FLIP ROBO

Micro Credit defaulter Model

Submitted by:

NIHAL SINGH

ACKNOWLEDGMENT

This Project includes the details about Microfinance services (MFI) that offers financial services to the low-income populations especially the unbanked poor families living in remote areas with not so much source of income.

MFI provide the micro-credit on mobile balances and that to be paid back in 5 days. The micro-credit Defaulter model basically used to predict the client will paying back the loaned amount within the time duration of 5 days. It is a classification model as the target column has only two outcome.

INTRODUCTION

- Business Problem Framing

  The micro credit model is related to the financial business problem.it is basically related to consumers who has taken the micro-credit loan on mobile balance and to be paid back in 5 days.

- Conceptual Background of the Domain Problem

  The model is basically related to the financial domain and consist of loan problem provided to the customers and that should be paid back in 5 days on the date of issue. It is basically consist of loan that is provided to customer and paid back with some interest to the company.

- Review of Literature

  MFIs are organisations such as credit unions, downscaled commercial banks, and financial cooperatives that provide financial services to the poor.

  Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans.

  Microfinance charge high-interest rates to cover high administrative incurred in the process of financial service delivery microenterprises. The huge volume of transactions, micro amounts involved and “high touch” in financial service delivery makes it more expensive.

- Motivation for the Problem Undertaken

  The micro credit loan model consist of large amount of data which help to analyse and predict the customer pay back the loaned amount within 5 days of insurance loan.

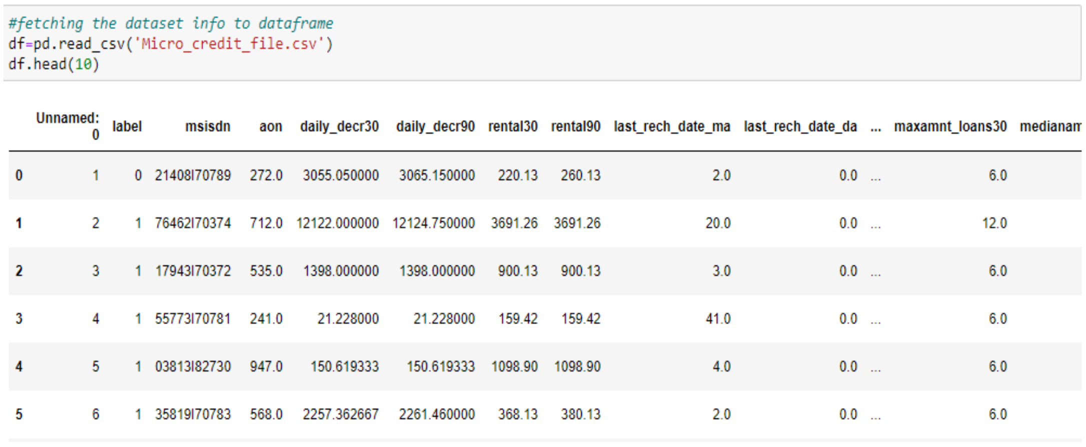
Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The Dataset is in csv format which consist of all details i.e rows and columns. Analysing the data, there data type, which columns is impotent. Plotting the multiple plots like bar plot, dist plot, box plot ,heat map to analyse the data and which features are important and related to the target and delete the non-important features. Using z-score and iqr method to find the outliers and deleting it. Scaling the data to build model.

