Project Phase 2

TRAINING , TESTING AND VALIDATION OF A PREDICTION MODEL

Mohammed Zeeshan Ali
Yachen Chang
Shreya Chauhan
Shivangi Jagdishkumar Bhavsar

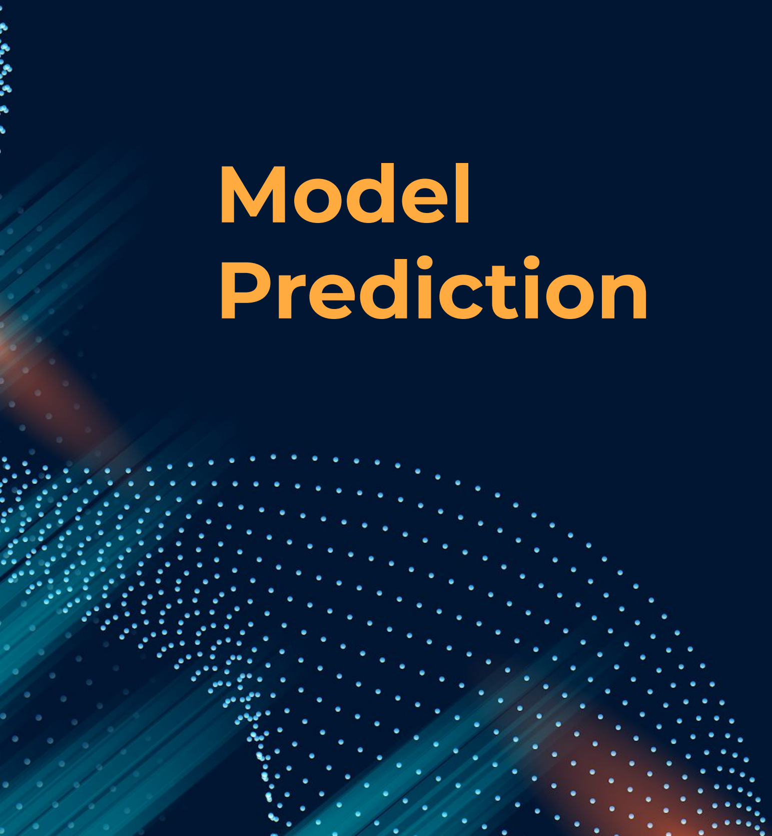
PHASE - 1 Conclusion

- EDA performed on selected variables
- Independent Variable - Attrition Flag
- Variables Selected in Phase - 1 - Gender , Income Category , Education Level , Credit Limit , Average Utilisation Ratio , Customer Age and Total Revolving Balance.

Phase - 2

Model Prediction

- Logistic Regression
- Naive Bayes
- MLP Classifier
- Decision Tree
- Random Forest
- K-Nearest Neighbours



