

**Big Data Project**

**Submitted To**

**Eng. Omar Samir**

By Team 10

**Asmaa Sayed**

**Aya Samir**

**Bassant Mohammed**

**Khalid Ali**

**Problem Description**

Consider you are a Data Analyst with a private bank or a loan distribution firm. Your organization receives many applications in a given day. In order to process the applications, you sometimes miss out on accepting applications from people who are able to pay loans in time and end up sanctioning loans to those who later turn out to be defaulters.

We worked on Current\_app data set to analyze loan applications whether or not clients are defaulters. The data set has 307511 rows and 122 columns.

**Project Pipeline**

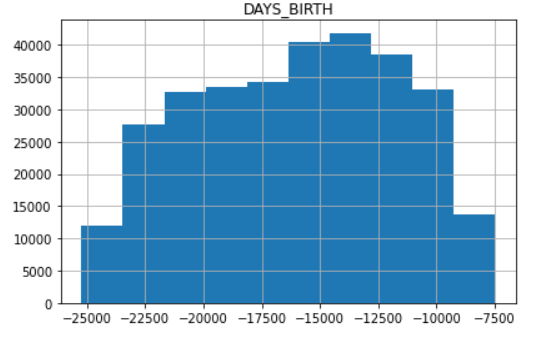
Due to the large number of columns and considering the purpose of the project which is classification with Big Data techniques. followed the following flow:

**Notice: Due to the large number of features in this dataset we mentioned the most important features for each step so that the report won’t be too large. For further details you could visit notebook attached.**

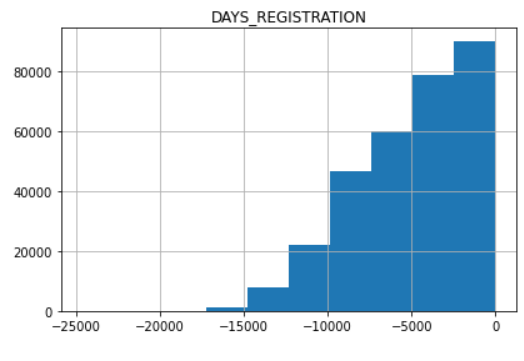
1. **Data Preprocessing**
   1. ***Check invalid values***

Check if any features have invalid data and replace it with valid data if possible. We Noticed that the following features have invalid data.

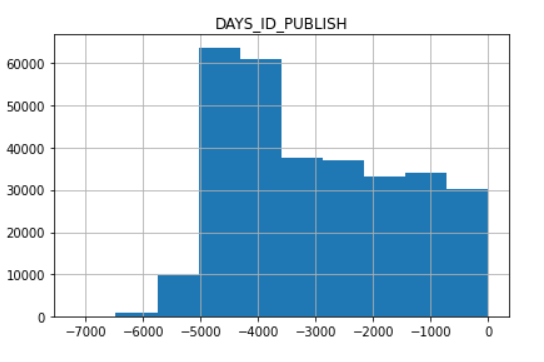
* + 1. DAYS\_BIRTH have values from -25000 to -7500



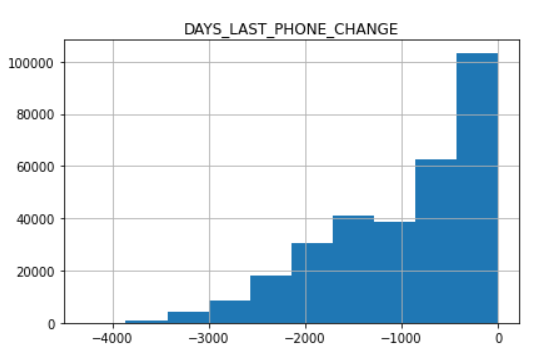
* + 1. DAYS\_REGESTRATION have values from -25000 to 0



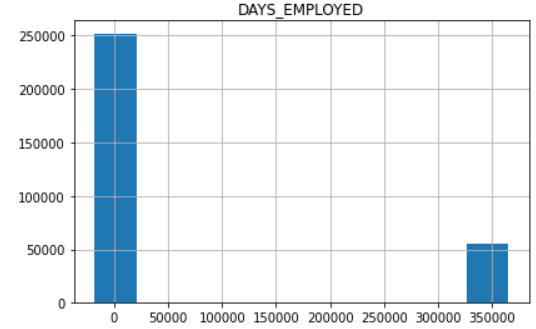
* + 1. DAYS\_ID\_PUBLISH have values from -7000 to 0



* + 1. DAYS\_LAST\_PHONE\_CHANGE has values from -4000 to 0



* + 1. DAYS\_EMPLOYED have values less than 0

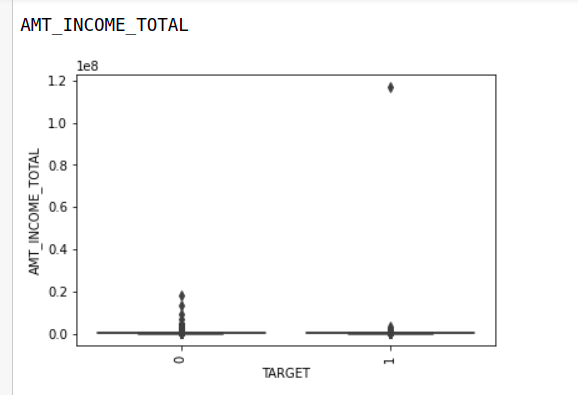


* 1. ***Outliers Detection***

Draw boxplot for continuous features and check/remove outliers.