- Data Sources and their formats

  Data is in the CSV format.the data has been imported into the data frame using the python library.



```
#fetching the dataset info to dataframe
df=pd.read_csv('Micro_credit_file.csv')
df.head(10)
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	medianam
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	
5	6	1	35819I70783	568.0	2257.362667	2261.460000	368.13	380.13	2.0	0.0	...	6.0	

The data having the different datatypes i.e. int, and object.

- ## Data Preprocessing Done

  The Data Preprocessing consist of below mentioned points:

  1. Checking the null values form the dataset, if null value is available them replace them with the suitable values.
  2. Describing the data.
  3. Using the encoding method to encode the categorical data into the numerical data.
  4. Checking all the features and checking the impact with target if no impact is there then deleting the features.
  5. Plotting the multiple plots like bar plot, dist plot, box plot to check the data distribution and outliers from the datasets.
  6. Using IQR method to check the outliers and remove the outliers from the data frame.
  7. Checking the skewness after removing the outliers from the dataset and if skewness is there then remove it using multiple method.
  8. Using the heat map and variance inflation factor to check the multicollliniarity issue between the features.
  9. If issue is there then delete the feature which having such issue.
  10. Scaling the data using the standard scalar method.
  11. Separating the features and target to build the model.

- ## State the set of assumptions (if any) related to the problem under consideration

  Deleted the multiple columns based on the analysis which does not having any relation with target and the final outcome of the model.

- ## Data Inputs- Logic- Output Relationships

