Census Income Project

Project Flow :

1.Problem Statement

2.Data Analysis

3.EDA Conclusions

4.Pre-Processing Pipeline

5.Model Building

6.Concluding remarks

By :

Naveen

Problem Statement

This particular dataset was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker.

There were set of conditions which has been kept in the mind while extracting the dataset, : ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)).

In this case we have to predict whether a person will makes over $50k a year.

Description of fnlwgt (final weight)

The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian non-institutional population of the US. These are prepared monthly for us by Population Division here at the Census Bureau. We use 3 sets of controls. These are:

- A single cell estimate of the population 16+ for each state.
- Controls for Hispanic Origin by age and sex.
- Controls by Race, age and sex.

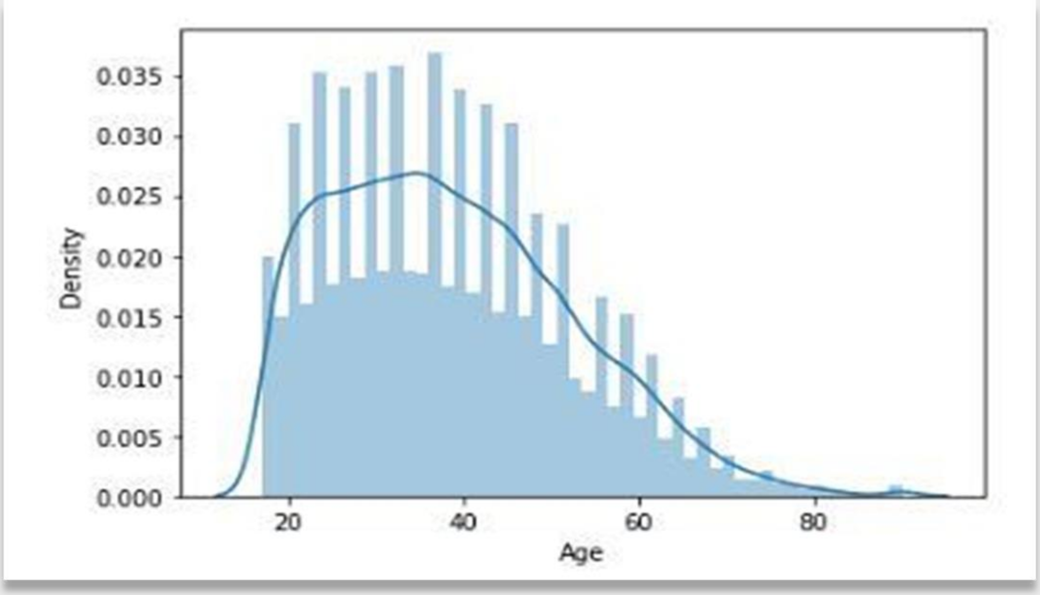
We use all three sets of controls in our weighting program and "rake" through them 6 times so that by the end we come back to all the controls we used. The term estimate refers to population totals derived from CPS by creating "weighted tallies" of any specified socio-economic characteristics of the population. People with similar demographic characteristics should have similar weights. There is one important caveat to remember about this statement. That is that since the CPS sample is actually a collection of 51 state samples, each with its own probability of selection, the statement only applies within state.

Data Analysis

We have analysed this dataset in three stages

1.Univariate Analysis :

In this analysis we have taken each column one by one and we thoroughly analysed it, depending upon the data it holds we used suitable technique to extract maximum information out of it,

