A PRESENTATION ON

EDA OF A BANK RISK ANALYSIS

BY

VARSHA YADAV

BUSINESS PROBLEM

The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specializes in lending various types of loans to urban customers. You have to use EDA to analyze the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.

When the company receives a loan application, the company has to decide for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:

- If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
- If the applicant is not likely to repay the loan, i.e. He/she is likely to default, then approving the loan may lead to a financial loss for the company.

BUSINESS OBJECTIVES

This case study aims to identify patterns which indicate if a client has difficulty paying their installments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc. This will ensure that the consumers capable of repaying the loan are not rejected. Identification of such applicants using EDA is the aim of this case study.

In other words, the company wants to understand the **driving factors (or driver variables) behind loan default,** i.e. The variables which are strong indicators of default. The company can utilize this knowledge for its portfolio and risk assessment.

TOOLS WE HAVE

We are provided with two datasets, and one data dictionary.

- 1. **'application_data.csv'** contains all the information of the client at the time of application. The data is about whether a **client has payment difficulties.**
- 2. **'previous_application.csv'** contains information about the client's previous loan data. It contains the data whether the previous application had been **Approved, Cancelled, Refused or Unused offer.**
- 3. **'columns_description.csv'** is data dictionary which describes the meaning of the variables.

STEPS INVOLVED IN THE ANALYSIS

- ✓ Preparing the **JUPYTER NOTEBOOK** (importing important libraries, adjusting the view, ignoring the warnings)
- ✓ Reading the files
- ✓ Performing some sanity checks (shape, info, numerical information's)
- ✓ Data cleaning(searching for the null values, impute those , or drop those, delete unnecessary columns)
- ✓ Data manipulation (conversion of data types, taking absolute values of negative figures)
- ✓ Outliers checking
- ✓ Data imbalance checking
- ✓ Univariate analysis
- ✓ Segmented univariate analysis
- ✓ Bivariate analysis
- ✓ Merged data analysis
- ✓ Correlation between the variables
- ✓ Conclusion

BASIC STEPS OF AN ANALYSIS

To prepare the jupyter notebook, i have imported the NumPy, pandas, matplotlib, and seaborn libraries, As the datasets are very high to see the maximum rows and columns, we have set the display option. After doing it all and ignoring the warnings beforehand I have loaded the files into the python using **pd.Read_csv** command.

After that we Have performed some sanity checks, the shape of the data frames, head, info etc.

reading the files

```
app_df=pd.read_csv("application_data.csv")
```

```
app_df.head(10)
```

```
# printing the shape of the two datasets
print("application data shape:", app_df.shape)
print("previous application data shape:", prev_df.shape)
```

```
application data shape: (307511, 122)
previous application data shape: (1670214, 37)
```

checking the datatypes of all the column

```
app_df.info(verbose=True)
```

As shown in the pictures we have done the same for two of the datasets.

DATA CLEANING

After the basic operations are performed on the datasets now we should move to the next step which is data cleaning. This is the most important part of an EDA, this includes handling null values, deleting unnecessary columns etc.

```
#creating a new dataframe with the null value percentage along with the column name
null_app_df=pd.DataFrame(app_df.isnull().sum()/app_df.shape[0]*100).reset_index()
null_app_df.columns=["Column Name", "Null value percentage"]
null_app_df
```

	Column Name	Null value percentage
0	SK_ID_CURR	0.000000

We have calculated the null value percentages and created a separate data frame with the null percentage and the column names

Next we have set a threshold of 40% , any columns having a null percentage of that will not bw considering those in my analysis, and those can be dropped.

Calculations of nulls >40% and drop columns list are shown below.
Same is done for two of the datasets.

