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 wheather or not clients are defaulters. The data set has 307511 rows and 122 columns.

Project Pipeline

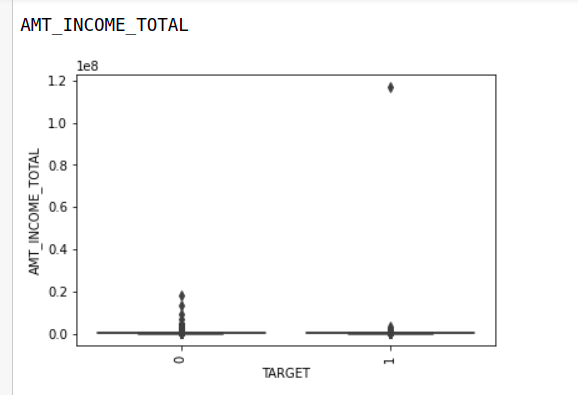
Due to the large number of columns and considering the purpose of the project which is classification with **Big Data** techniques. followed the following flow:

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

**Check if any features have invalid data and replace it with valid data if possible**

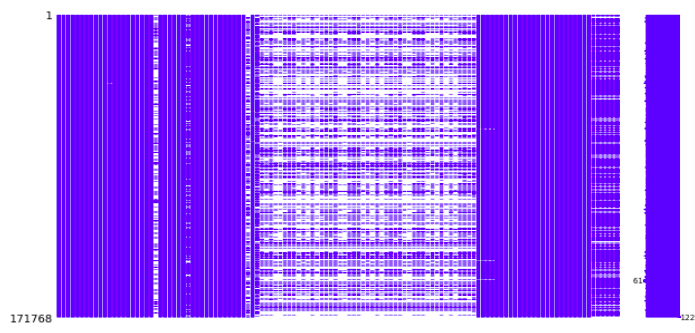
* 1. Outliers Detection

Draw boxplot for continuous features and check/remove outliers



* 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%**

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

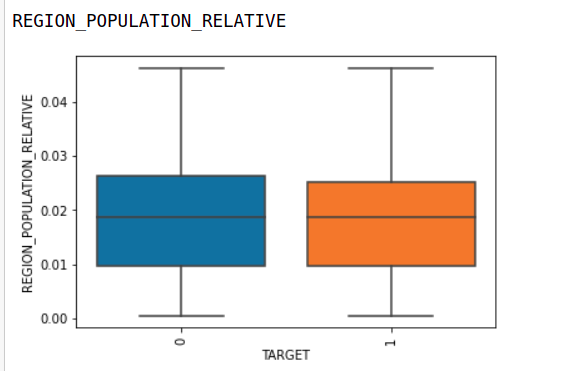
**Drop remaining rows with Nan values**

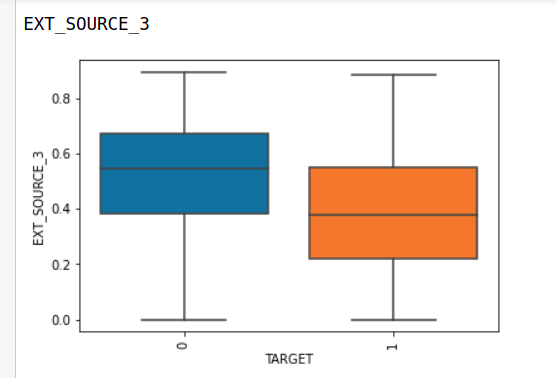
1. **Bivariate Analysis**
   1. **Continuous Vs Continuous**

**Compute correlation between continuous features to remove dependent features.**

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

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





* 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.**

* 1. **Binary Features Vs Output**

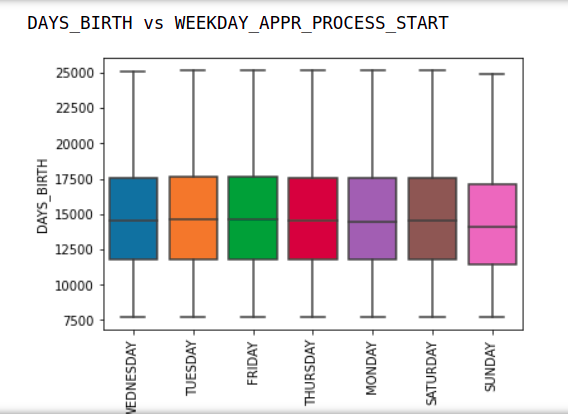
**We used same method (PEARSON’R) to remove features with correlation r less than 0.05 with output.**

* 1. **Multiple Categorical Vs Continuous**

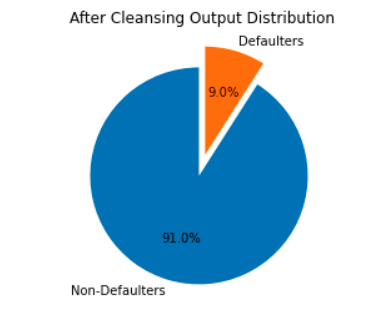
**Draw box plot for continuous features Vs output target to and check if distribution change for each class. Remove features that have same distribution for both classes**

* 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.**

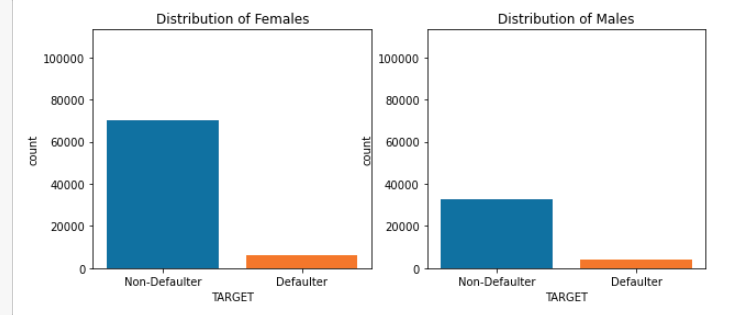


1. **Univariate Analaysis**
   1. **Distribution of defaulter and non-defaulters**