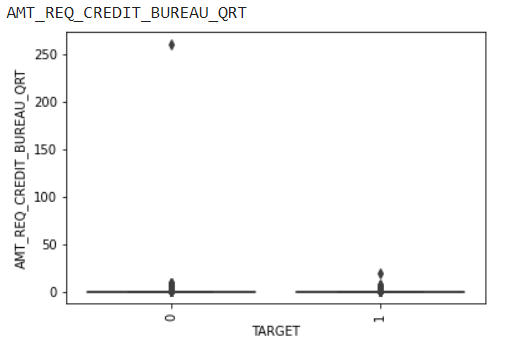
From the plotted figures we notice the following features have outliers

* + 1. AMT\_INCOME\_TOTAL has outliers, so we removed rows with values larger than 0.2e8



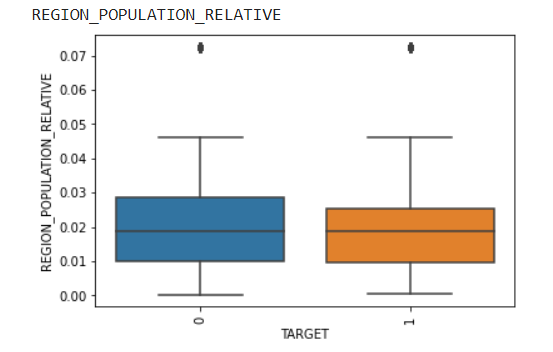
* + 1. AMT\_REQ\_CREDIT\_BUREAU\_QRT

Has outliers, so we removed rows with values larger than 200



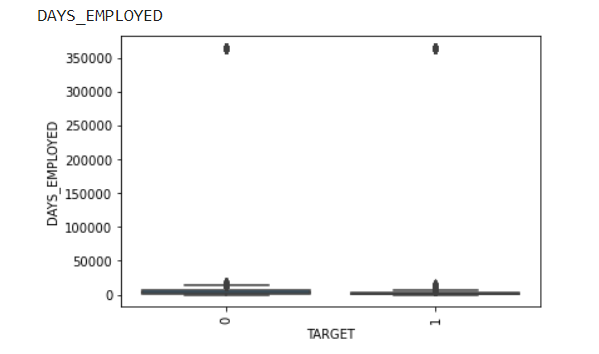
* + 1. REGION\_POPULATION\_RELATIVE

Has outliers, so we removed rows with values larger than 0.05



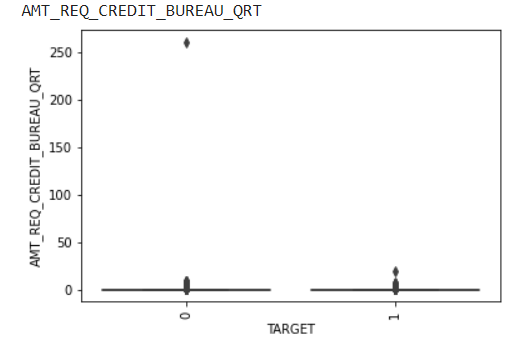
* + 1. DAYS\_EMPOLYED

Has outliers, so we removed rows with values larger than 50000



* + 1. AMT\_REQ\_CREDIT\_BUREAU\_QRT

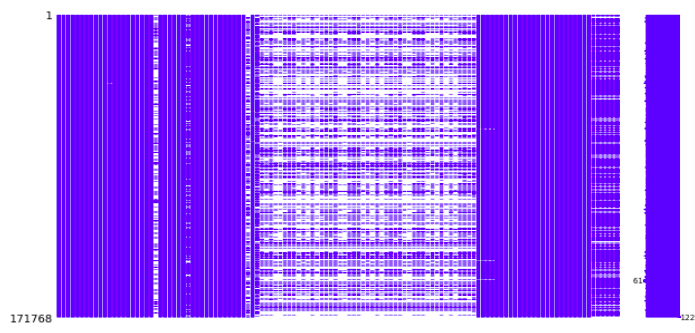
Has outliers, so we removed rows with values larger than 40



After this Step the dataset has shape of (245384, 122)

* 1. ***Check & visualize Nan values***

Visualize Nan values using Missingno library



* 1. ***Fill Nan Values***

Recommended 1: Drop columns with Nan values larger than 50%

Columns decreased from 122 to 73 columns.

Recommended 2: Fill columns with Nan values less than 13%

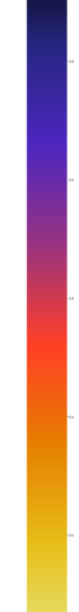
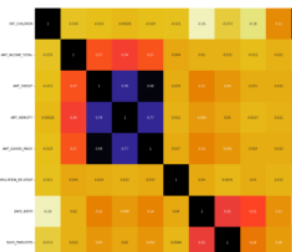
Drop remaining rows with Nan values

The output dataset after this step has the shape of (83997, 77)   
  
after removing SK\_ID\_CURR column which is not useful.

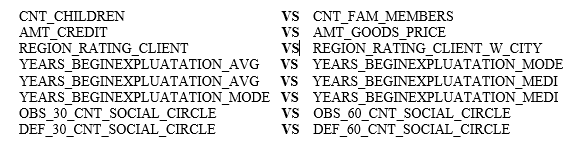
1. **Bivariate Analysis**
   1. ***Continuous Vs Continuous***

Compute correlation between continuous features to remove dependent features.