Ratio Validation Set

20%

TESTING

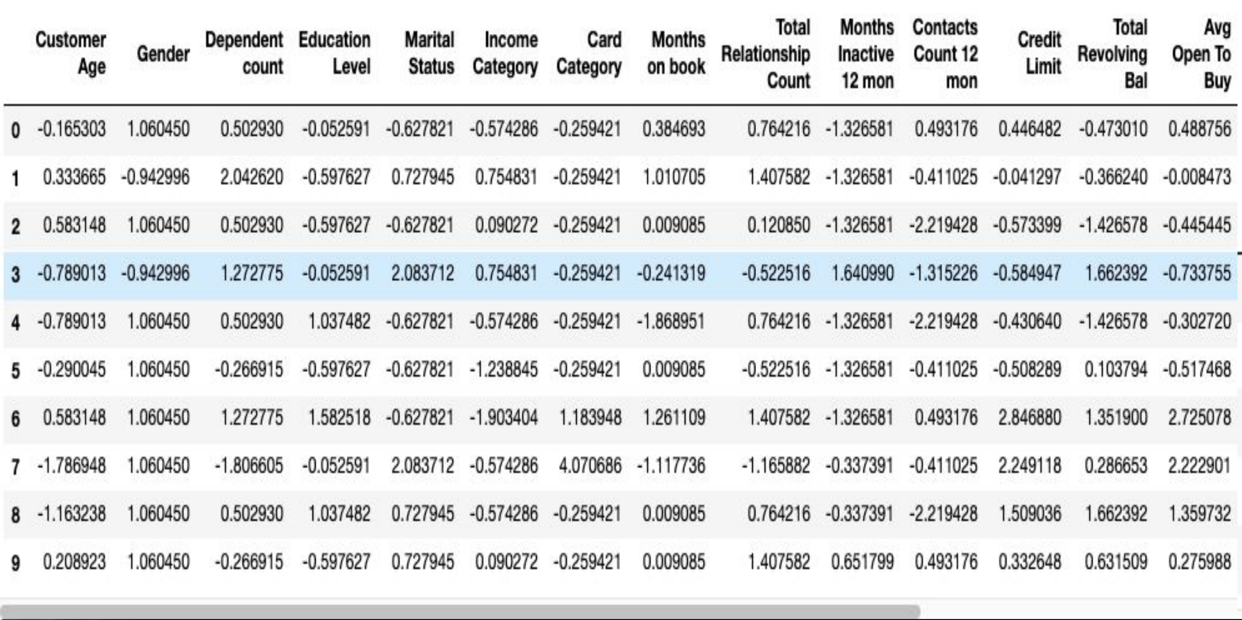
70%

TRAINING

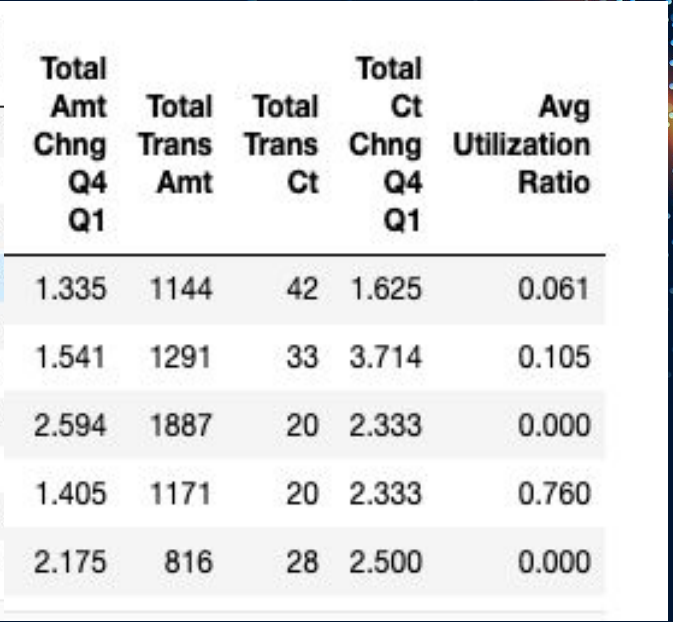
10%

VALIDATION

MOHAMMED ZEESHAN ALI



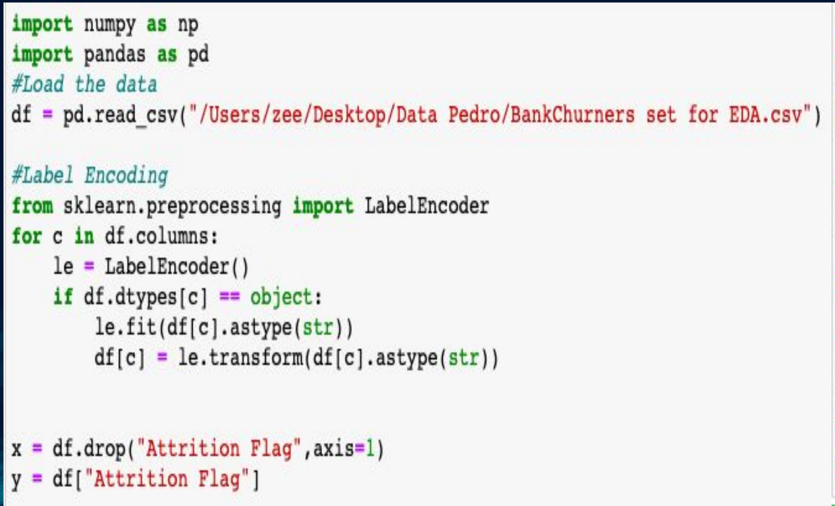
	Customer Age	Gender	Dependent count	Education Level	Marital Status	Income Category	Card Category	Months on book	Total Relationship Count	Months Inactive 12 mon	Contacts Count 12 mon	Credit Limit	Total Revolving Bal	Avg Open To Buy
0	-0.165303	1.060450	0.502930	-0.052591	-0.627821	-0.574286	-0.259421	0.384693	0.764216	-1.326581	0.493176	0.446482	-0.473010	0.488756
1	0.333665	-0.942996	2.042620	-0.597627	0.727945	0.754831	-0.259421	1.010705	1.407582	-1.326581	-0.411025	-0.041297	-0.366240	-0.008473
2	0.583148	1.060450	0.502930	-0.597627	-0.627821	0.090272	-0.259421	0.009085	0.120850	-1.326581	-2.219428	-0.573399	-1.426578	-0.445445
3	-0.789013	-0.942996	1.272775	-0.052591	2.083712	0.754831	-0.259421	-0.241319	-0.522516	1.640990	-1.315226	-0.584947	1.662392	-0.733755
4	-0.789013	1.060450	0.502930	1.037482	-0.627821	-0.574286	-0.259421	-1.868951	0.764216	-1.326581	-2.219428	-0.430640	-1.426578	-0.302720
5	-0.290045	1.060450	-0.266915	-0.597627	-0.627821	-1.238845	-0.259421	0.009085	-0.522516	-1.326581	-0.411025	-0.508289	0.103794	-0.517468
6	0.583148	1.060450	1.272775	1.582518	-0.627821	-1.903404	1.183948	1.261109	1.407582	-1.326581	0.493176	2.846880	1.351900	2.725078
7	-1.786948	1.060450	-1.806605	-0.052591	2.083712	-0.574286	4.070686	-1.117736	-1.165882	-0.337391	-0.411025	2.249118	0.286653	2.222901
8	-1.163238	1.060450	0.502930	1.037482	0.727945	-0.574286	-0.259421	0.009085	0.764216	-0.337391	-2.219428	1.509036	1.662392	1.359732
9	0.208923	1.060450	-0.266915	-0.597627	0.727945	0.090272	-0.259421	0.009085	1.407582	0.651799	0.493176	0.332648	0.631509	0.275988



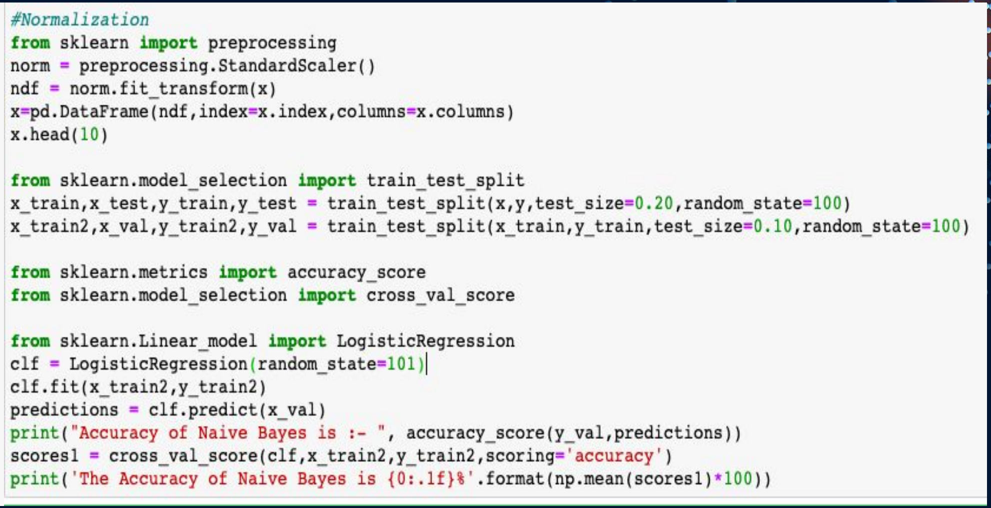
Total Amt Chng Q4 Q1	Total Trans Amt	Total Trans Ct	Total Ct Chng Q4 Q1	Avg Utilization Ratio
1.335	1144	42	1.625	0.061
1.541	1291	33	3.714	0.105
2.594	1887	20	2.333	0.000
1.405	1171	20	2.333	0.760
2.175	816	28	2.500	0.000