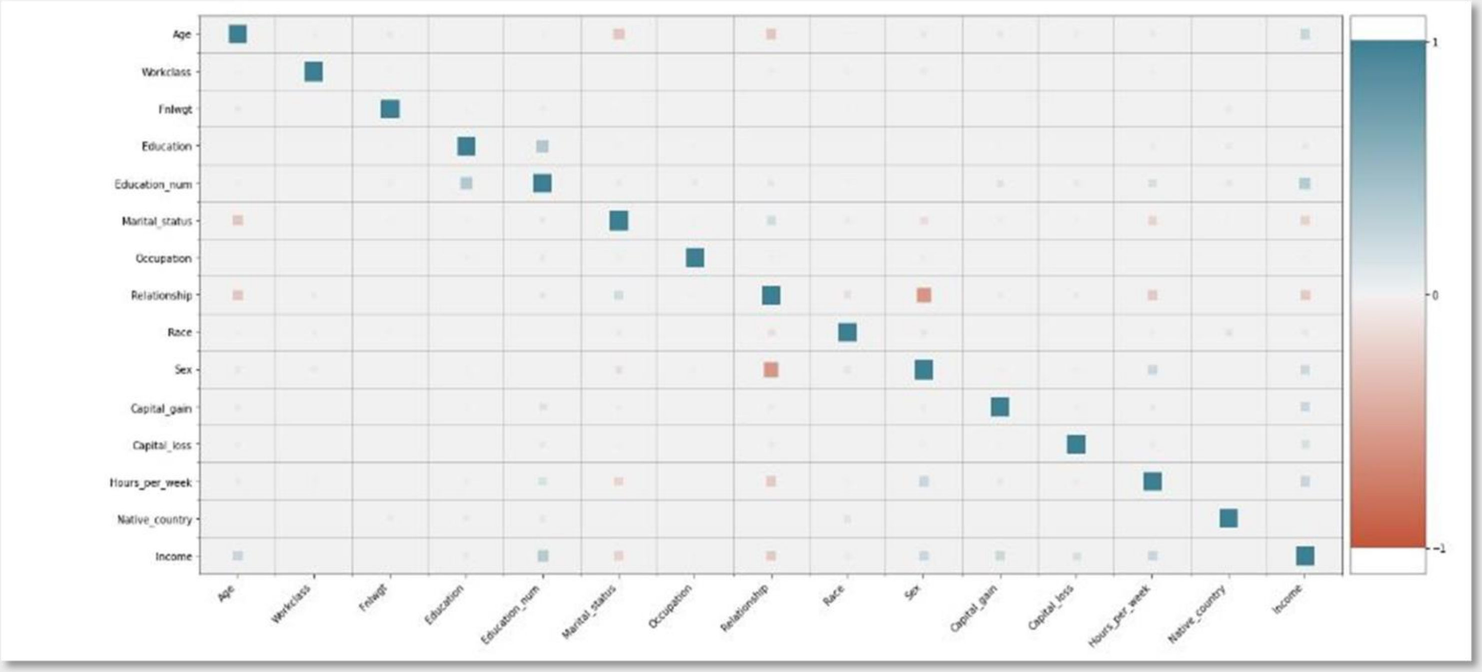


Above is an example of Age column analysis through distplot showing how the ages are being distributed, We can see that the ages are right skewed, Univaiate analysis mainly infers that analysing only one variable.

Through Univariate analysis we will get to know how the data is been distributed like whether its uniformly distributed or skewed, or any outliers are there etc.

2.Multivariate Analysis :

In case of the multivariate analysis we analyse more than two variables simultaneously, We will get to know the relationship between differerent variables, From this technique we can extract maximum informations which are hidden.



Above is an example of multivariate analysis, the above plot is corrplot depicting the relationship between all the elements simultaneously.

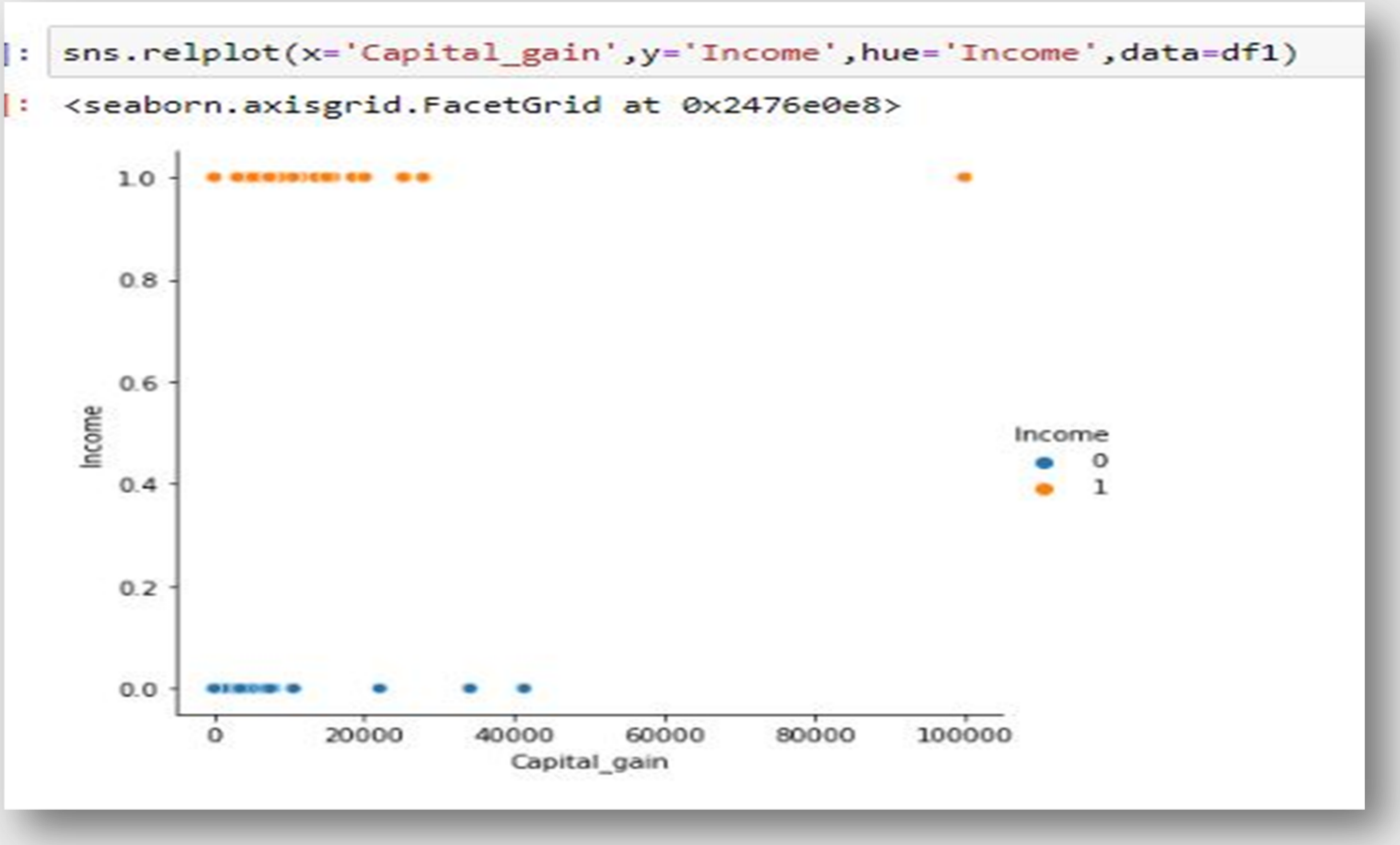
We have different plots for carrying out the multivariate analysis, Heatmap and corrplot are the major ones through which we will get to know how each variables are related to each other.