*Slice of heatmap*



We found the following columns highly correlated to each other, over 0.85;

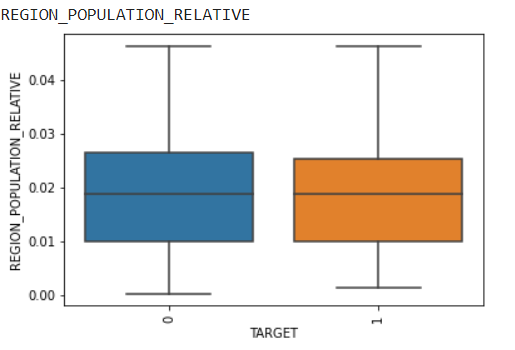
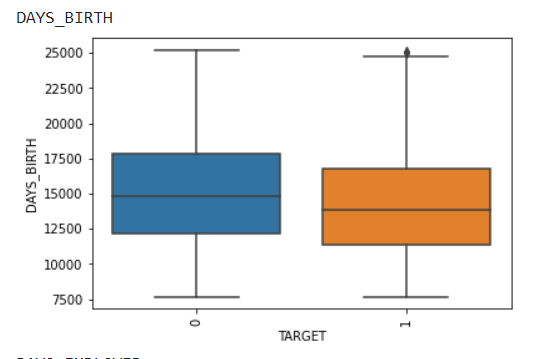


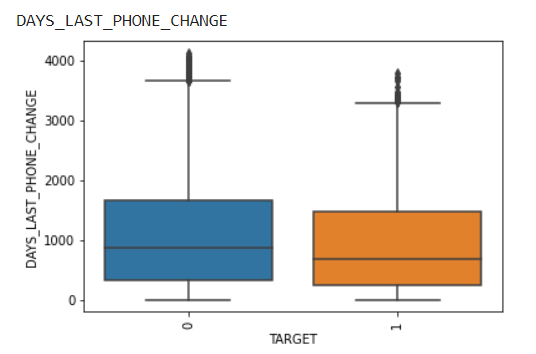
We Removed Unique columns from left side.

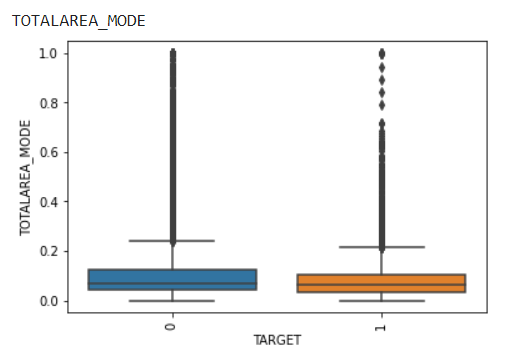
* 1. ***Continuous Vs Output (Categorical)***

Draw box plot for continuous features Vs output target to and check if distribution changes for each class. Remove features that have same distribution for both classes

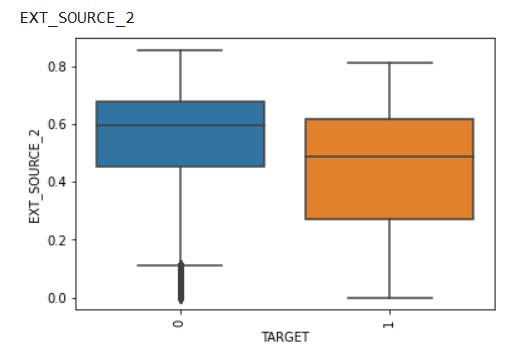
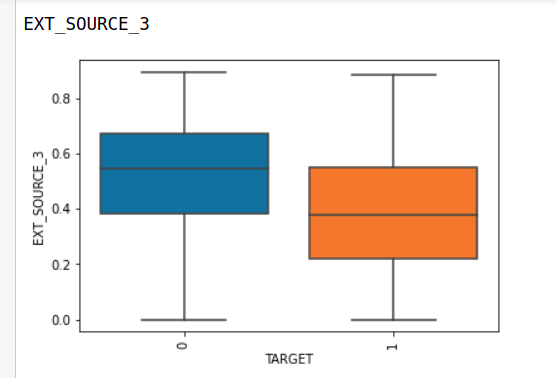
*Examples for Features that are* ***NOT*** *correlated to response feature*







*Examples for Features that* ***ARE*** *correlated to response feature*

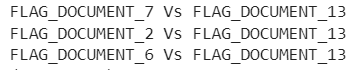


After This step the dataset has a shape of (83997, 58)

* 1. ***Binary Categorical Vs binary categorical***

To find correlation between binary categorical features we used PEARSON’R method. Remove dependent features with correlation r larger than 0.85 and p-value less than 0.05.

We found the following columns correlated



We removed whole three columns on the left side.

After This step the dataset has a shape of (83997, 55)

* 1. ***Binary Features Vs Output***

We used same method (PEARSON’R) to remove features with correlation r less than 0.04 with output or p-value less than 0.05.

The only remaining binary feature after this step was CODE GENDER

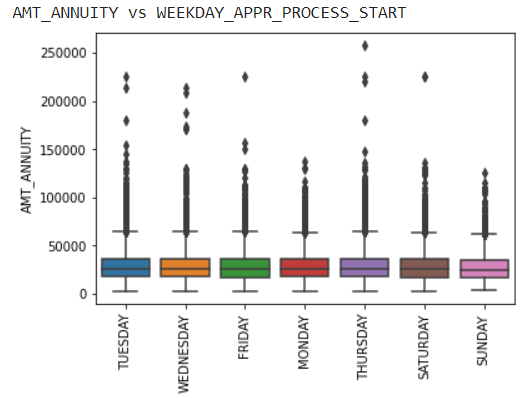
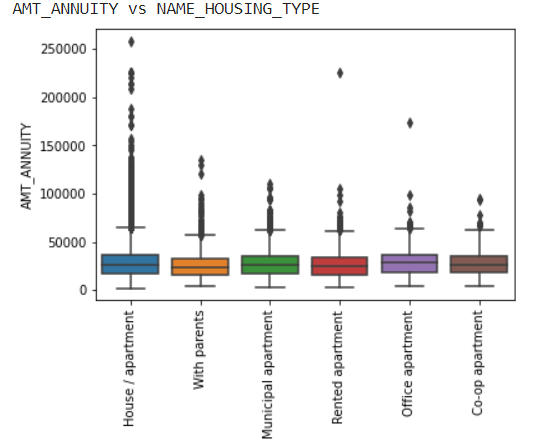
After This step the dataset has a shape of (83997, 22)