  There are multiple features present in the data frame. In all the features some are having impact on the final outcome and some features are not useful. Like in the data frame Unnamned: 0, msisdn features shows no impact on the target and final outcome. There are multiple features which impact the target and the final models.

- Hardware and Software Requirements and Tools Used

  The Tool used to build the model is anaconda jupyter.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Multiple problem faced during the model building. All are listed bellows.

  1. Checking for the unwanted columns analyse the relation with target and removal of that features.
  2. Checking the null values and replace it.
  3. Checking of zeros and replace it.
  4. Managing the categorical columns using multiple methods.
  5. Removal of outliers form the features using the IQR.
  6. Removal of multicolliniarity issue from the features.

- Testing of Identified Approaches (Algorithms)

  This is a classification problem so multiple classification model has been used for the model building and basically training and testing the data on multiple models to select the best model.

  The multiple models are used as given below:

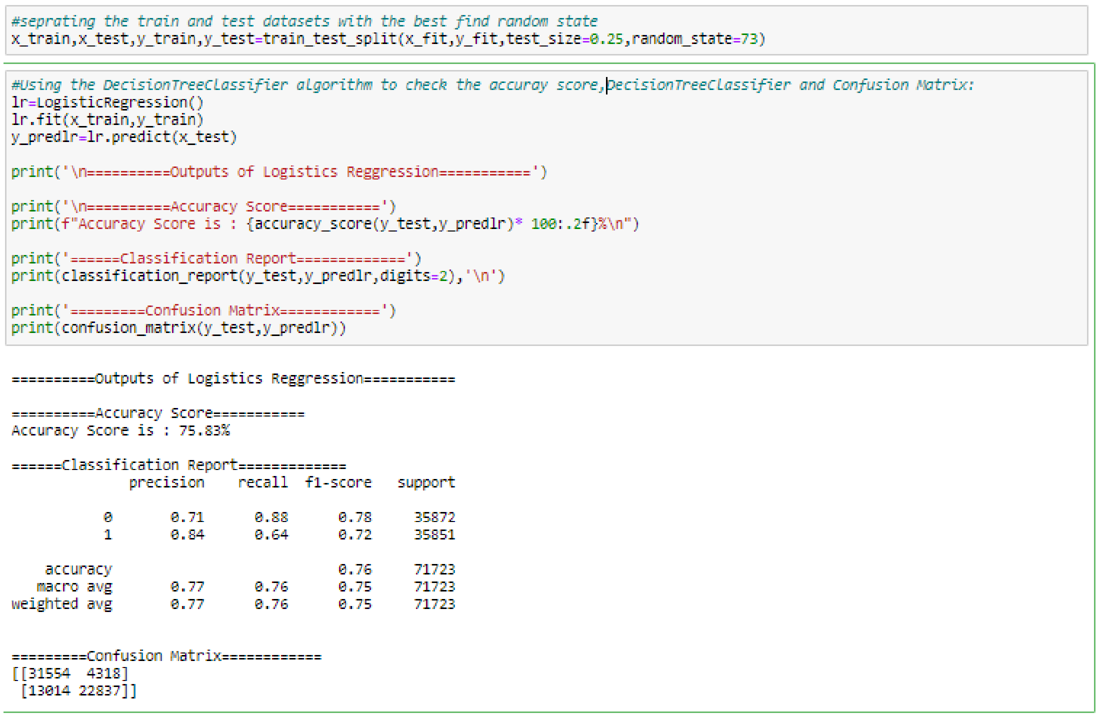
  1. Logistics Regression
  2. Decision Tree classifier
  3. Ada boost classifier
  4. Random forest classifier
  5. KNN Classifiers

- Run and Evaluate selected models

  The model is a classification model so all the classification model building algorithm has been used to build the model and across all the model the best one has been selected as final model.

  The different models which has been used has been listed below:

  1. Logistics Regression : Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes or No, Pass or fail etc) or 0.



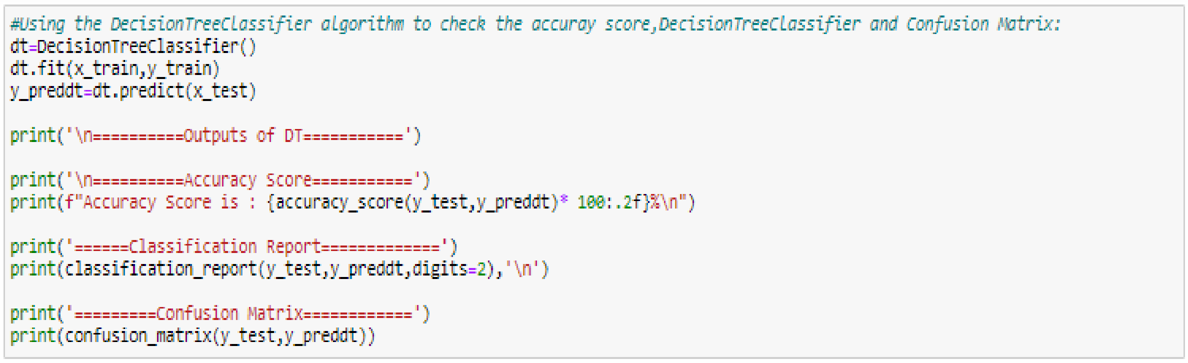
```
#seprating the train and test datasets with the best find random state
x_train,x_test,y_train,y_test=train_test_split(x_fit,y_fit,test_size=0.25,random_state=73)
```