Grouping the variable and analysing them is also a type of multivariate analysis, from which we can extract maximum information because we can group the categories like yes or no and we can compare the features.from heatmap we will get to know whether an variable is positively correlated or negatively correlated.

3.Bivariate Analysis :

From multivariate analysis we will get to know which are the independent variables which are significantly correlated to each other we will analyse those pair of variables in Bi vaiate analysis.

From multivariate analysis we will get to know which are those independent variables which are significantly correlated to the response variable we can analyse those pairs in the bi variate analysis.



```
]: sns.relplot(x='Capital_gain',y='Income',hue='Income',data=df1)
]: <seaborn.axisgrid.FacetGrid at 0x2476e0e8>
```

Above is an example of bi variate analysis in which we are analysing capital gain and Income.

From bi variate analysis we will get know how an independent variable is realated to response variable, and we can prescribe required precautions to be effective.

Conclusions from EDA

1.At the age of 40 years most of the people probably gonna make 50k per annum.

2. Those people who cross more than 50k per annum will have most probably 13 as their Education_num.

3. Exec-managerial,Prof-specialty are the occupations which are highly practiced by those category of the people who are going to cross 50k per annum.

4. Those people who belongs to the category of crossing 50k perannum will have the relationship as Husband.

5. The density of capital gain is uniform for that category which belongs to crossing 50k per annum upto 4000.

6. Those people who crosses 50k per year will have the an average working per week around 45 to 60

7. Only 10.42% of people of divorced category managed to cross 50k per annum

8. 43.48% of people who belongs to Married-AF-spouse has managed to cross 50k per annum

9. More than 44.68% are married civ spouse and income is above 50k.

10. 8.13% of the married spouse absent has managed to earn income above 50k.

11. Only 4.6% never married population managed to cross 50k

12. 6.4% of separated has managed to gain income more than 50k.

13. 10.95% of womens were able to earn the income above 50k.

14. Among mens 30.58% are able to earn the income above 50k.

15. Those people with capital gain more than 13k are more favorable to achieve more than 50k annualy.

16. The maximum number of hours per week for womens are around 80, and for mens its almost 100.

Cleaning and preparing an precise input data:

1.Cleaning the datatset: This phase includes finding null values or unexpected characters in the datatset and treating them,We can treat them by various techniques like replacing, deleting, or by removing the rows.

2.**Removing the outliers :** This phase includes removing of the outliers, There are various methods for removing the outliers 1.zscore method 2.IQR method,3.Standard deviation method, We can use convinient method depending upon the situation, we have to ensure that the data loss shouldnt be more than 7%.

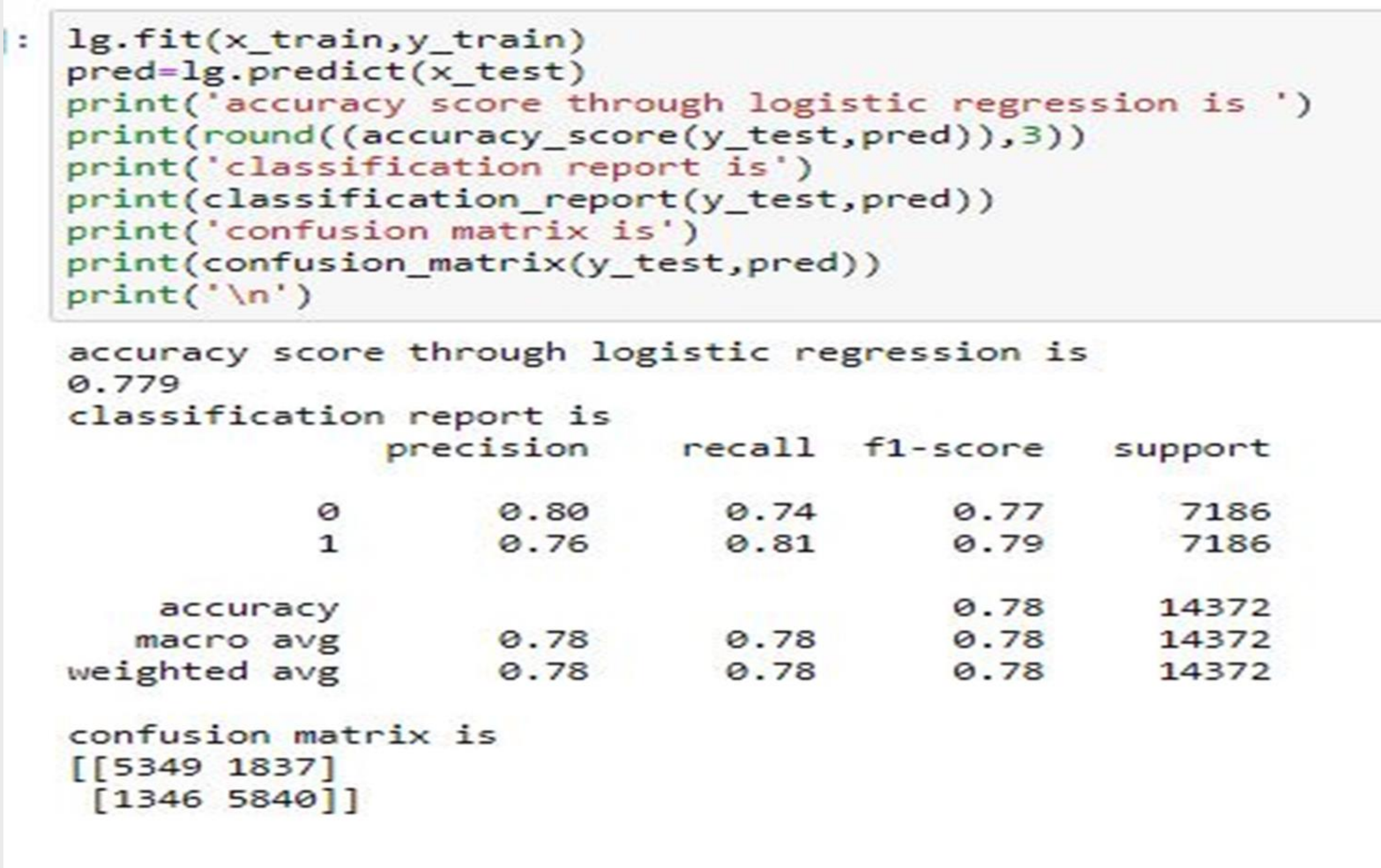
3.**Removal of the skewness :** Except categorical columns we have to remove the skewness of every other column.