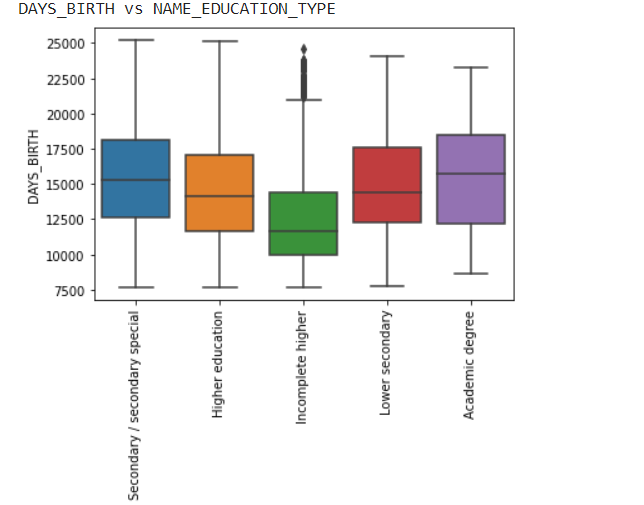
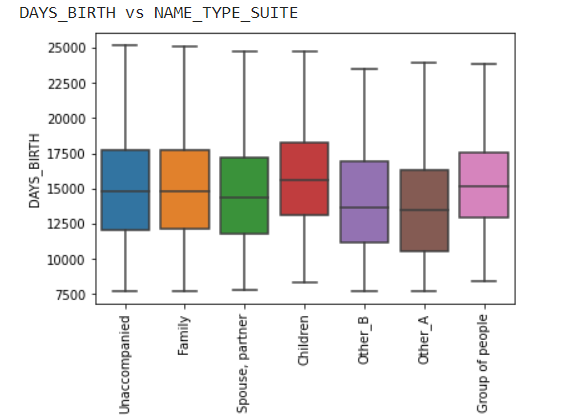
* 1. ***Multiple Categorical Vs Continuous***

Draw box plot for continuous features Vs output target to and check if distribution change for each class. Remove features that have different distribution for all classes. Which indicates that each class has a different range of values (features are dependent).

From The output figures we didn’t find any features that are highly correlated where there were always large intersection between class ranges.

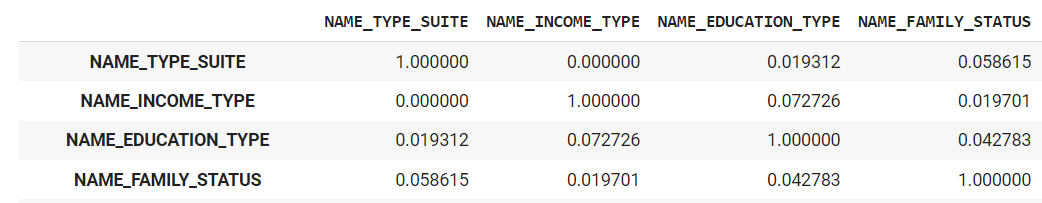
*Examples for output figures*





* 1. ***Multiple Categorical Vs Categorical***

For this test we used Chi-Square test to calculate the correlation between multi-categorical features. Remove dependent features with correlation larger than 0.85 and features that have correlation with response feature of value less than 0.05.



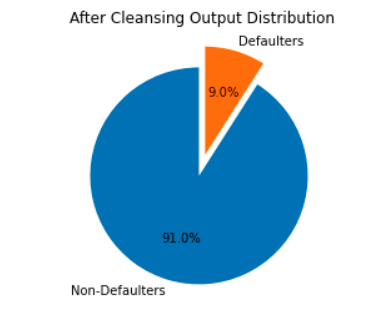
*Cross section from output matrix*

We found that no features are highly correlated to each other. And we removed features that have correlation with response features less than 0.05. The following features were removed.

* NAME\_TYPE\_SUITE
* NAME\_INCOME\_TYPE
* NAME\_FAMILY\_STATUS
* NAME\_HOUSING\_TYPE
* WEEKDAY\_APPR\_PROCESS\_START'

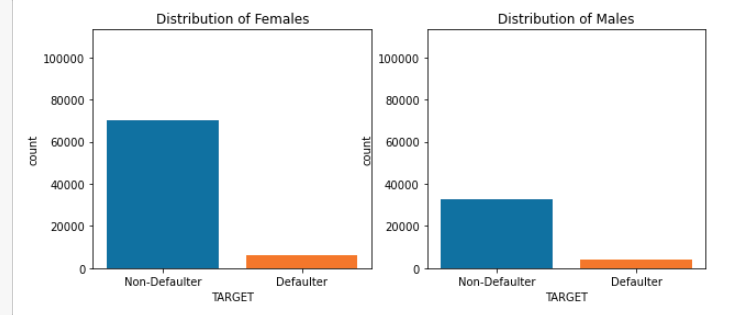
After This step the dataset has a shape of (83997, 17)

1. **Univariate Analysis**
   1. ***Distribution of defaulter and non-defaulters***