```
#Using the DecisionTreeClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_predlr=lr.predict(x_test)

print('\n==========Outputs of Logistics Reggression===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_predlr)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_predlr,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_predlr))
```

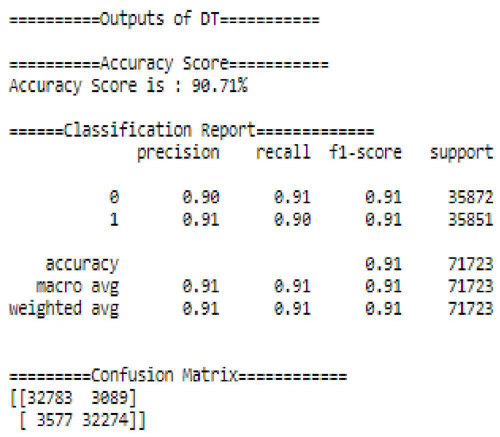
```
==========Outputs of Logistics Reggression===========

==========Accuracy Score===========
Accuracy Score is : 75.83%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.71      0.88      0.78     35872
           1       0.84      0.64      0.72     35851

    accuracy                           0.76     71723
   macro avg       0.77      0.76      0.75     71723
weighted avg       0.77      0.76      0.75     71723


=========Confusion Matrix============
[[31554  4318]
 [13014 22837]]
```

After training the data in Logistic Regression the accuracy score of the model is 75.83%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of Logistics regression model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of logistic regression model.

2. Decision Tree Classifier: Decision Trees are a non-parametric supervised learning method used for classification and regression. Decision tree basically consist of root node, branch and leaf node concept.



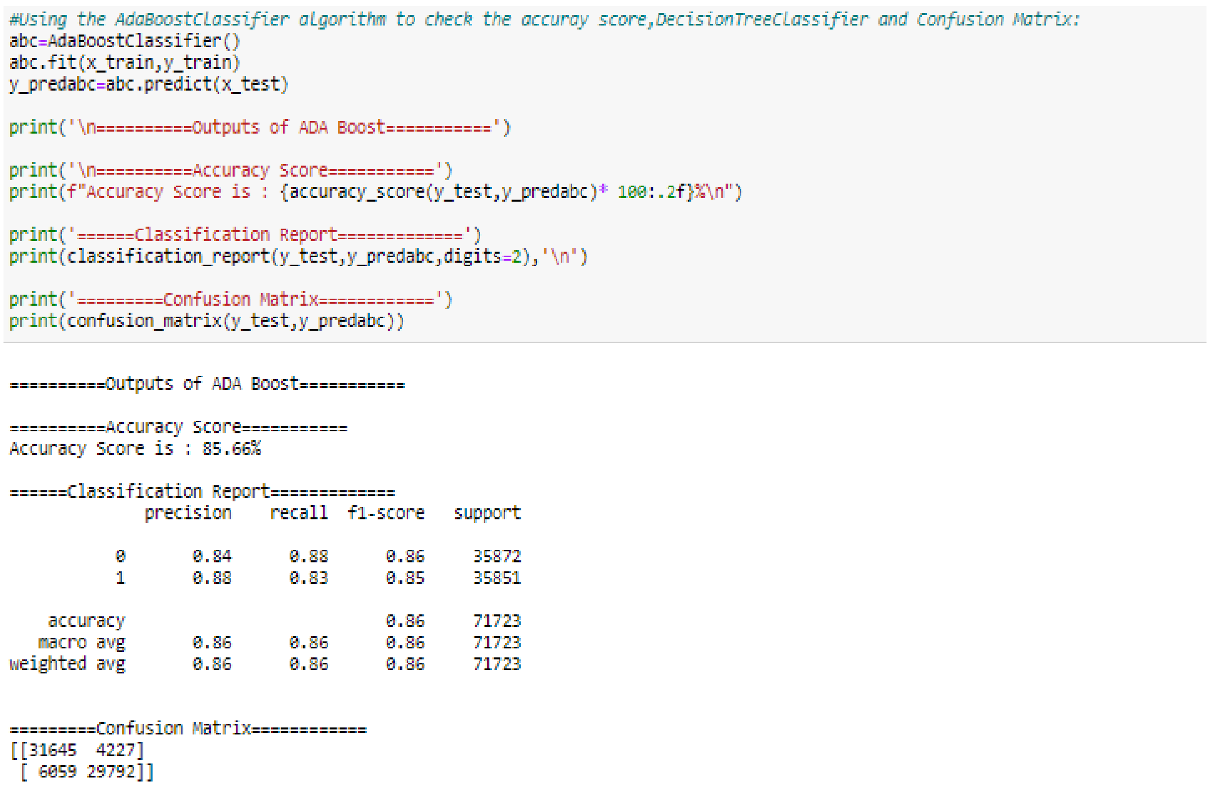
```
#Using the DecisionTreeClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_preddt=dt.predict(x_test)

print('\n==========Outputs of DT===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_preddt)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_preddt,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_preddt))
```



```
==========Outputs of DT===========

==========Accuracy Score===========
Accuracy Score is : 90.71%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.90      0.91      0.91     35872
           1       0.91      0.90      0.91     35851

    accuracy                           0.91     71723
   macro avg       0.91      0.91      0.91     71723
weighted avg       0.91      0.91      0.91     71723


