Project report PyCK

1. Introduction:

Problem statement: To automate the Credit Card eligibility process based on customer detail provided while filling online application form & Credit history of customer.

Problem is to identify the customer's segments which are eligible for Credit Card approval, so that we can specifically target these customers.

Importance:

From many applications filled for issuance of credit cards. Many of them are rejected for many reasons, like high-loan balances, low-income levels, or too many inquiries on an individual's credit report. Manually analyzing these applications is error-prone and a time-consuming process. Luckily, this task can be automated with the power of machine learning and pretty much every organization does so nowadays. In this project, we will be build an automatic credit card approval predictor using machine learning techniques which is efficient and fast.

Motivation:

The project topic includes the finance scenario, several machine learning modals and python libraries which will help us to get more knowledge of the domain. Since learning anything remains useless as long as we apply it in real life. Hence using this project we'll be applying many of those things which we've learnt so far during this course. In addition, we'll also be applying a few things which we've learned on our own (ML part) just to get our hands on these.

Hence the project will help us a lot in the future as well.

2. Implementation details:

- **Importing various libraries like pandas matplotlib seaborn numpy etc.**
- **Importing the dataset with application_record as a_rec and credit_record as c_rec:**

  Application record data: Application record Dataset contains 18 columns which are following :

  ID ; CODE_GENDER ; FLAG_OWN_CAR ; FLAG_OWN_REALTY ; CNT_CHILDREN ; AMT_INCOME_TOTAL ; NAME_INCOME_TYPE ; NAME_EDUCATION_TYPE ; NAME_FAMILY_STATUS ; NAME_HOUSING_TYPE ; DAYS_BIRTH ; DAYS_EMPLOYED ; FLAG_MOBIL ; FLAG_WORK_PHONE ; FLAG_PHONE ; FLAG_EMAIL ; OCCUPATION_TYPE ; CNT_FAM_MEMBERS

  Credit record data: Credit record Dataset contains 3 columns which are following :

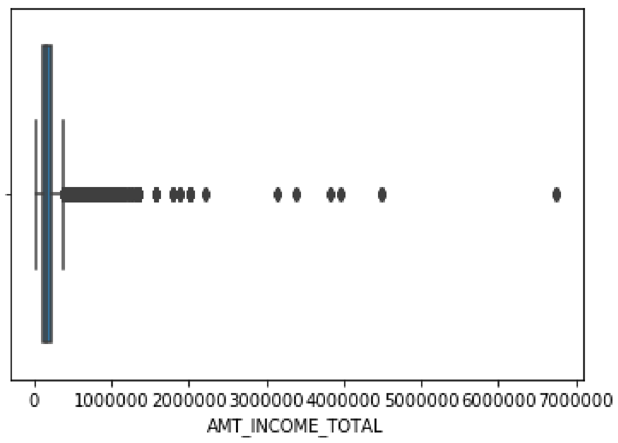
  ID ; MONTHS_BALANCE ; STATUS

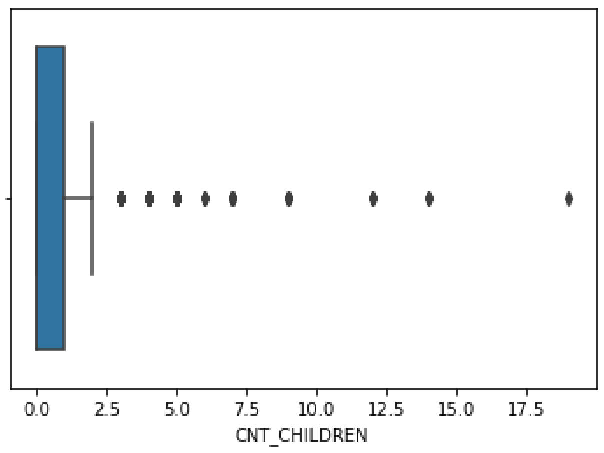
- **Checking for null values in the dataset and dropping the column which has missing values:**The cause of **missing values** can be data corruption or failure to record data. The handling of **missing** data is very important during the preprocessing of the dataset as many machine learning algorithms do not support **missing values** and so we drop rows corresponding to these values.
- **Dimension check after dropping the column :** since occupation_type column has maximum null value so we drop it and we get (438557, 17)
- **Printing columns with numerical data and categorical data:**