**Insight**: Our dataset has 91% non-Defaulters and 9% Defaulters so our dataset is imbalanced

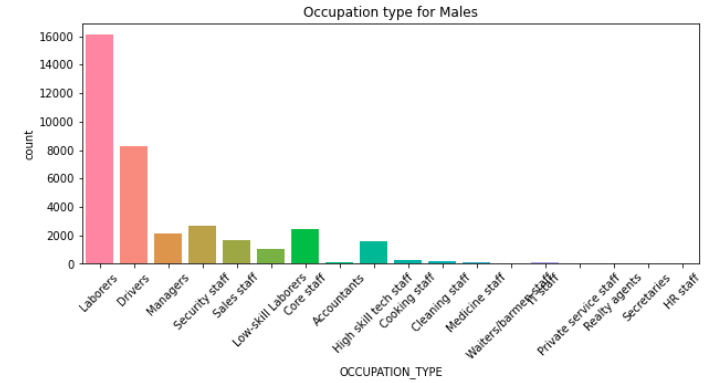
* 1. ***Distribution of Applicants gender***



**Insight**: the number of non-Defaulters for men is less than the number of defaulters for women to get better insight we better calculate percentage.

**Insight**: the percentage Defaulters of males is larger than females

* 1. ***Occupation Type Analysis***



**Insight**: majority of male clients are laborers followed by drivers.

**Insight**: majority of female clients are Sales staff followed by Laborers, followed by Core staff.

1. **Model/Classifier training**

We used Logistic regression from Statsmodels library to find out the significance of features in prediction. We removed features that have p-value less than 0.05 which indicates that features are not significant for prediction.

We conclude that the most effective features for our prediction are the following;

* AMT\_ANNUITY
* EXT\_SOURCE\_3
* NAME\_EDUCATION\_TYPE

**Results and Evaluation**

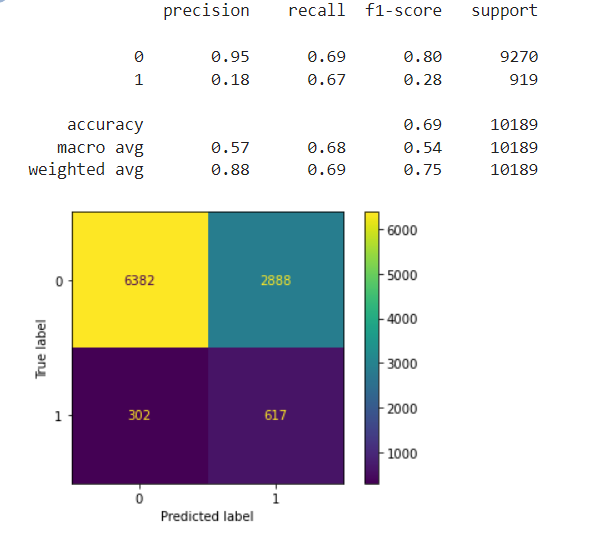
We trained the models to achieve best f1-score for model class 1 on validation set.

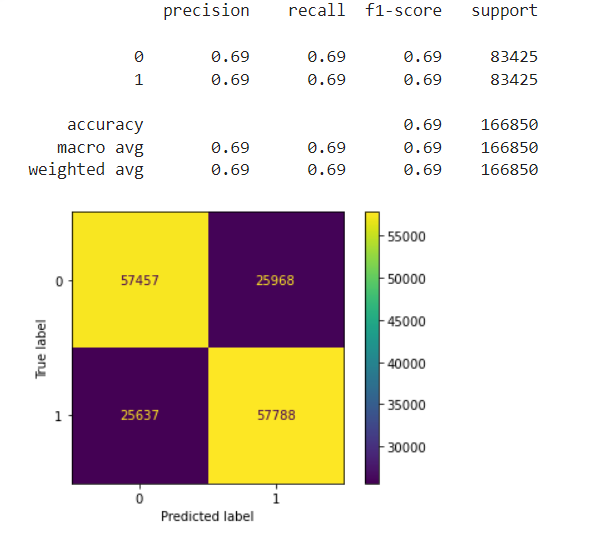
* The Logistic Regression model achieved on Over Sampled data the following scores

Train: accuracy = 69%, f1-score for class 1 = 69%, f1-score for class 0 = 69%

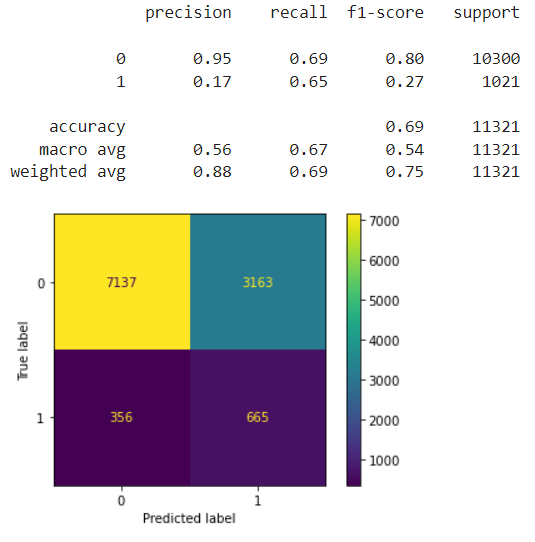
Valid: accuracy = 69%, f1-score for class 1 = 28%, f1-score for class 0 = 80%