MOHAMMED ZEESHAN ALI

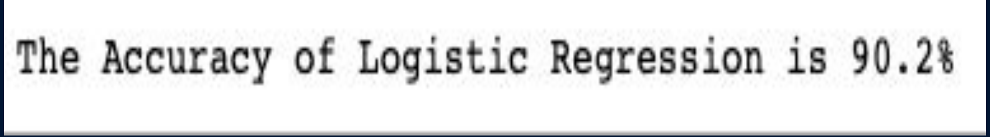
Logistic Regression



```
import numpy as np
import pandas as pd
#Load the data
df = pd.read_csv("/Users/zee/Desktop/Data Pedro/BankChurners set for EDA.csv")

#Label Encoding
from sklearn.preprocessing import LabelEncoder
for c in df.columns:
    le = LabelEncoder()
    if df.dtypes[c] == object:
        le.fit(df[c].astype(str))
        df[c] = le.transform(df[c].astype(str))


x = df.drop("Attrition Flag",axis=1)
y = df["Attrition Flag"]
```

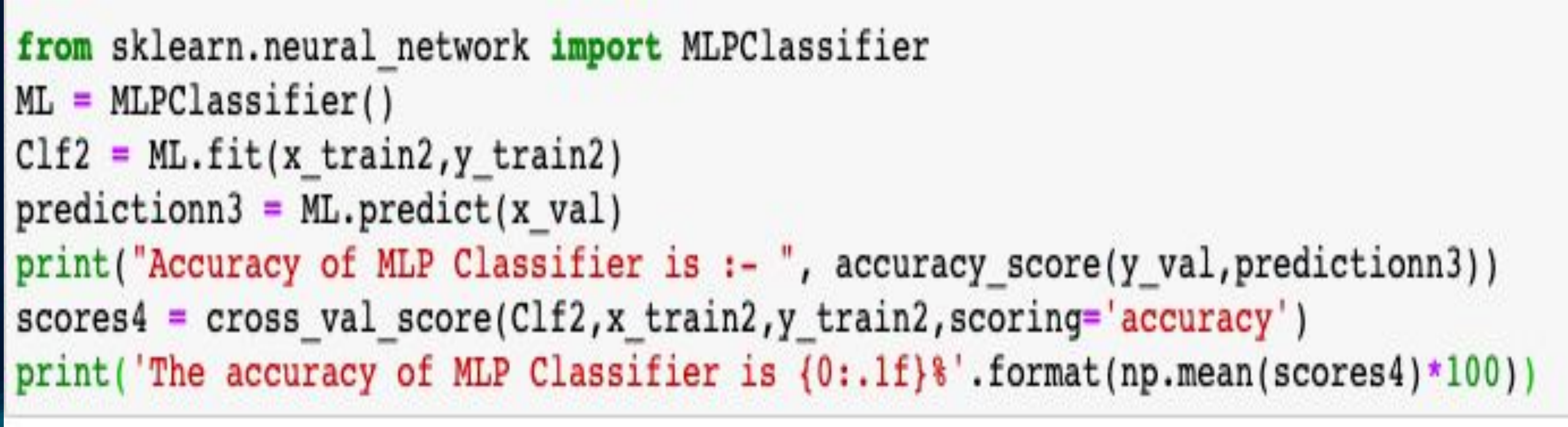


```
#Normalization
from sklearn import preprocessing
norm = preprocessing.StandardScaler()
ndf = norm.fit_transform(x)
x=pd.DataFrame(ndf,index=x.index,columns=x.columns)
x.head(10)

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20,random_state=100)
x_train2,x_val,y_train2,y_val = train_test_split(x_train,y_train,test_size=0.10,random_state=100)

from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score

from sklearn.Linear_model import LogisticRegression
clf = LogisticRegression(random_state=101)
clf.fit(x_train2,y_train2)
predictions = clf.predict(x_val)
print("Accuracy of Naive Bayes is :- ", accuracy_score(y_val,predictions))
scores1 = cross_val_score(clf,x_train2,y_train2,scoring='accuracy')
print('The Accuracy of Naive Bayes is {0:.1f}%'.format(np.mean(scores1)*100))
```

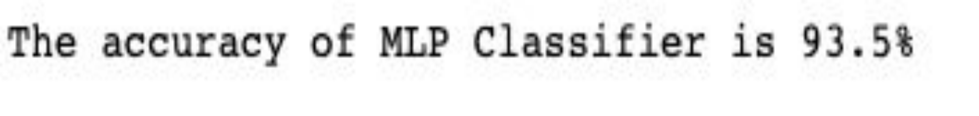


```
The Accuracy of Logistic Regression is 90.2%
```

MLP Classifier



```
from sklearn.neural_network import MLPClassifier
ML = MLPClassifier()
Clf2 = ML.fit(x_train2,y_train2)
predictionn3 = ML.predict(x_val)
print("Accuracy of MLP Classifier is :- ", accuracy_score(y_val,predictionn3))
scores4 = cross_val_score(Clf2,x_train2,y_train2,scoring='accuracy')
print('The accuracy of MLP Classifier is {0:.1f}%'.format(np.mean(scores4)*100))
```