4.We have to scale the independent variables if required so that the model will adapt high accuracy.

Building the models

We have ample number of models for analysing and building a solution for the classfication type of problem, In this particular scenario we gonna look at the best accuracy score and f1 score of each model and select the best model based on the result.

1.**Logistic Regression :**

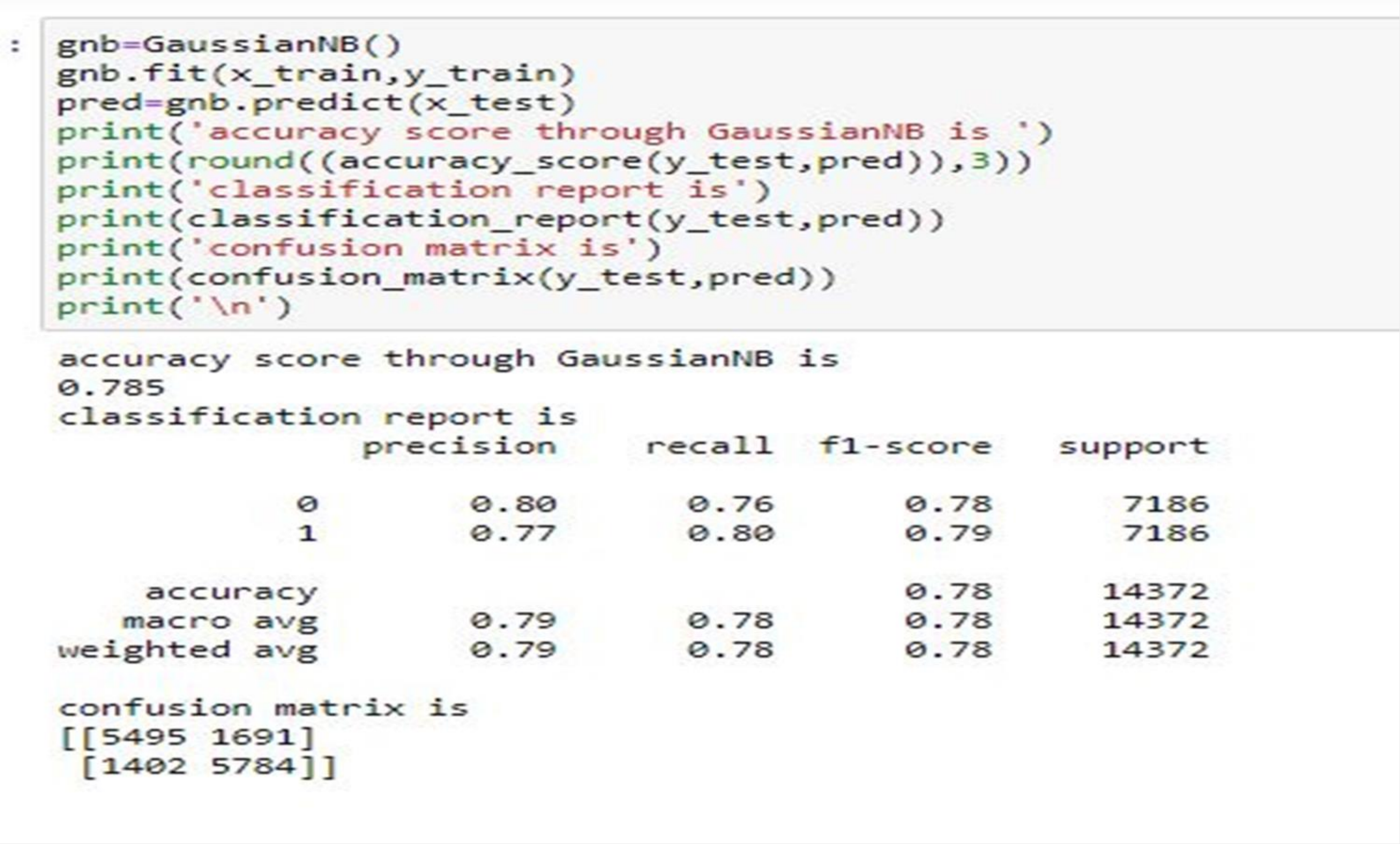


```
lg.fit(x_train,y_train)
pred=lg.predict(x_test)
print('accuracy score through logistic regression is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through logistic regression is
0.779
classification report is
              precision    recall  f1-score   support

           0       0.80      0.74      0.77      7186
           1       0.76      0.81      0.79      7186

    accuracy                           0.78     14372
   macro avg       0.78      0.78      0.78     14372
weighted avg       0.78      0.78      0.78     14372

confusion matrix is
[[5349 1837]
 [1346 5840]]
```