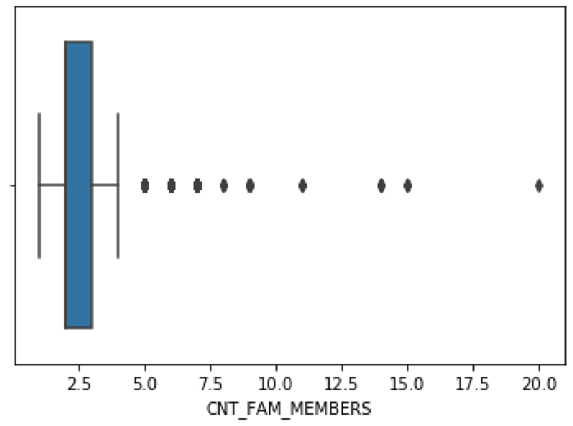
Numerical data columns'ID', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'FLAG_MOBIL', 'FLAG_WORK_PHONE', 'FLAG_PHONE', 'FLAG_EMAIL', 'CNT_FAM_MEMBERS'
Categorical data column 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE'

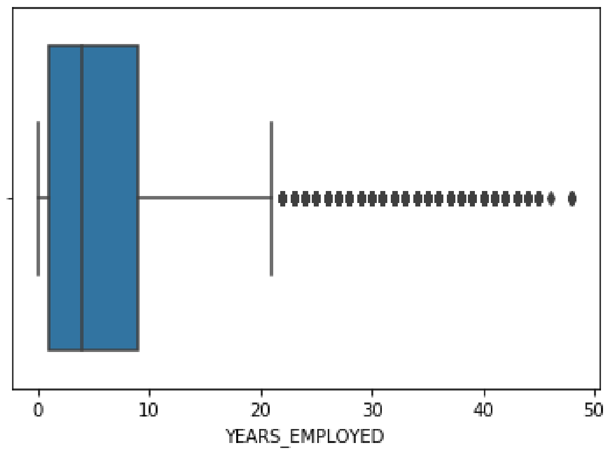
- **Checking for unemployed people and replacing there values with 0**

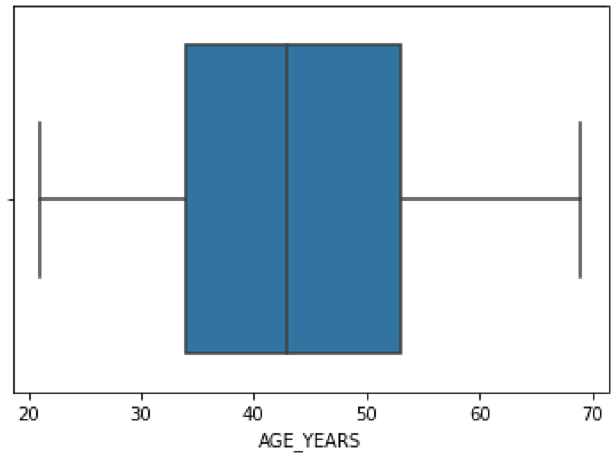
- **Checking counts of unique values in each column** if the column has a single unique values then removing the particular column here 'FLAG_MOBIL' all values as 1 so removing it
- **Checking count of unique values in credit_record file**
- **Categorizing status column to binary values** we 0 and 1 as the values 1 for approved and 0 for rejected
- **Normalizing c_rec dataset and assigning it to c_rec_new :**In this step, we normalize all the data of c_rec as our dataset contains many observations of different attributes having very high and low scale. **Normalization** is a technique often applied as part of data preparation for **machine learning**. The goal of **normalization** is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information
- **Dropping the months balance column :**The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on therefore this is not of our matter of interest and dropping out this column.
- **Visualizing the application record data:** after drawing the box plot we check for the outliers