Test: accuracy = 69%, f1-score for class 1 = 27%, f1-score for class 0 = 80%

***Train Set Valid Set***



***Test Set***



* Stochastic Logistic Regression on Over Sampled data using **MapReduce**

Train: accuracy = 61%, f1-score for class 1 = 66%, f1-score for class 0 = 54%

Valid: accuracy = 48%, f1-score for class 1 = 20%, f1-score for class 0 = 61%

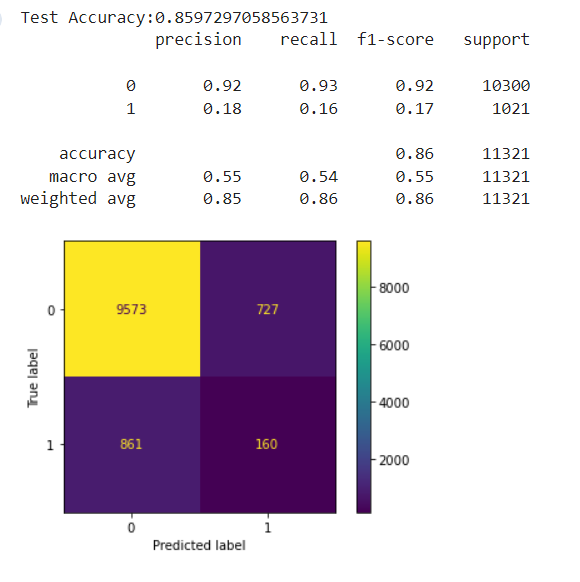
Test: accuracy = 48%, f1-score for class 1 = 20%, f1-score for class 0 = 81%

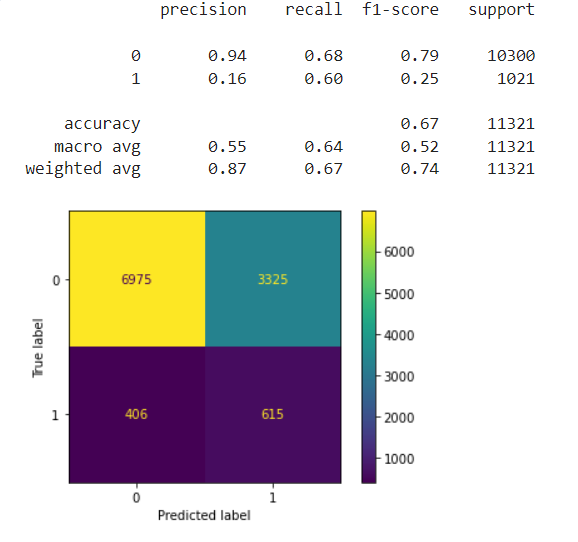
**Other Trials**

We also tried the following classifiers on imbalanced data, under sampled data, over sampled data: we will show the confusion matrix

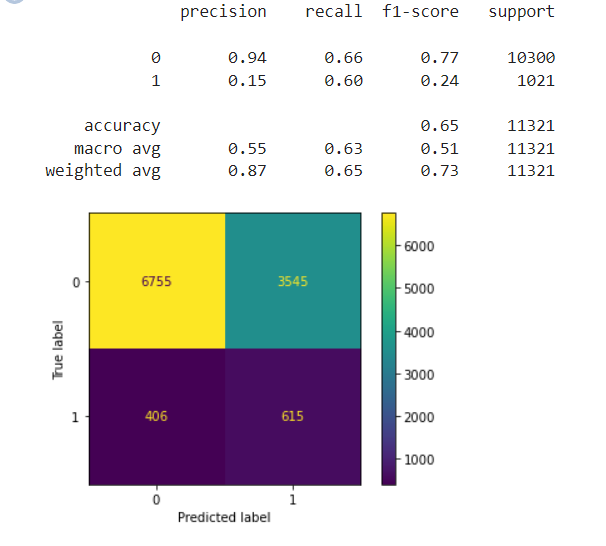
* 1. Naive Bayes

***Imbalanced Data UnderSampled***



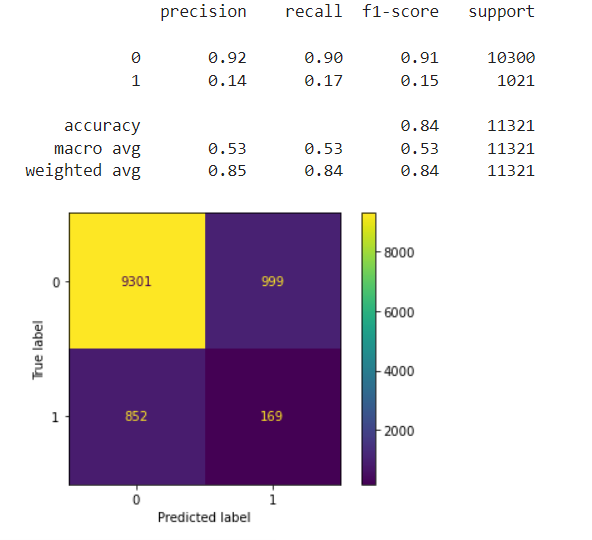
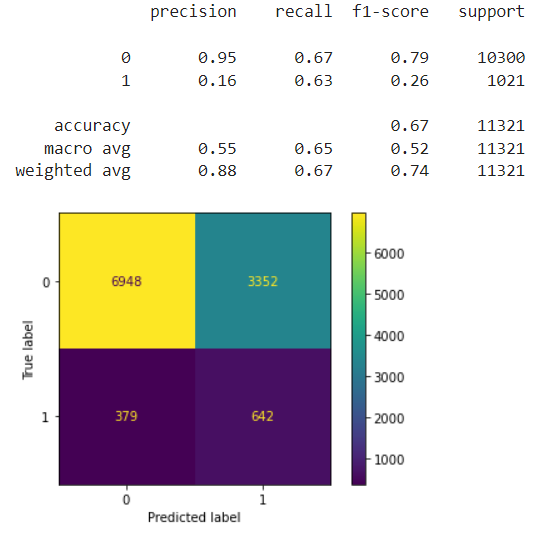