=========Confusion Matrix============
[[32783  3089]
 [ 3577 32274]]
```

After training the data in Decision Tree Classifier the accuracy score of the model is 90.71%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of Decision Tree Classifier model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of Decision Tree Classifier model.

3. Ada boost Classifier: AdaBoost Classifier, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method.

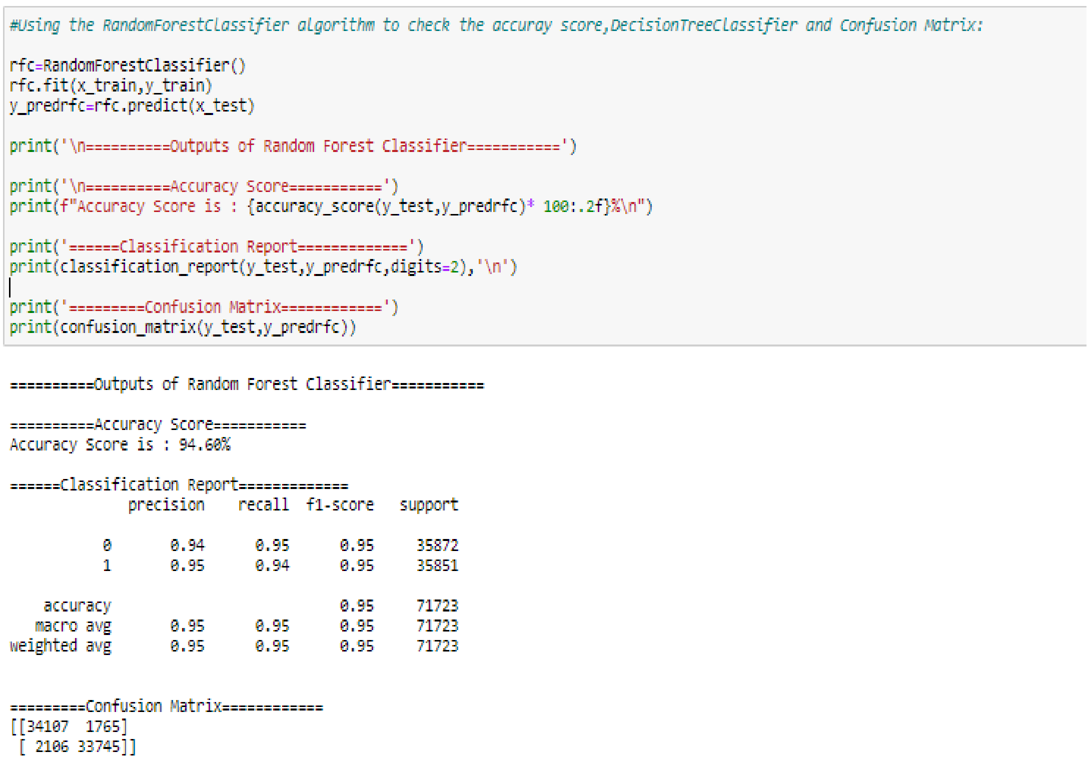


```
#Using the AdaBoostClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:
abc=AdaBoostClassifier()
abc.fit(x_train,y_train)
y_predabc=abc.predict(x_test)

print('\n==========Outputs of ADA Boost===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_predabc)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_predabc,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_predabc))
```

```
==========Outputs of ADA Boost===========

==========Accuracy Score===========
Accuracy Score is : 85.66%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.84      0.88      0.86     35872
           1       0.88      0.83      0.85     35851

    accuracy                           0.86     71723
   macro avg       0.86      0.86      0.86     71723
weighted avg       0.86      0.86      0.86     71723


=========Confusion Matrix============
[[31645  4227]
 [ 6059 29792]]
```

After training the data in Ada boost Classifier the accuracy score of the model is 85.66%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of Ada boost Classifier model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of Ada boost Classifier model.

4. Random forest Classifier: Random forests is a supervised learning algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

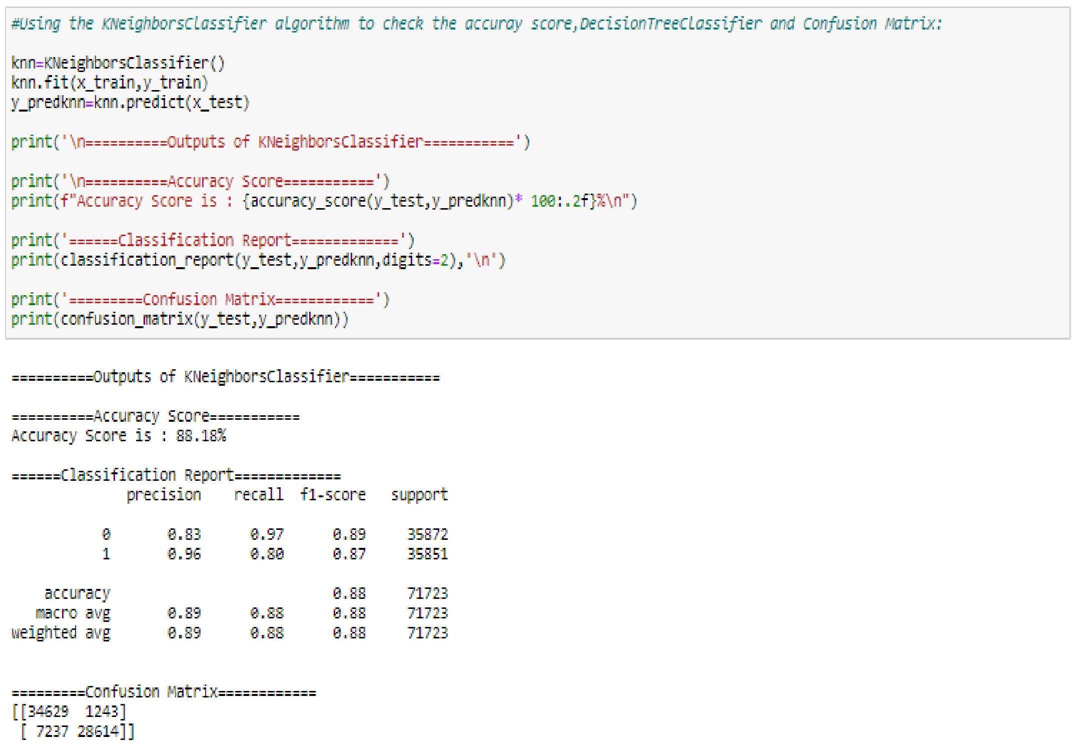