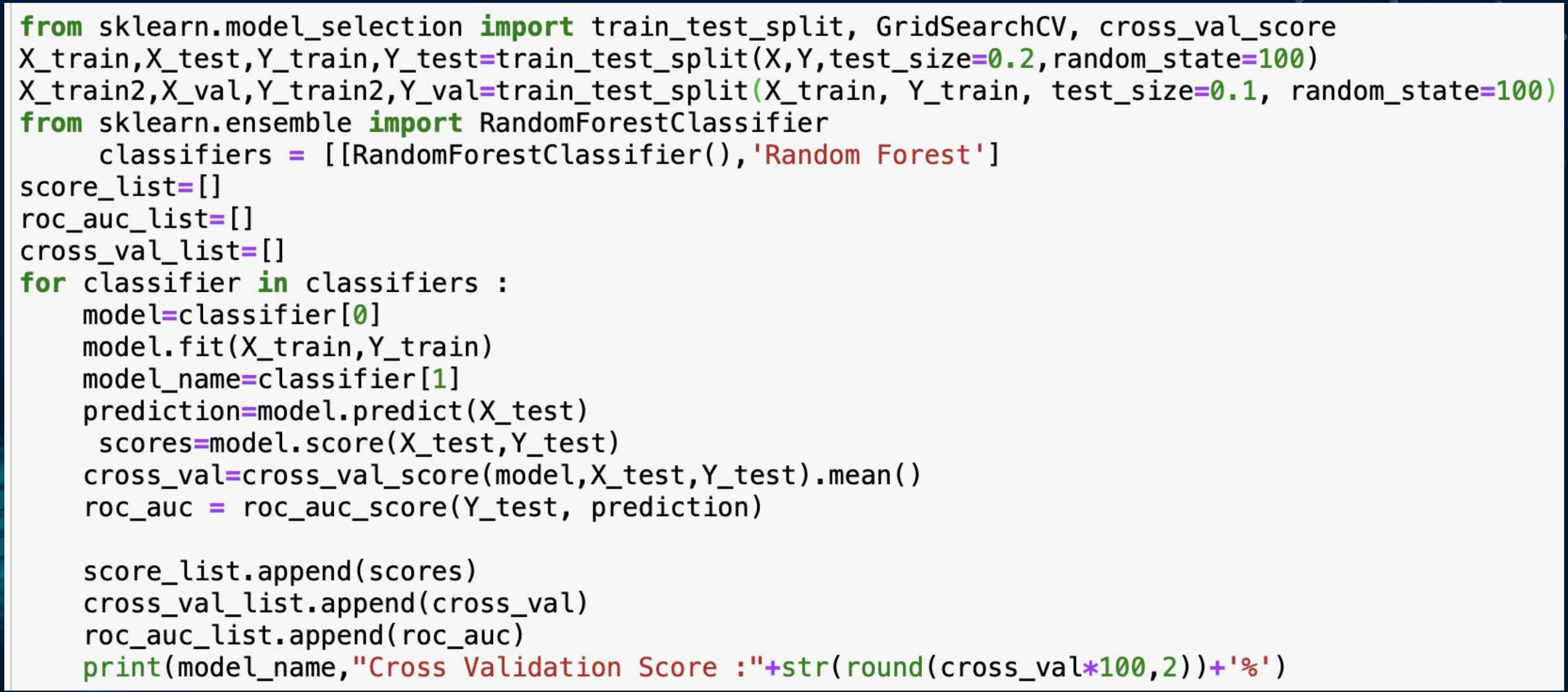
```
The accuracy of MLP Classifier is 93.5%
```

ANALYSIS

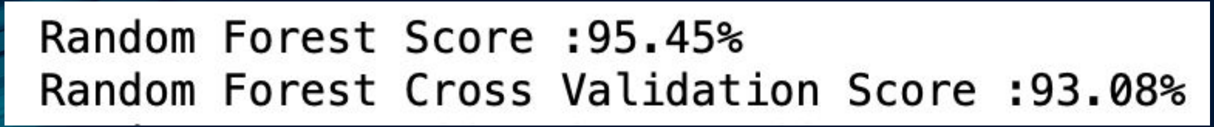
- Neural Network’s MLP Classifier is more accurate than Logistic Regression as a model for prediction

- The accuracies obtained are best explained with the fact that there are strong relationships between different variables in the given dataset

Random Forest



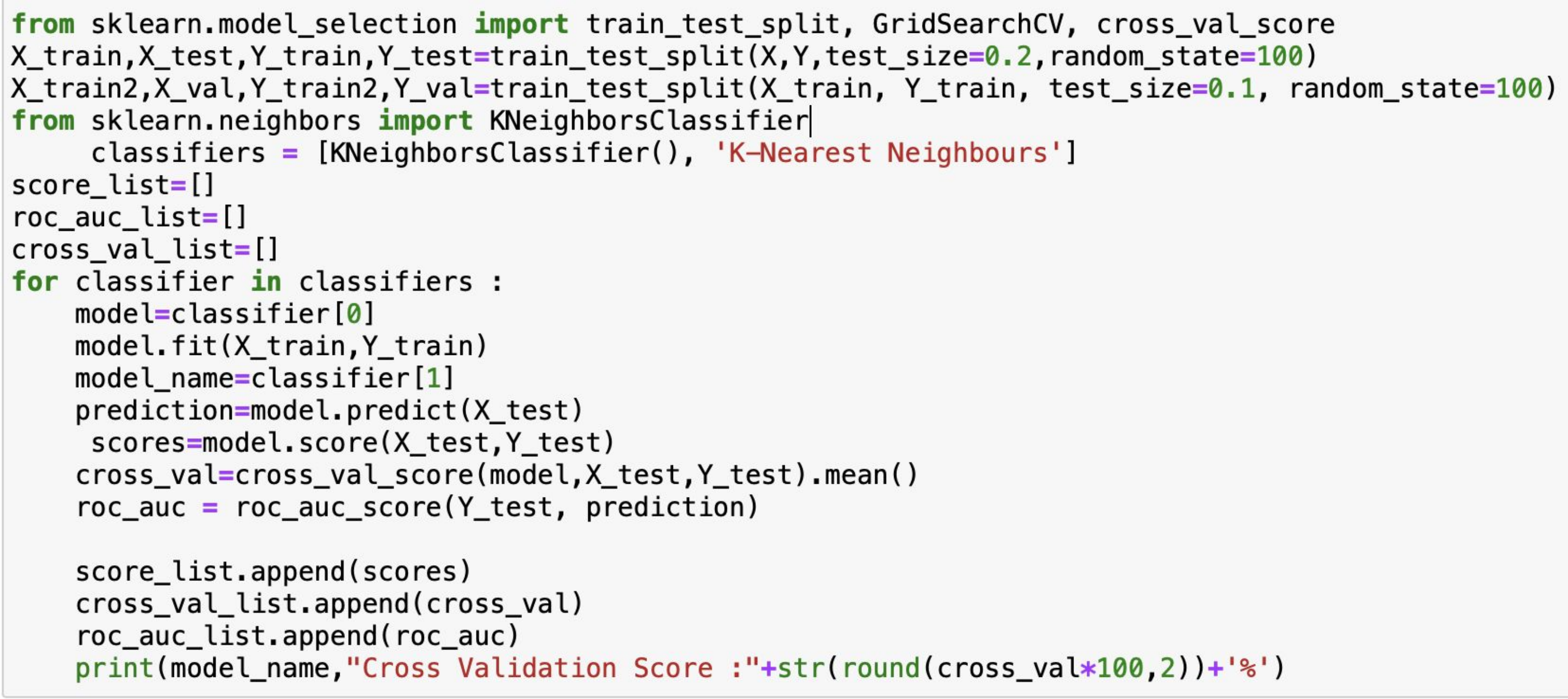
```
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=100)
X_train2,X_val,Y_train2,Y_val=train_test_split(X_train, Y_train, test_size=0.1, random_state=100)
from sklearn.ensemble import RandomForestClassifier
     classifiers = [[RandomForestClassifier(),'Random Forest']
score_list=[]
roc_auc_list=[]
cross_val_list=[]
for classifier in classifiers :
    model=classifier[0]
    model.fit(X_train,Y_train)
    model_name=classifier[1]
    prediction=model.predict(X_test)
     scores=model.score(X_test,Y_test)
    cross_val=cross_val_score(model,X_test,Y_test).mean()
    roc_auc = roc_auc_score(Y_test, prediction)

    score_list.append(scores)
    cross_val_list.append(cross_val)
    roc_auc_list.append(roc_auc)
    print(model_name,"Cross Validation Score :"+str(round(cross_val*100,2))+'%')
```