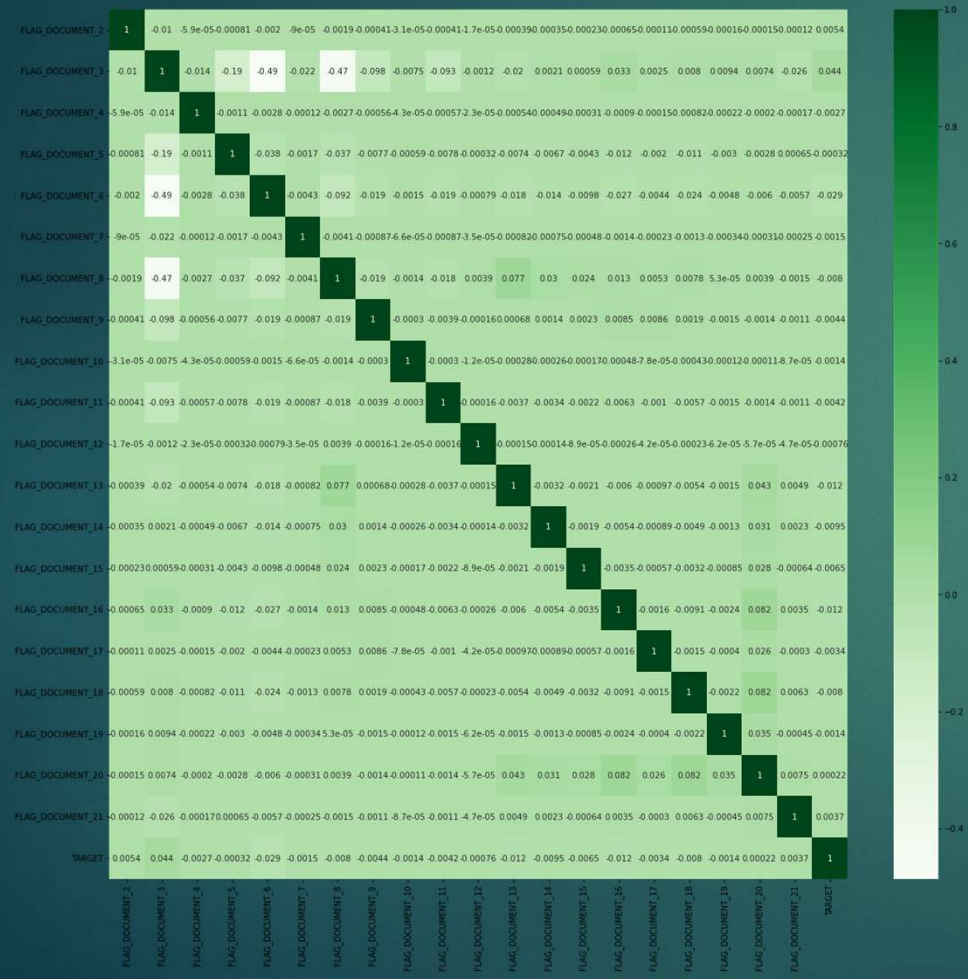
```
# checking the null value percentage that are above or equal to 40%
null_40_app=null_app_df[null_app_df["Null value percentage"] >= 40]
null_40_app
```

```
drop_app=list(null_40_app["Column Name"])
```

```
drop_app
```

```
['OWN_CAR_AGE',
 'EXT_SOURCE_1',
```

Next we have checked the correlation of some of the columns named flag_document_1, flaf_document_2,……,flag_document_21 with the target column and plotted a heatmap against that.



The deeper are the colors higher the correlations, as we can see, there no high correlation between the flag document columns and the target columns except for one flag document_3 column, so we can delete the rest of the columns.

In the drop list there are two more columns are included named "ext_source_2","ext_source_3", as they are not really useful according to me "ext_source_1" is already include as it has over 50% of missing values, and "ext_source_3" has 20% of null values, so these column can be deleted.

```
#adding the flag documents in the drop list
drop_app=list(null_40_app["Column Name"])+["FLAG_DOCUMENT_2","FLAG_D
                                           "FLAG_DOCUMENT_8","FLAG_DO
                                            "FLAG_DOCUMENT_14","FLAG_
                                            "FLAG_DOCUMENT_20","FLAG_
```

MANIPULATION

In the analysis I haven't dropped any columns, I have just added them in a list , and for application_data.Csv the drop column list is almost 70 and that of the previous_application_data.Csv is 11. Leaving these columns i have completed my case study.