From the above results we can see that Logistic regression has produced an accuracy of 0.779 and f1 score as 0.77/0.79

2.GaussianNB :

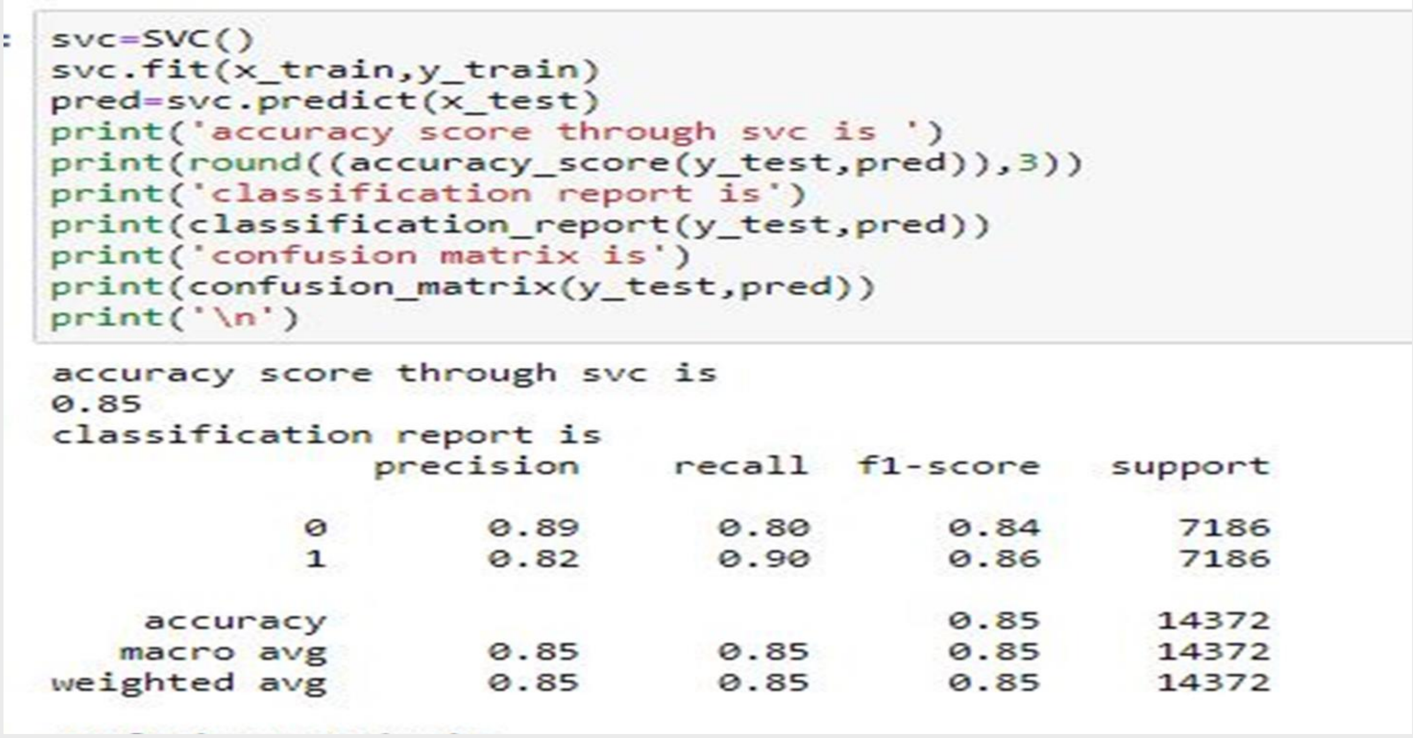


```
gnb=GaussianNB()
gnb.fit(x_train,y_train)
pred=gnb.predict(x_test)
print('accuracy score through GaussianNB is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through GaussianNB is
0.785
classification report is
              precision    recall  f1-score   support

           0       0.80      0.76      0.78      7186
           1       0.77      0.80      0.79      7186

    accuracy                           0.78     14372
   macro avg       0.79      0.78      0.78     14372
weighted avg       0.79      0.78      0.78     14372

confusion matrix is
[[5495 1691]
 [1402 5784]]
```

From the above results we can see that the accuracy score is 0.785 and f1 score is 0.78/0.79.