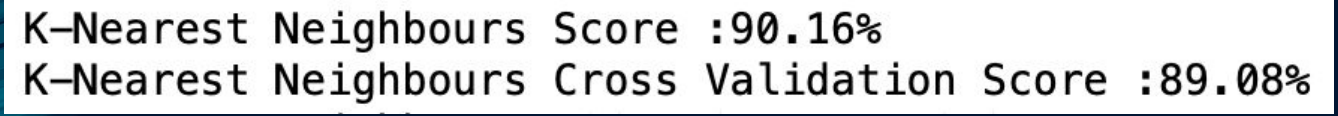
```
Random Forest Score :95.45%
Random Forest Cross Validation Score :93.08%
```

Yachen Chang

K-Nearest Neighbours



```
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=100)
X_train2,X_val,Y_train2,Y_val=train_test_split(X_train, Y_train, test_size=0.1, random_state=100)
from sklearn.neighbors import KNeighborsClassifier
     classifiers = [KNeighborsClassifier(), 'K-Nearest Neighbours']
score_list=[]
roc_auc_list=[]
cross_val_list=[]
for classifier in classifiers :
    model=classifier[0]
    model.fit(X_train,Y_train)
    model_name=classifier[1]
    prediction=model.predict(X_test)
     scores=model.score(X_test,Y_test)
    cross_val=cross_val_score(model,X_test,Y_test).mean()
    roc_auc = roc_auc_score(Y_test, prediction)

    score_list.append(scores)
    cross_val_list.append(cross_val)
    roc_auc_list.append(roc_auc)
    print(model_name,"Cross Validation Score :"+str(round(cross_val*100,2))+'%')
```



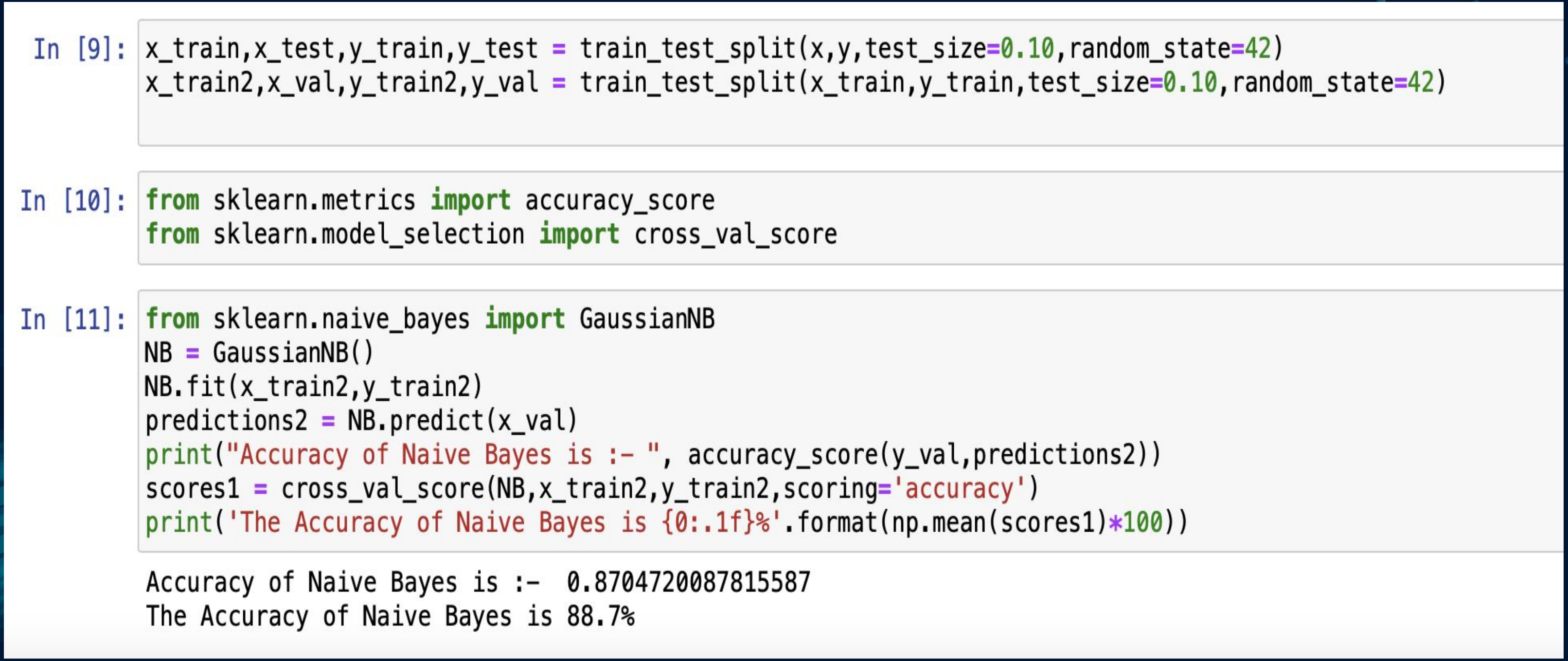
```
K-Nearest Neighbours Score :90.16%
K-Nearest Neighbours Cross Validation Score :89.08%
```

Yachen Chang

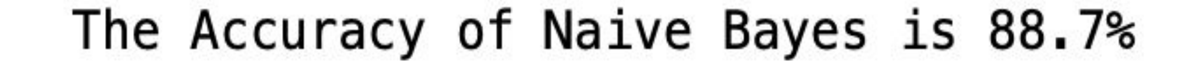
Summary

- Random forest classifier is more accuracy than K-Nearest Neighbours
- The models without applying cross validation are all higher than the validated models

