracy is slightly increased as compared to other models trained.

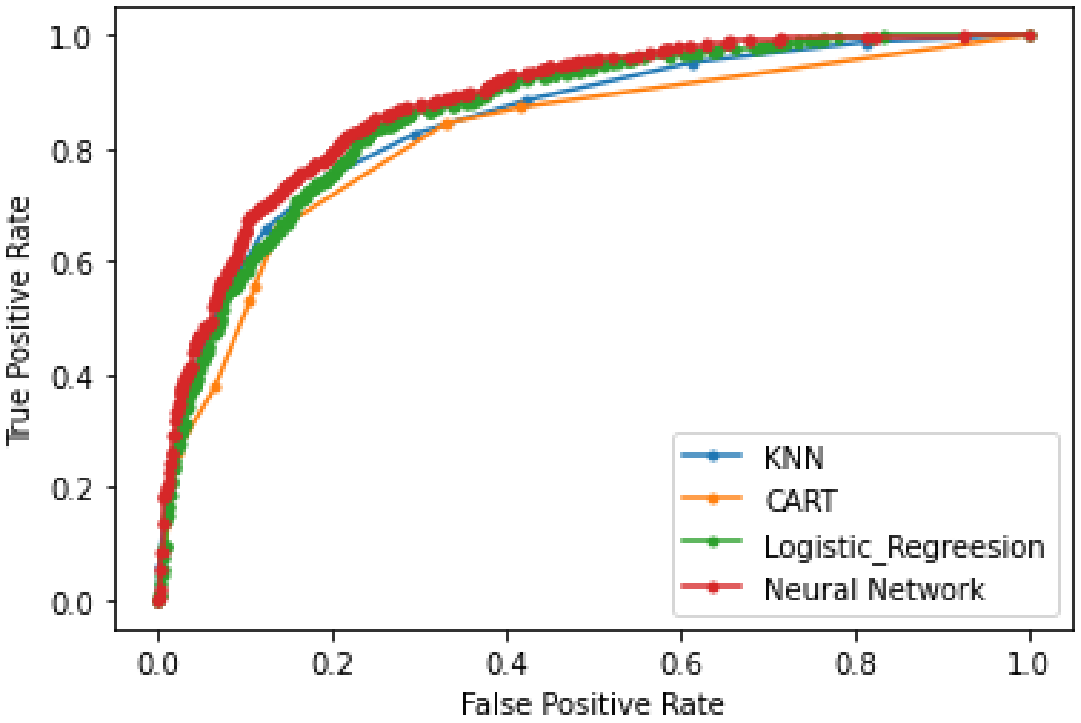
vii) Linear Discriminant Analysis (LDA)
After applying the LDA, training_accuracy, testing accuracy and F1_score achieved was 77.87%, 77.15%, and 67.53% respectively.

PERFORMANCE EVALUATION

To compare all the models together, we get the below results.

Model	Training Acc	Testing Acc	F1_score
KNN	0.791	0.79	0.7106
Naïve-Bayes	0.6588	0.6623	0.4901
CART	0.7962	0.7807	0.7032
Random Forest	0.877	0.7984	0.7433
Logistic Regression	0.7997	0.7808	0.7016
Neural Network	0.8014	0.7958	0.7483
LDA	0.8014	0.77	0.6753

ROC curves for best-performing models are as below,



After comparing model parameters and ROC curves Neural Network, Logistic Regression, KNN, and CART are the best-performing models.

Discussion and Recommendation

After applying all the models, the best accuracy we get is with Neural Network. Neural Network can catch complex structures in data, so further hyperparameter tuning can increase the accuracy of the model. KNN is performing better than CART, but KNN is a lazy learner and can take a lot more time while predicting the result if the testing dataset is large enough. Whereas, in CART and Logistic Regression, once the model is trained, it took lesser time as compared with KNN.

Summary

In this study, we implement various supervised machine learning models to perform binary classification. We conclude that Neural Network is the best-performing model among all the models we applied.