3.Support vector classifier :



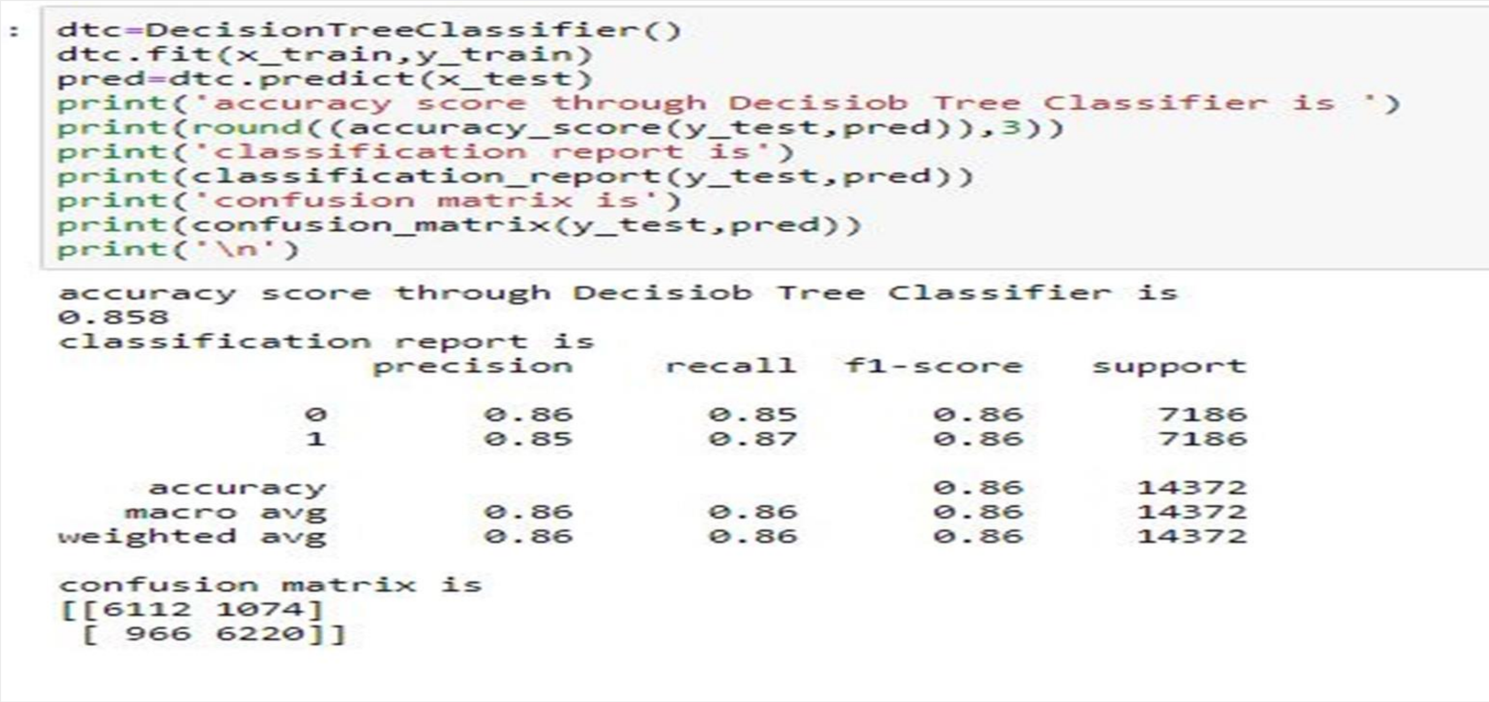
```
svc=SVC()
svc.fit(x_train,y_train)
pred=svc.predict(x_test)
print('accuracy score through svc is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through svc is
0.85
classification report is
              precision    recall  f1-score   support

           0       0.89      0.80      0.84      7186
           1       0.82      0.90      0.86      7186

    accuracy                           0.85     14372
   macro avg       0.85      0.85      0.85     14372
weighted avg       0.85      0.85      0.85     14372
```

Through svc we got an accuracy of 0.85 and f1 score of 0.84/0.86.

Decision tree classifier :