Naive Bayes

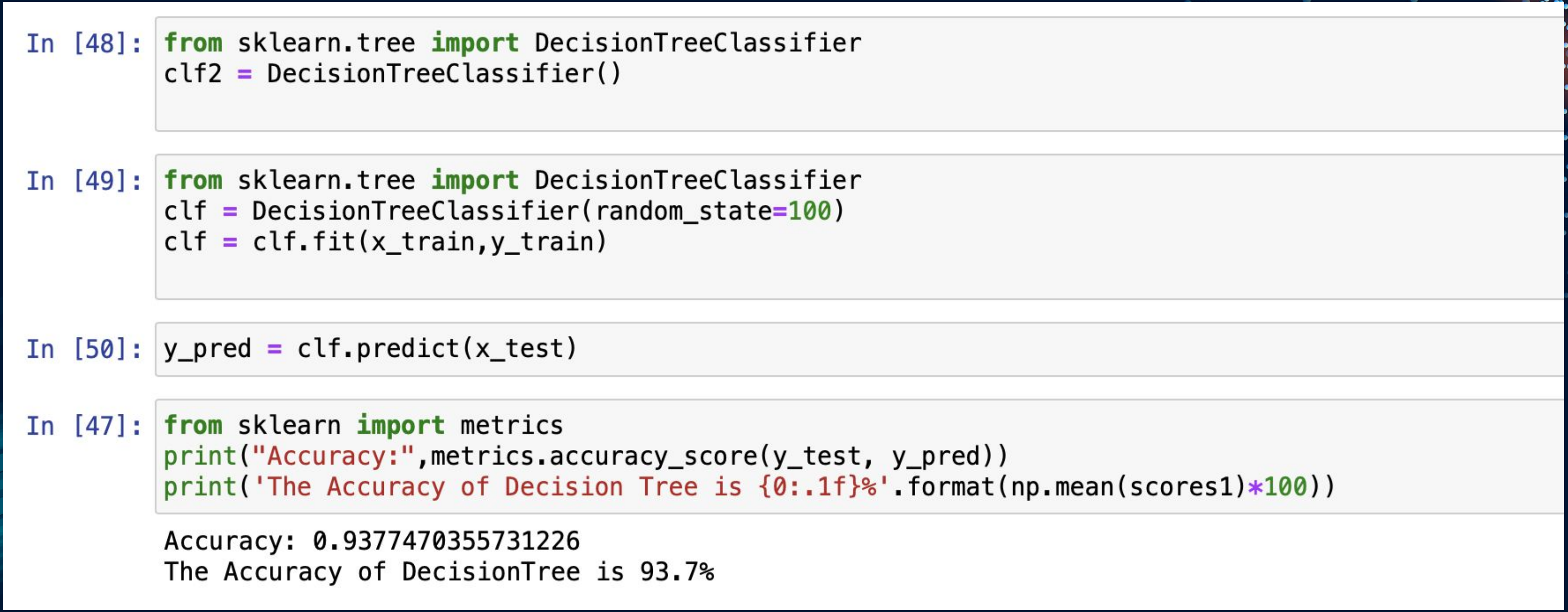


```
In [9]:  x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.10,random_state=42)
          x_train2,x_val,y_train2,y_val = train_test_split(x_train,y_train,test_size=0.10,random_state=42)

In [10]: from sklearn.metrics import accuracy_score
          from sklearn.model_selection import cross_val_score

In [11]: from sklearn.naive_bayes import GaussianNB
          NB = GaussianNB()
          NB.fit(x_train2,y_train2)
          predictions2 = NB.predict(x_val)
          print("Accuracy of Naive Bayes is :- ", accuracy_score(y_val,predictions2))
          scores1 = cross_val_score(NB,x_train2,y_train2,scoring='accuracy')
          print('The Accuracy of Naive Bayes is {0:.1f}%'.format(np.mean(scores1)*100))

          Accuracy of Naive Bayes is :-  0.8704720087815587
          The Accuracy of Naive Bayes is 88.7%
```



```
The Accuracy of Naive Bayes is 88.7%
```

Decision Tree



```
In [48]: from sklearn.tree import DecisionTreeClassifier
         clf2 = DecisionTreeClassifier()

In [49]: from sklearn.tree import DecisionTreeClassifier
         clf = DecisionTreeClassifier(random_state=100)
         clf = clf.fit(x_train,y_train)

In [50]: y_pred = clf.predict(x_test)

In [47]: from sklearn import metrics
         print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         print('The Accuracy of Decision Tree is {0:.1f}%'.format(np.mean(scores1)*100))

         Accuracy: 0.9377470355731226
         The Accuracy of DecisionTree is 93.7%
```

The Accuracy of DecisionTree is 93.7%

Summary

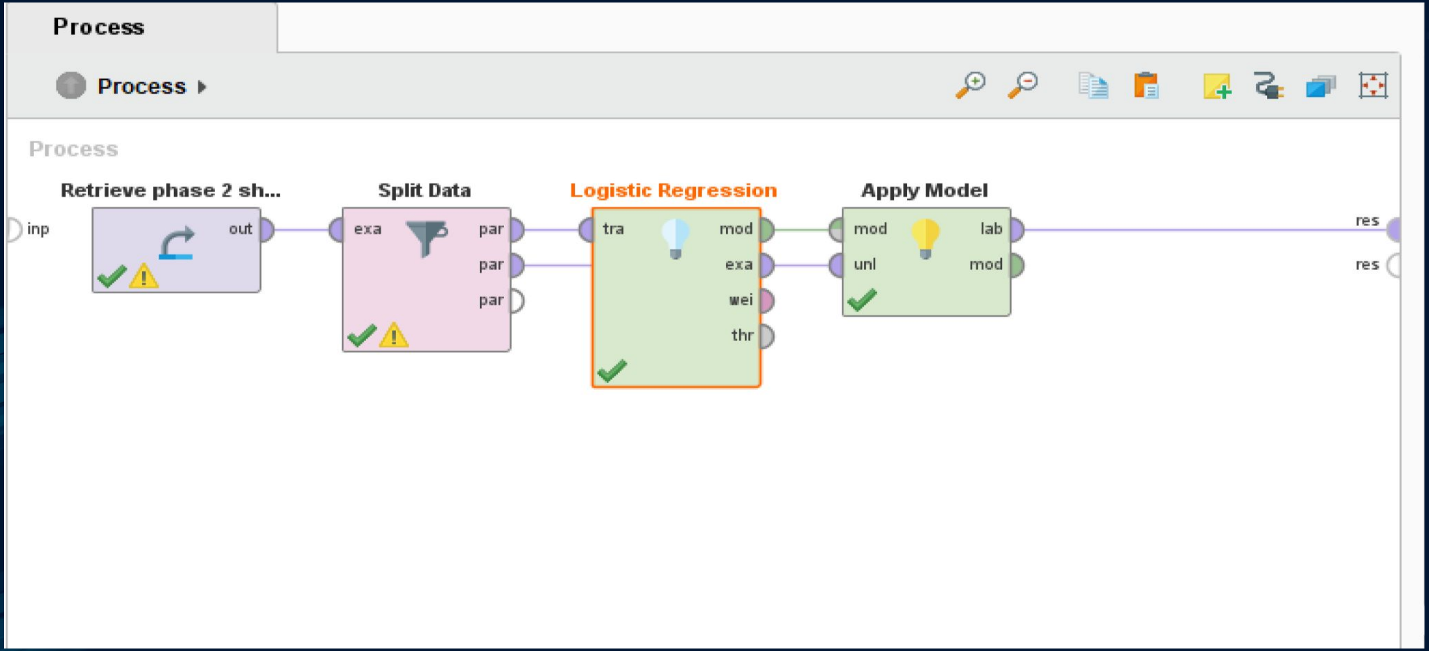
- Decision Tree Model is showing more accuracy than Naive Bayes model