***OverSampled***

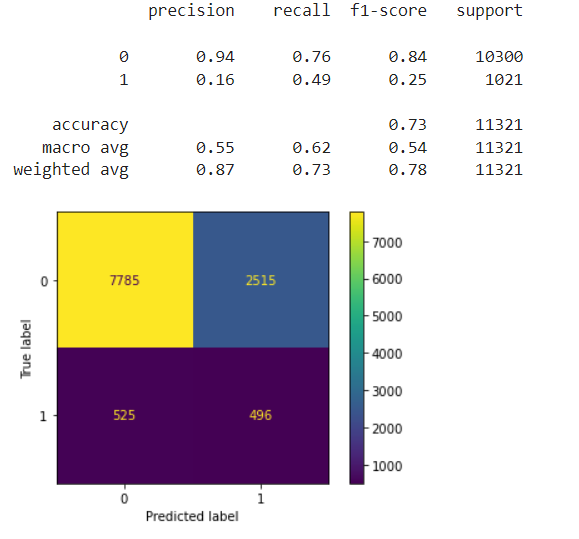


* 1. Decision Tree

***Imbalanced*** ***UnderSampled***

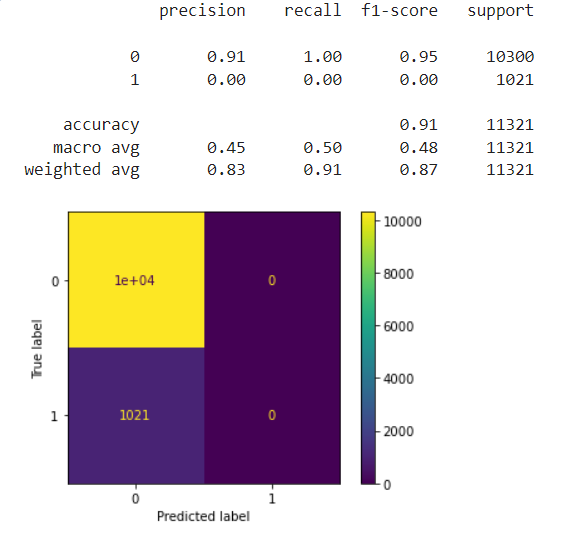
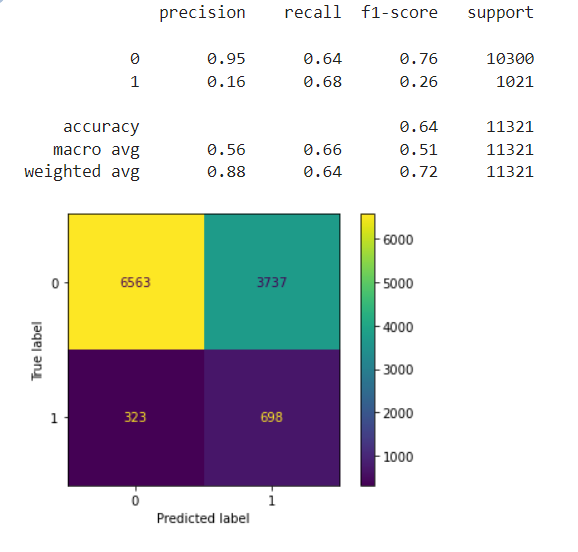


***OverSampled***

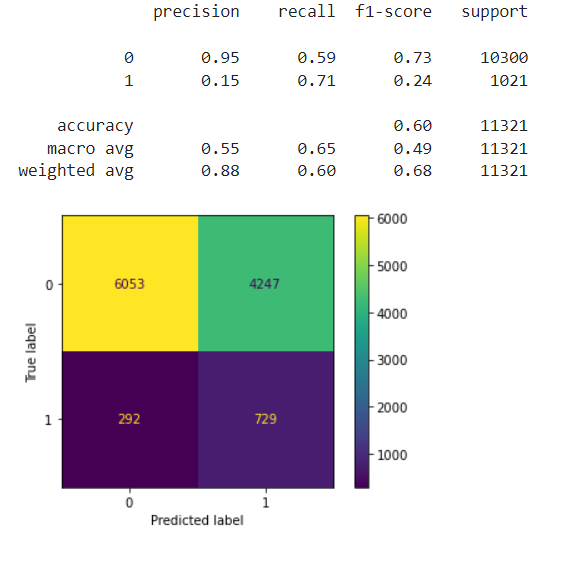


* 1. KNN

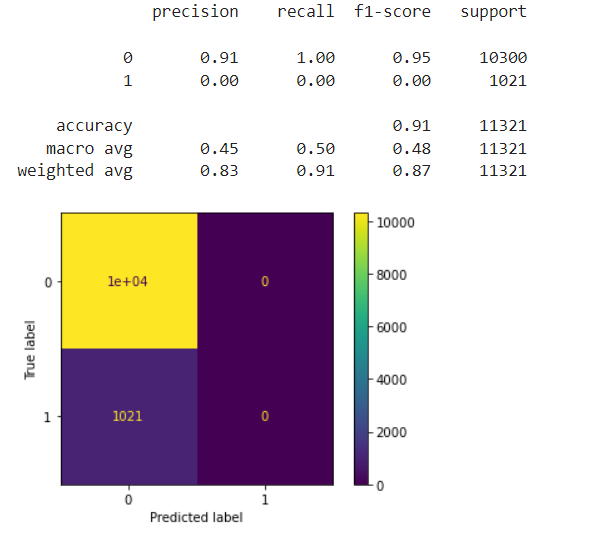
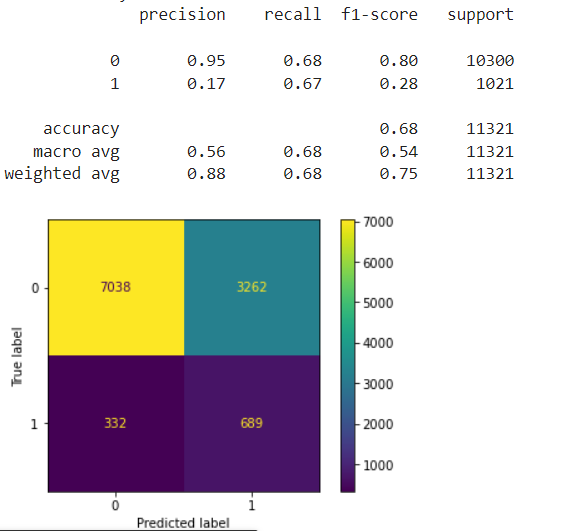
***Imbalanced Data UnderSampled***