**Insight: O**ur dataset has 91% Non-Defaulters and 9% Defaulters so our dataset is impalanced

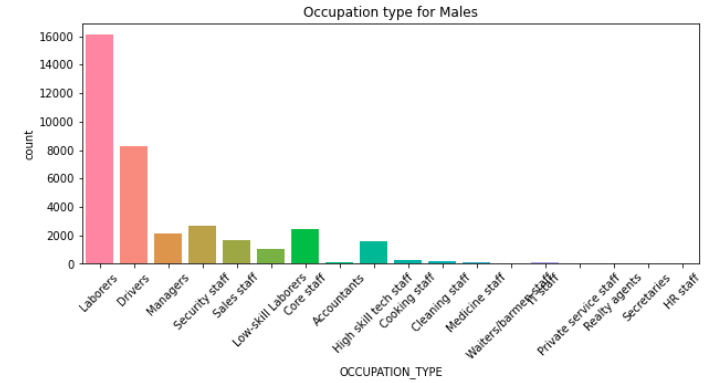
* 1. **Distribution of Applicants gender**



the number of Non-defalters 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

1. Results and Evaluation

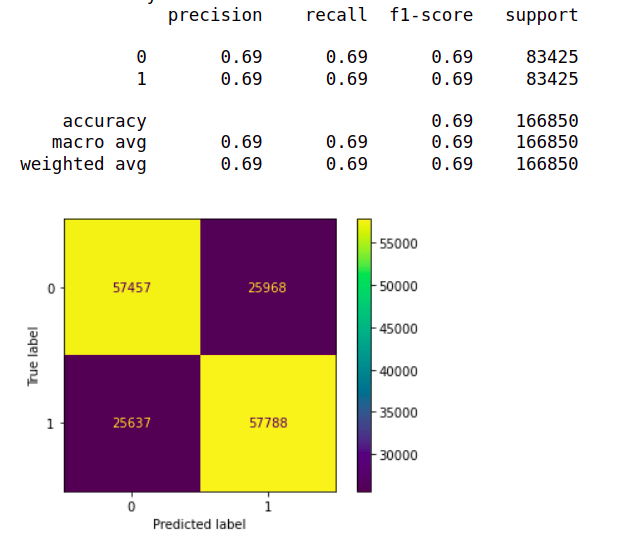
We trained the models to achieve best f1-score for model class 1 on validation set.

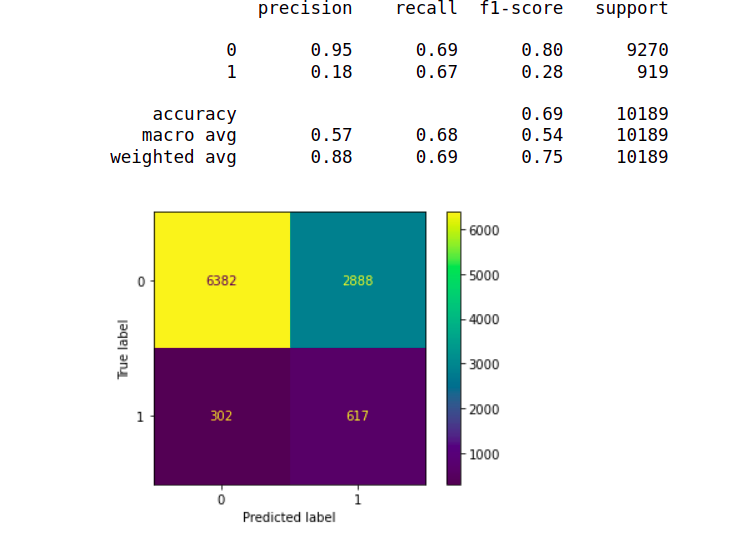
* The Logistic Regression model achieved on Over Sampled data the following scores

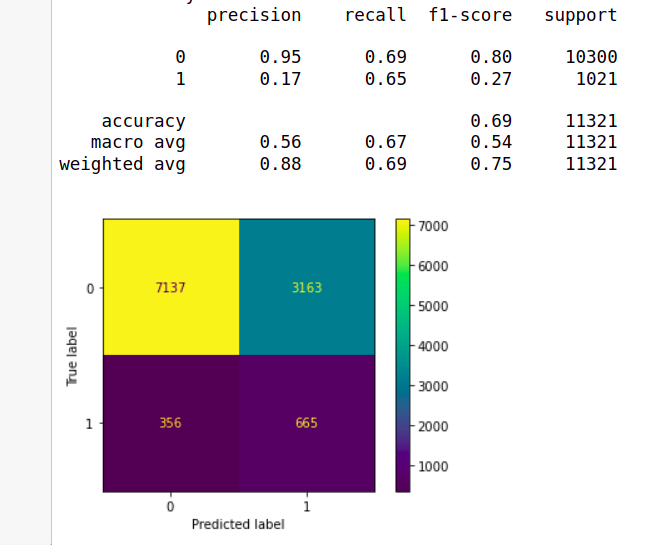
Train: accuracy = 69%, f1-score for class 1 = 69%, f1-score for class 0 = 69%

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%







* 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%

We Also Tried the following classifiers on Imbalanced Data, Under Sampled data, Over Sampled Data:

1. **Navie Bayes Impalanced**
2. Decision Tree
3. **KNN UnderSampled**
4. Support Vector Machine
5. Random Forest

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

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.