- Decision Tree Model is so far the best model for prediction that is 93. 7% accuracy

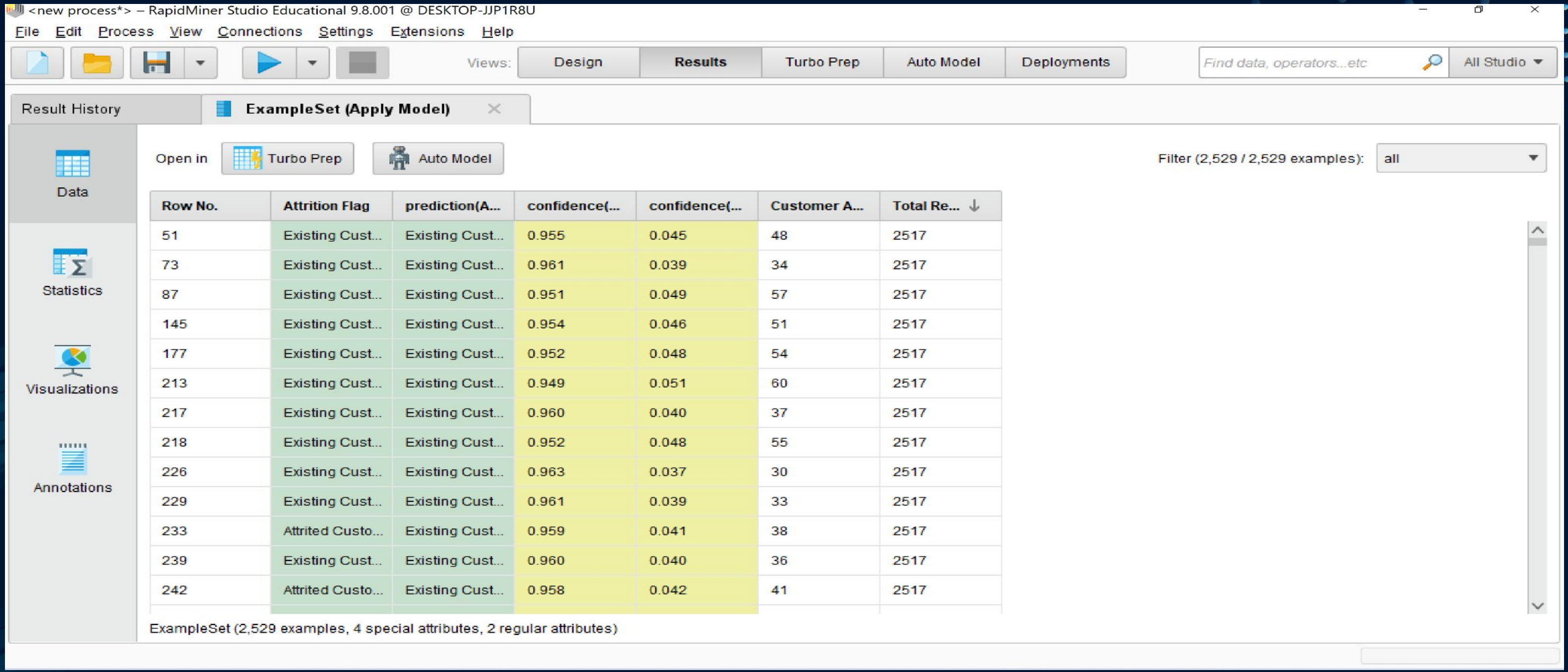
Shivangi J Bhavsar

Logistic regression for,

Attributes such, Customer age , Attrition flag , Total revolving balance



Result set of Logistics Regression process


<new process*> – RapidMiner Studio Educational 9.8.001 @ DESKTOP-JJP1R8U
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
Find data, operators...etc
All Studio
Result History
ExampleSet (Apply Model)
Data
Statistics
Visualizations
Annotations
Open in Turbo Prep Auto Model