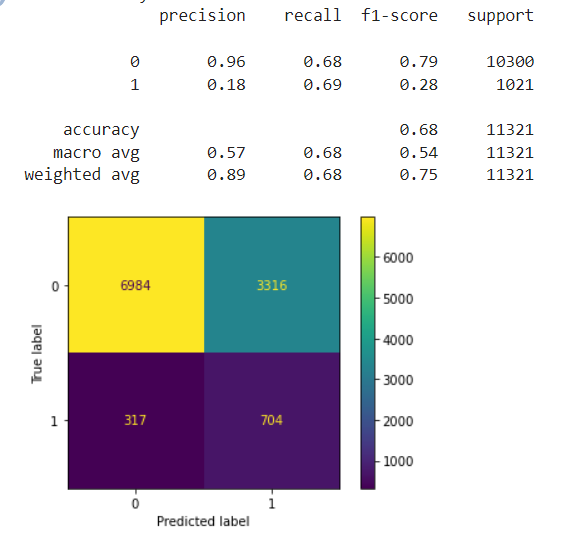
***OverSampled***



* 1. Support Vector Machine

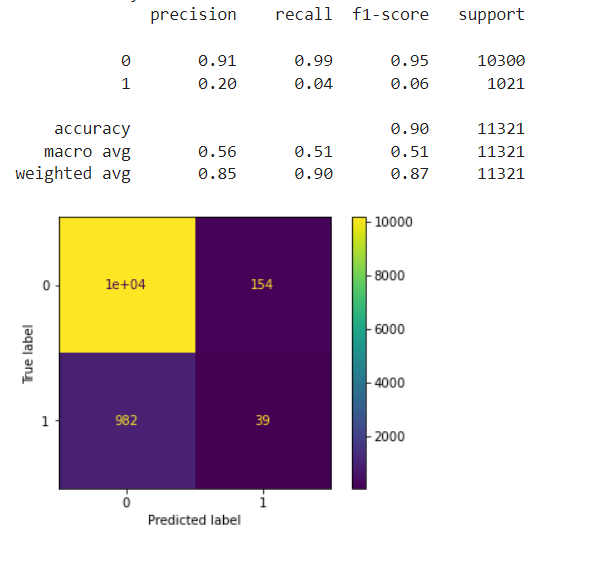
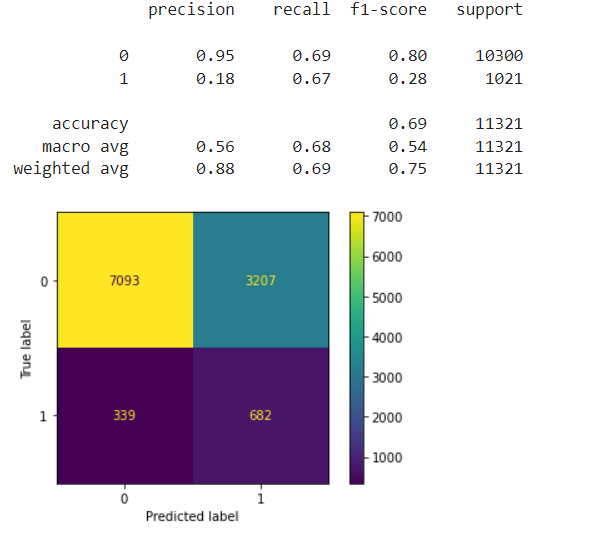
***Imbalanced Data UnderSampled***

***OverSampled***

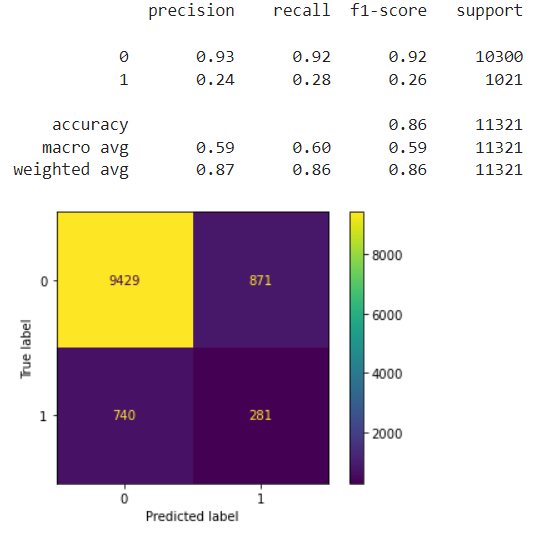


* 1. Random Forest

***Imbalanced Data UnderSampled***



***OverSampled***



Over Sampling and Under Sampling achieved almost similar results, imbalanced data achieved the worst f1-score for class 1 where it was equal to zero.

Best Model Based on F1-Score is **Support Vector Machine**

Which achieved the following scores on Over-Sampled Data

**Train**: accuracy = 68%, f1-score for class 1 = 28%, f1-score for class 0 = 80%

**Future Work:** We can work on collecting more data of class 1 to achieve some sort of balance which will be useful for our prediction.