```
#Using the RandomForestClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:

rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
y_predrfc=rfc.predict(x_test)

print('\n==========Outputs of Random Forest Classifier===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_predrfc)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_predrfc,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_predrfc))
```

```
==========Outputs of Random Forest Classifier===========

==========Accuracy Score===========
Accuracy Score is : 94.60%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.94      0.95      0.95     35872
           1       0.95      0.94      0.95     35851

    accuracy                           0.95     71723
   macro avg       0.95      0.95      0.95     71723
weighted avg       0.95      0.95      0.95     71723


=========Confusion Matrix============
[[34107  1765]
 [ 2106 33745]]
```

After training, the data in Random forest Classifier the accuracy score of the model is 94.60%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of Random forest Classifier model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of Random forest Classifier model.

5. KNN Classifier: KNN also known as K-nearest neighbour is a supervised and pattern classification learning algorithm which helps us find which class the new input(test value) belongs to when k nearest neighbours are chosen and distance is calculated between them.



```
#Using the KNeighborsClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:

knn=KNeighborsClassifier()
knn.fit(x_train,y_train)
y_predknn=knn.predict(x_test)

print('\n==========Outputs of KNeighborsClassifier===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_predknn)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_predknn,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_predknn))
```

```
==========Outputs of KNeighborsClassifier===========

==========Accuracy Score===========
Accuracy Score is : 88.18%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.83      0.97      0.89     35872
           1       0.96      0.80      0.87     35851

    accuracy                           0.88     71723
   macro avg       0.89      0.88      0.88     71723
weighted avg       0.89      0.88      0.88     71723


=========Confusion Matrix============
[[34629  1243]
 [ 7237 28614]]
```

After training, the data in KNN Classifier the accuracy score of the model is 88.18%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of KNN Classifier model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of KNN Classifier model.

- Key Metrics for success in solving problem under consideration

  There are multiple points which impact the final outcome for the model.

  1. Deleting the non related features from the data frame, which is not shows impact on final model.
  2. Checking the multicolliniarity issue and removing it by deleting the features from the data frame.
  3. Converting the date column into separate date and month column using to_datetime function.

- Visualizations

  Visualization is basically finding some outcomes after visualizing the data in form of some graph or some plots. Different visualizing methods has been use the to do the analysis on the data. In this model multiple plots has been used to do analysis on the provided data.

  1. Bar Plot – To check the relation between feature and target.
  2. Dist plot – To check the data distribution and checking the linearity of data distribution.
  3. Heat Map – To check the null count and multicolliniarity issue between the features.
  4. Count Plot – To check the count of the target variable and checking the imbalanced issue of target.

- Interpretation of the Results

  From visualization, pre-processing and modelling the result should indicates 1 for non-defaulter customer and 0 for the defaulter customer. Based on the provided features data to model, It will predict the outcome as customer is defaulter or non defaulter.

CONCLUSION

- Key Findings and Conclusions of the Study

  Main finding in the models are not all the provided features are relevant to the model building. Some of them are not useful. The data consist of lots of zero values, which can be replaced with some values.

- Learning Outcomes of the Study in respect of Data Science

  There are multiple features available in the data frame but all are not useful with respect to the final outcome. Many of the unwanted features and all the features has the outliers which has been taken care of using the iqr method. There are multicolliniarity issue is also available in the features that is also managed by deleting the features having multicolliniarity issues. From the visualization the data imbalanced issue has been solved form the target columns using over sampling method.