After deleting the nulls, I have converted the data types to its suitable, for example, numeric to categorical , changing the negative values to positives by taking the absolute of it.

```
cat_col_app=["NAME_CONTRACT_TYPE","NAME_IN
for i in cat_col_app:
    app_df[i]=pd.Categorical(app_df[i])
```

```
days_app=["DAYS_BIRTH","DAYS_EM
for i in days_app:
    app_df[i]=abs(app_df[i])
```

Next I have categorize some new columns i.e., Created age buckets, income buckets, experience in their work field.

MISSING VALUE IMPUTATION

To move ahead in the analysis, there are columns with lesser percentage of missing values, so we can impute those missing values whose percentages are below 40% or so. There are few strategies to impute the missing values for CATEGORICAL columns and NUMERICAL columns.

- For categorical columns we are going to impute the missing values with the mode of it.
- For numerical columns we are going to use the mean/median approach.

FOR CATEGORICAL COLUMNS

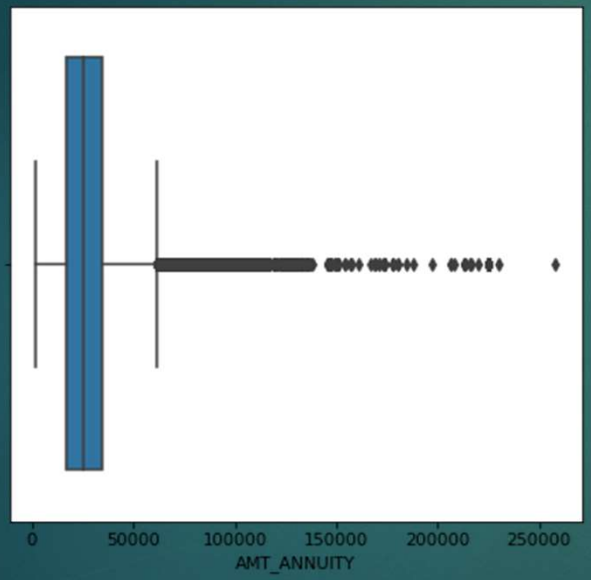
```
app_df["NAME_TYPE_SUITE"].fillna(app_df["NAME_TYPE_SUITE"].mode()[0],inplace=True)
```