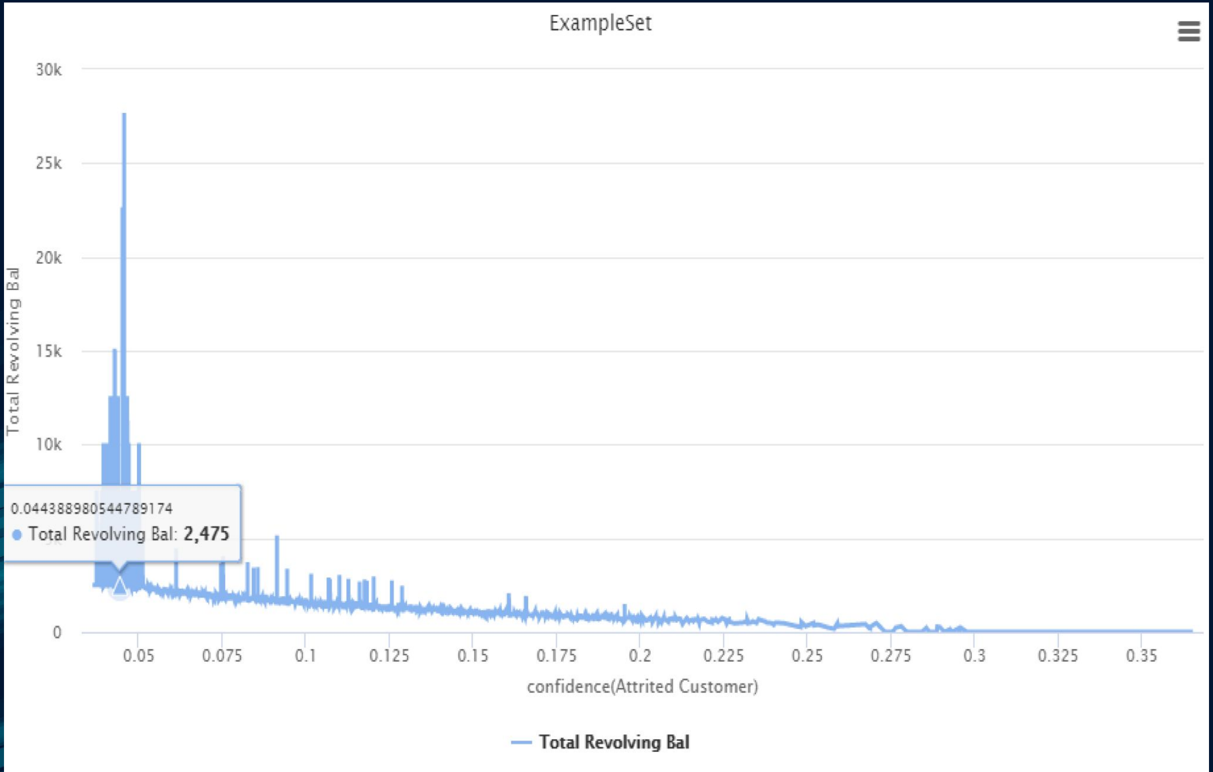
Filter (2,529 / 2,529 examples): all

Row No.	Attrition Flag	prediction(A...	confidence(...	confidence(...	Customer A...	Total Re... ↓
51	Existing Cust...	Existing Cust...	0.955	0.045	48	2517
73	Existing Cust...	Existing Cust...	0.961	0.039	34	2517
87	Existing Cust...	Existing Cust...	0.951	0.049	57	2517
145	Existing Cust...	Existing Cust...	0.954	0.046	51	2517
177	Existing Cust...	Existing Cust...	0.952	0.048	54	2517
213	Existing Cust...	Existing Cust...	0.949	0.051	60	2517
217	Existing Cust...	Existing Cust...	0.960	0.040	37	2517
218	Existing Cust...	Existing Cust...	0.952	0.048	55	2517
226	Existing Cust...	Existing Cust...	0.963	0.037	30	2517
229	Existing Cust...	Existing Cust...	0.961	0.039	33	2517
233	Attrited Custo...	Existing Cust...	0.959	0.041	38	2517
239	Existing Cust...	Existing Cust...	0.960	0.040	36	2517
242	Attrited Custo...	Existing Cust...	0.958	0.042	41	2517

ExampleSet (2,529 examples, 4 special attributes, 2 regular attributes)

Shivangi J Bhavsar

The significance between attrited customer and revolving balance .



One the confidence interval of .05% , fro the attitated customer the highest revolving balance is 2475 $, which is on higher side that indicates that, the one who has not capability to repay credit card bills , are being churned from bank customer list .

Shivangi J Bhavsar

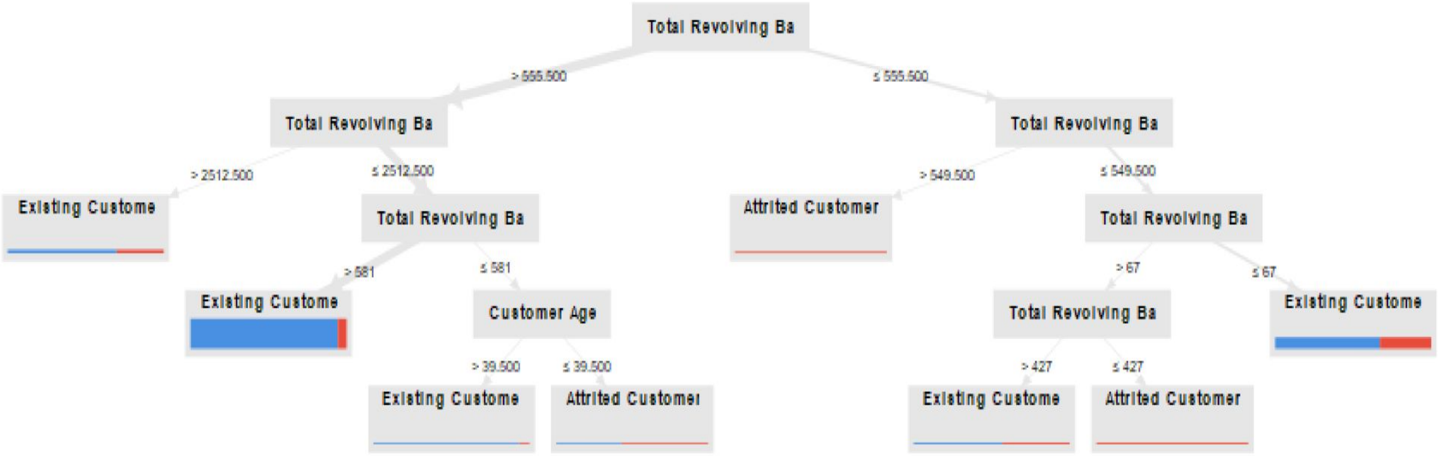
Analysis result for linear regression

- After checking the accuracy of model , the data was split in the ratio of 6:2 for training and testing purpose .

- From the result of examples set , the avg confidence of existing customer is .837 where for attrited customer .163

- Where as average customer age is 46 and revolvning balance is 1157 $. Though the attitation in example data set is nearly 15 % from entire tested data set . which clearly indicates that , revolving balance is logically related with the churning of customer

Decision tree for attributes



Analysis conclusion from decision tree algorithm :

When , total revolving balance is above $ 2000 the percentage of attrited customer is higher in compare to lower balance .

While total revolving balance is less then $ 100 , the attrition ratio is half the total number of existing customers for such balance .

Summary

The total revolving balance has direct association with churning rate .

As more and more customer are taking the option of goig into revolving balance , they are affecting their credit score and that ultimately , makes them to leave customer

There for to avoid churning due to such factors , bank should focus on the interest rate and paying capability of customers .

Accuracy Results

Logistic Regression	90.2%
Naive Bayes	88.7%
MLP Classifier	93.5%
Random Forest	92.93%
K-Nearest Neighbours	89.08%
Decision Tree	93.7%

Decision Tree

The Best Model

THANKS!