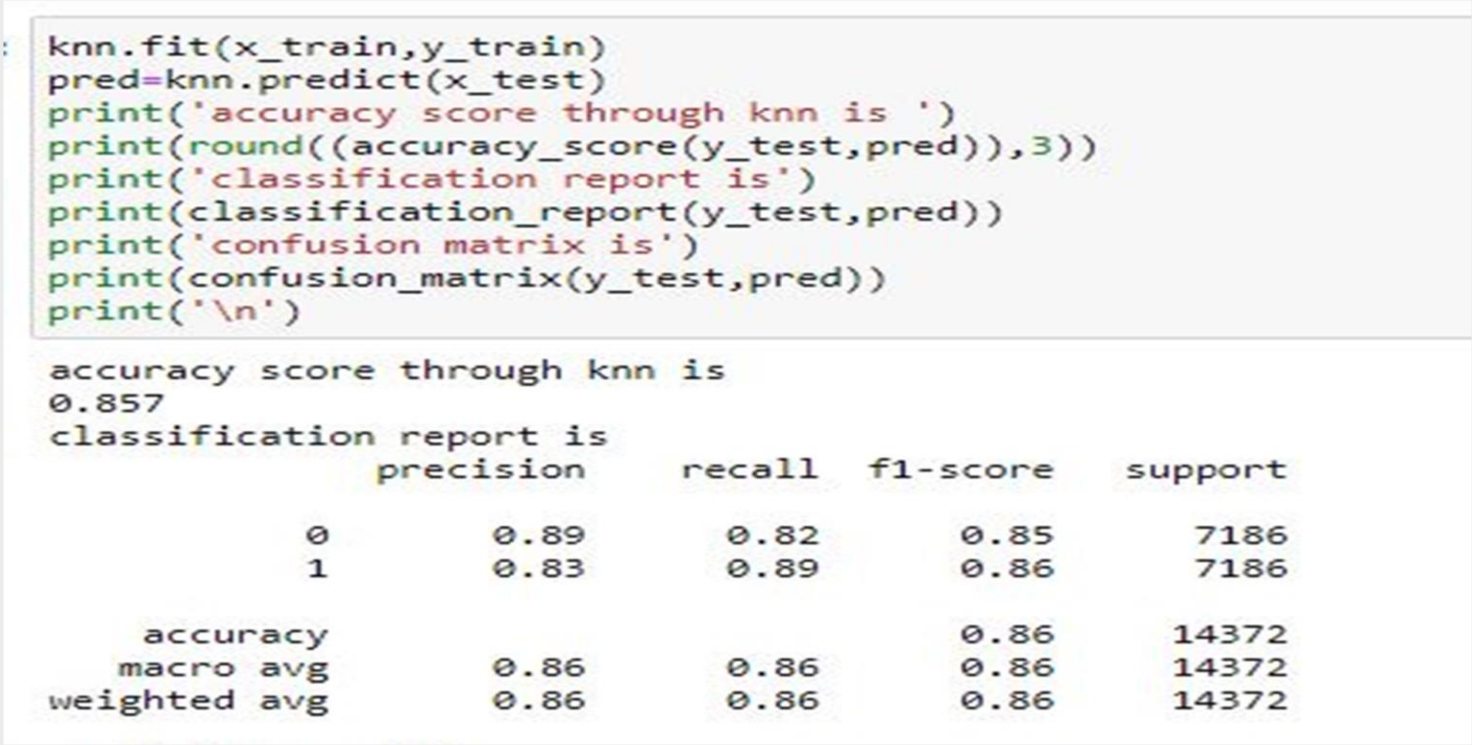
From decision tree classifier we got an accuracy score of 0.858 and f1 score of 0.86/0.86



```
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
pred=dtc.predict(x_test)
print('accuracy score through Decisiob Tree Classifier is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through Decisiob Tree Classifier is
0.858
classification report is
              precision    recall  f1-score   support

           0       0.86      0.85      0.86      7186
           1       0.85      0.87      0.86      7186

    accuracy                           0.86     14372
   macro avg       0.86      0.86      0.86     14372
weighted avg       0.86      0.86      0.86     14372

confusion matrix is
[[6112 1074]
 [ 966 6220]]
```

KnearestNeighborsclassifier:

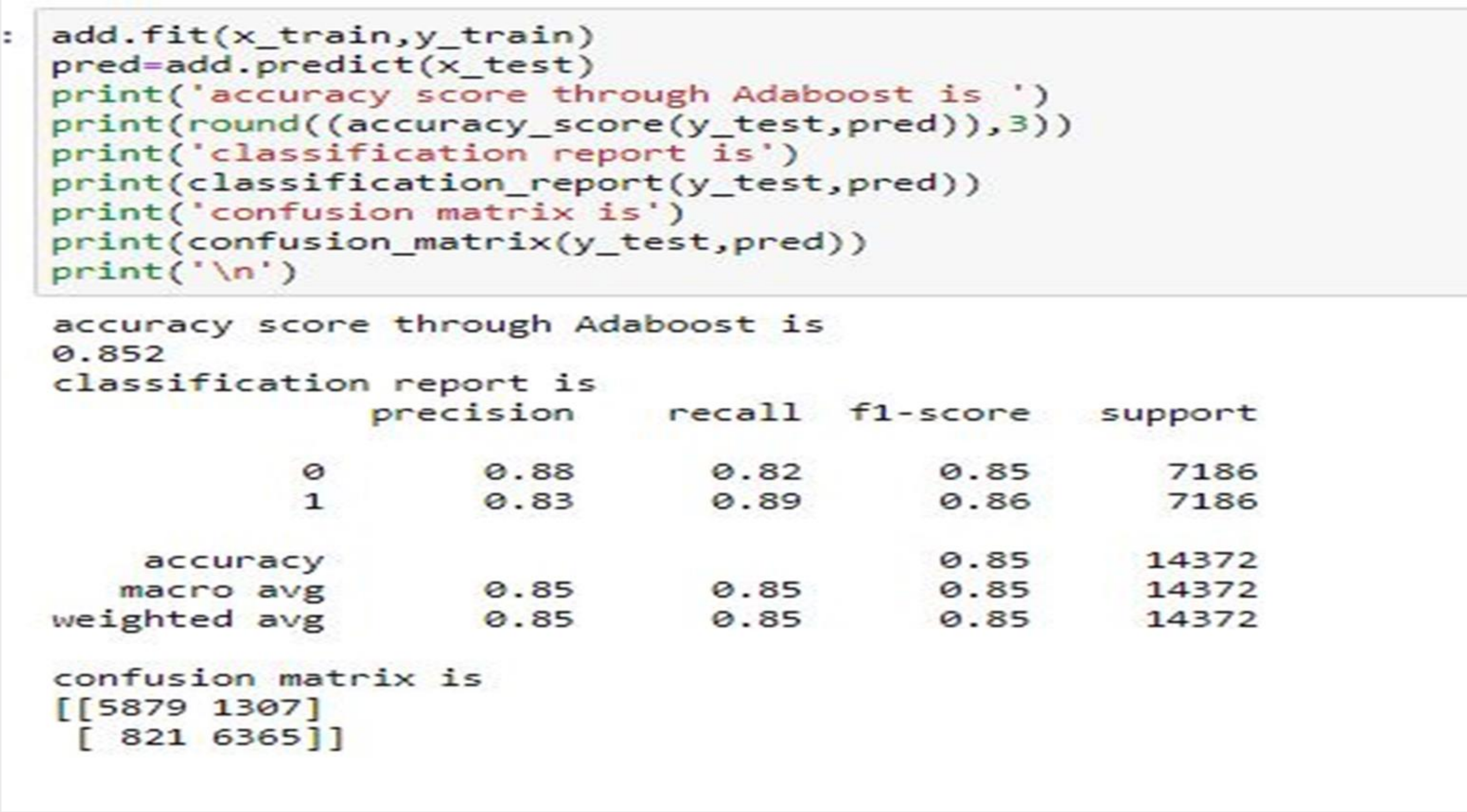


```
knn.fit(x_train,y_train)
pred=knn.predict(x_test)
print('accuracy score through knn is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through knn is
0.857
classification report is
              precision    recall  f1-score   support

           0       0.89      0.82      0.85      7186
           1       0.83      0.89      0.86      7186

    accuracy                           0.86     14372
   macro avg       0.86      0.86      0.86     14372
weighted avg       0.86      0.86      0.86     14372
```