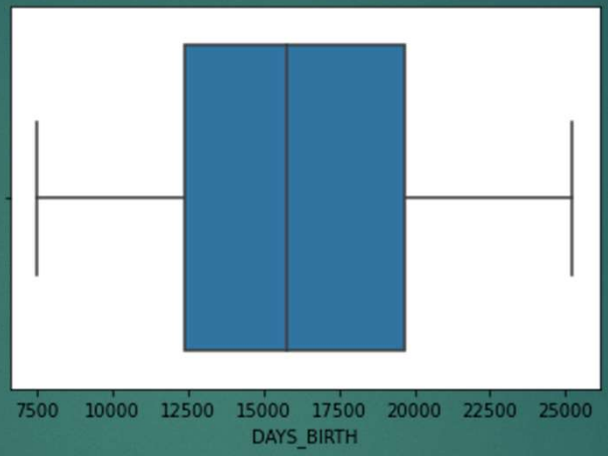
FOR NUMERICAL COLUMNS

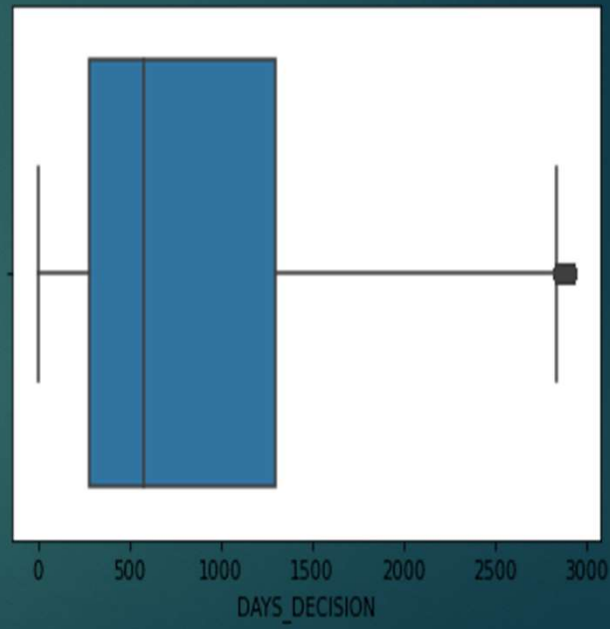
```
amt_list=["AMT_REQ_CREDIT_BUREAU_HOUR",
"AMT_REQ_CREDIT_BUREAU_DAY",
"AMT_REQ_CREDIT_BUREAU_WEEK",
"AMT_REQ_CREDIT_BUREAU_MON",
"AMT_REQ_CREDIT_BUREAU_QRT",
"AMT_REQ_CREDIT_BUREAU_YEAR"]
for i in amt_list:
    app_df[i].fillna(app_df[i].median(),inplace=True)
```

OUTLIERS CHECKING

Outliers are those which belong beyond the upper and lower bound.
These re the some results of outliers from the two different datasets, the observations are mentioned in details in the python coding.



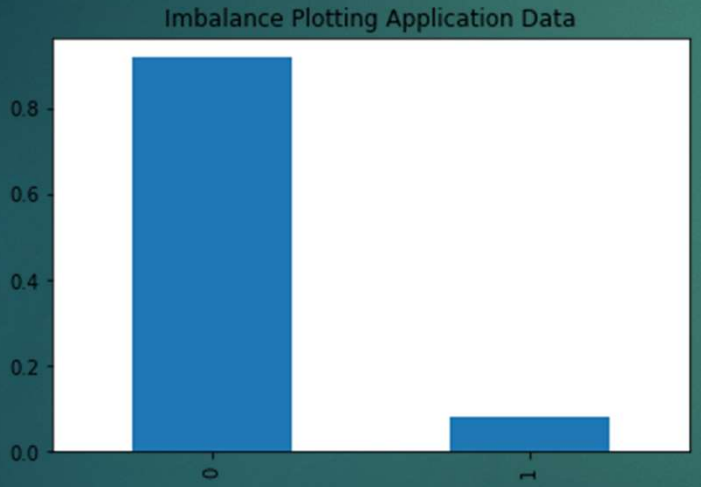




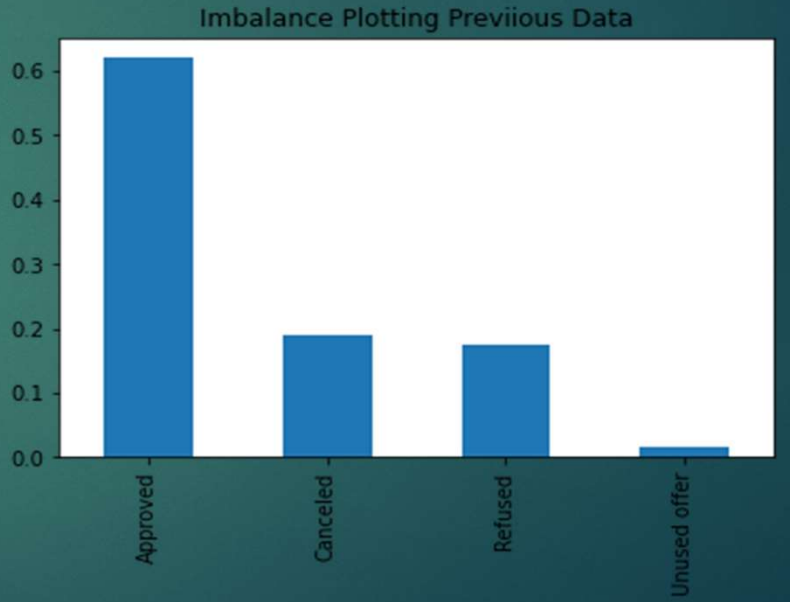
DATA IMBALANCE RATIO

Data imbalance ratio is nothing but the value counts of the target variable of the two dataset. It is plotted as follows:

```
app_df["TARGET"].value_counts(normalize=True).plot.bar()
plt.title("Imbalance Plotting Application Data")
plt.show()
```