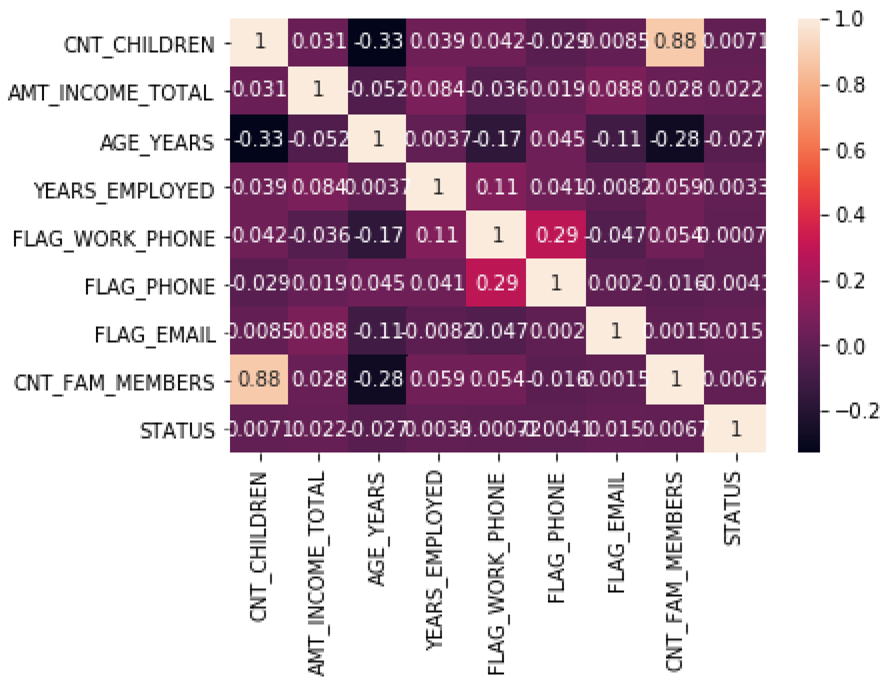






- **Removing the outliers :**An **outlier** is an object that deviates significantly from the rest of the objects. We remove the outliers from the a_rec_new dataset by the method of Box Plot after analysing the above box plots.
- **Merging the data frames and assigning to final_data :**As we have already cleaned out both of the datasets, a_rec and c_rec_new, now we are going to merge both the dataframes into one, named final_data for the modelling based on the column 'ID' .
- **Dropping the column 'ID' and after that duplicate records from final data:**Now ID column have only unique values so we drop this out from final_data. After that we again check for the duplicate values and drop them out.
- **Checking for final values in final data:**Here we observe what is the dimension of the final_data dataframe. (11186, 16)

- **Visualizing the final cleaned data** and drawing the correlation heatmap between the independent variables



- **Converting all non numerical data to numerical data:**Final_data contains many categorical variables and numerical variables. For the modelling of the dataset we are required to convert all the categories observations into numerical type.
- **Drawing correlation matrix of the final data:**In this step we try to observe correlation coefficients between the variables.
- **Assigning response and independent variables:**We assign all the columns of final_data dataframe to x as predictors except the column ‘STATUS’ and assign the ‘STATUS’ column to y as response variable

Now we are ready to fit the adequate ML models and their accuracies.

- **Fitting machine learning models and analyzing the accuracy of the models**

Logistic Regression: Logistic regression analysis is used to examine the association of (categorical or continuous) independent variable(s) with one dichotomous dependent variable.on applying this to our dataset we get

```
Logistic Model Accuracy :  78.41823056300268 %
```

Decision Tree classification: In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree

```
Decision Tree Model Accuracy :  73.05630026809652 %
```

Random Forest classification:*"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

```
Random Forest Model Accuracy :  78.41823056300268 %
```

Support Vector Machine classification: The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence, this algorithm is termed as Support Vector Machine.

```
Support Vector Classifier Accuracy :  76.2734584450402 %
```

K-Nearest Neighbour classification:K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

```
KNN Model Accuracy :  75.96067917783735 %
```

XGBoost classification:XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

```
XGBoost Model Accuracy :  75.69258266309204 %
```

- **Balancing the dataset** Apart from accuracy we get from the above models on our dataset a balanced data set for a model would generate higher accuracy models, higher balanced accuracy and balanced detection rate because balancing give equal priority to the classes. Hence, it’s important to have a balanced data set for a classification model.
- **Machine learning model after balancing**

Below are the accuracy of the various machine learning models after balancing and we from here we can get our best ML model for prediction with maximum accuracy:

```
Logistic Model Accuracy :  49.8005698005698 %
Decision Tree Model Accuracy :  75.18518518518519 %
Random Forest Model Accuracy :  77.20797720797721 %
Support Vector Classifier Accuracy :  47.77777777777778 %
KNN Model Accuracy :  44.900284900284895 %
XGBoost Model Accuracy :  83.73219373219372 %
```

- **Validation**:Model validation refers to the process of confirming that the model actually achieves its intended purpose. In most situations, this will involve confirmation that the model is predictive under the conditions of its intended use.So we apply below methods to to validate the models:

- K-Fold Cross Validation
- STRATIFIED SHUFFLE SPLIT

3. Conclusion:

After validating the given models for the given data set we predict that the best model for the credit cards approval prediction is XGBoost Model with accuracy of 83.73219373219372 %