From KNN classifier we got an accuracy score of 0.857 and f1 score of 0.85/0.86.

Adaboost Classifier :

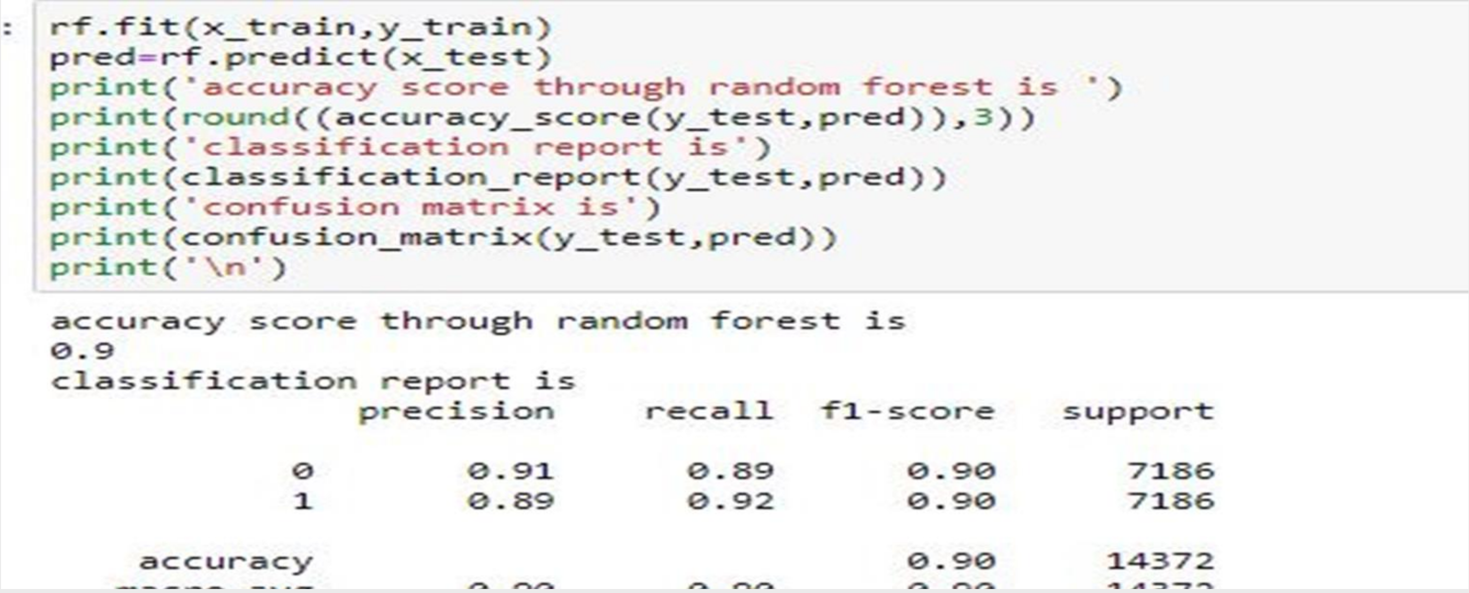


```
add.fit(x_train,y_train)
pred=add.predict(x_test)
print('accuracy score through Adaboost is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through Adaboost is
0.852
classification report is
              precision    recall  f1-score   support

           0       0.88      0.82      0.85      7186
           1       0.83      0.89      0.86      7186

    accuracy                           0.85     14372
   macro avg       0.85      0.85      0.85     14372
weighted avg       0.85      0.85      0.85     14372

confusion matrix is
[[5879 1307]
 [ 821 6365]]
```

From this classifier we got an accuracy score of 0.852 and f1 score of 0.85/0.86.

RandomForestClassifier :



```
rf.fit(x_train,y_train)
pred=rf.predict(x_test)
print('accuracy score through random forest is ')
print(round((accuracy_score(y_test,pred)),3))
print('classification report is')
print(classification_report(y_test,pred))
print('confusion matrix is')
print(confusion_matrix(y_test,pred))
print('\n')
```

```
accuracy score through random forest is
0.9
classification report is
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      7186
           1       0.89      0.92      0.90      7186

    accuracy                           0.90     14372
```

From Random forest classifier we got an accuracy of 0.9 and f1 score of 0.90/0.90

Based upon the least difference between accuracy and cross val score, and based upon the f1 score its better to choose Random Forest Classifier as the best model.

Conclusion remarks:

The model has been built for best accuracy optimising all loopholes in the datatset.The model has secured an auc_roc score of 0.898 and It scored very well in all its test.

......**Thank you!!**