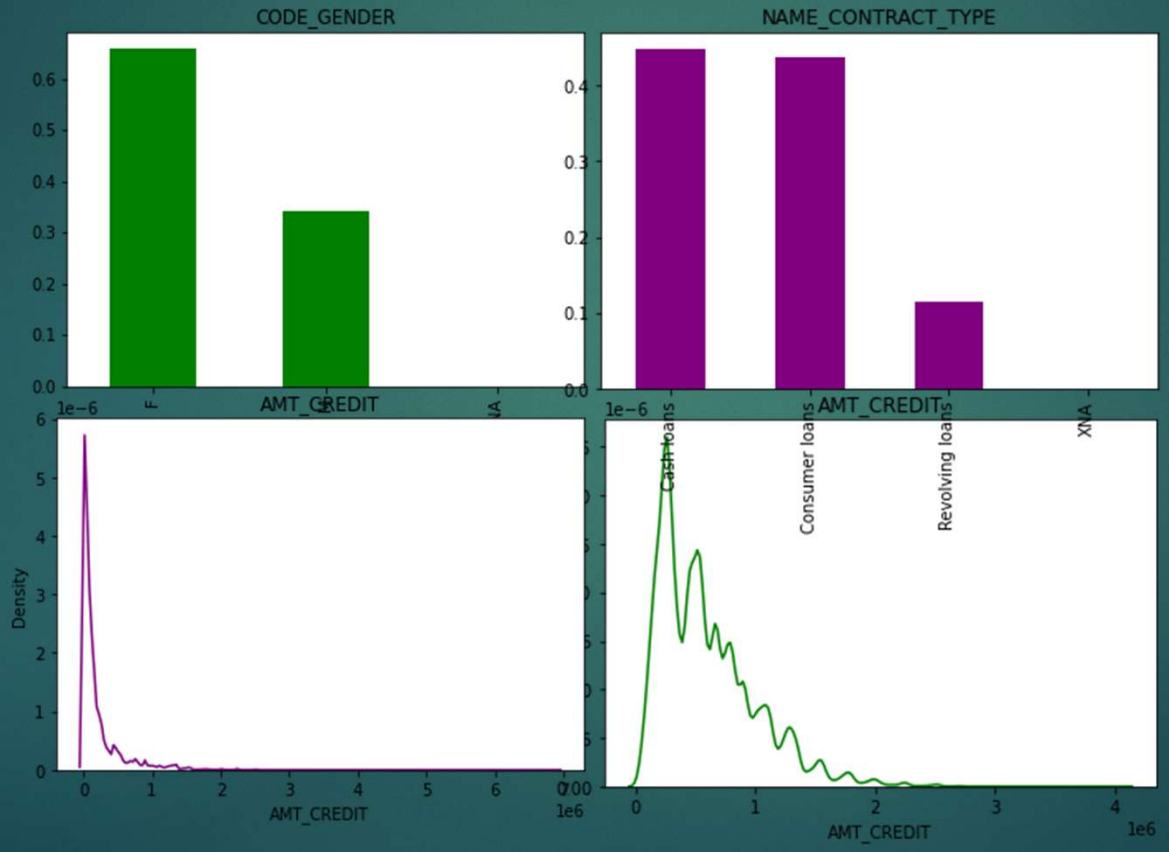

```
prev_df["NAME_CONTRACT_STATUS"].value_counts(normalize=True).plot.bar()
plt.title("Imbalance Plotting Previious Data")
plt.show()
```



UNIVARIATE ANALYSIS
FOR BOTH DATASETS

Uni refers to one, which means analyzing one variable at a time. There are two types of univariate analysis:

- CATEGORICAL
- NUMERICAL

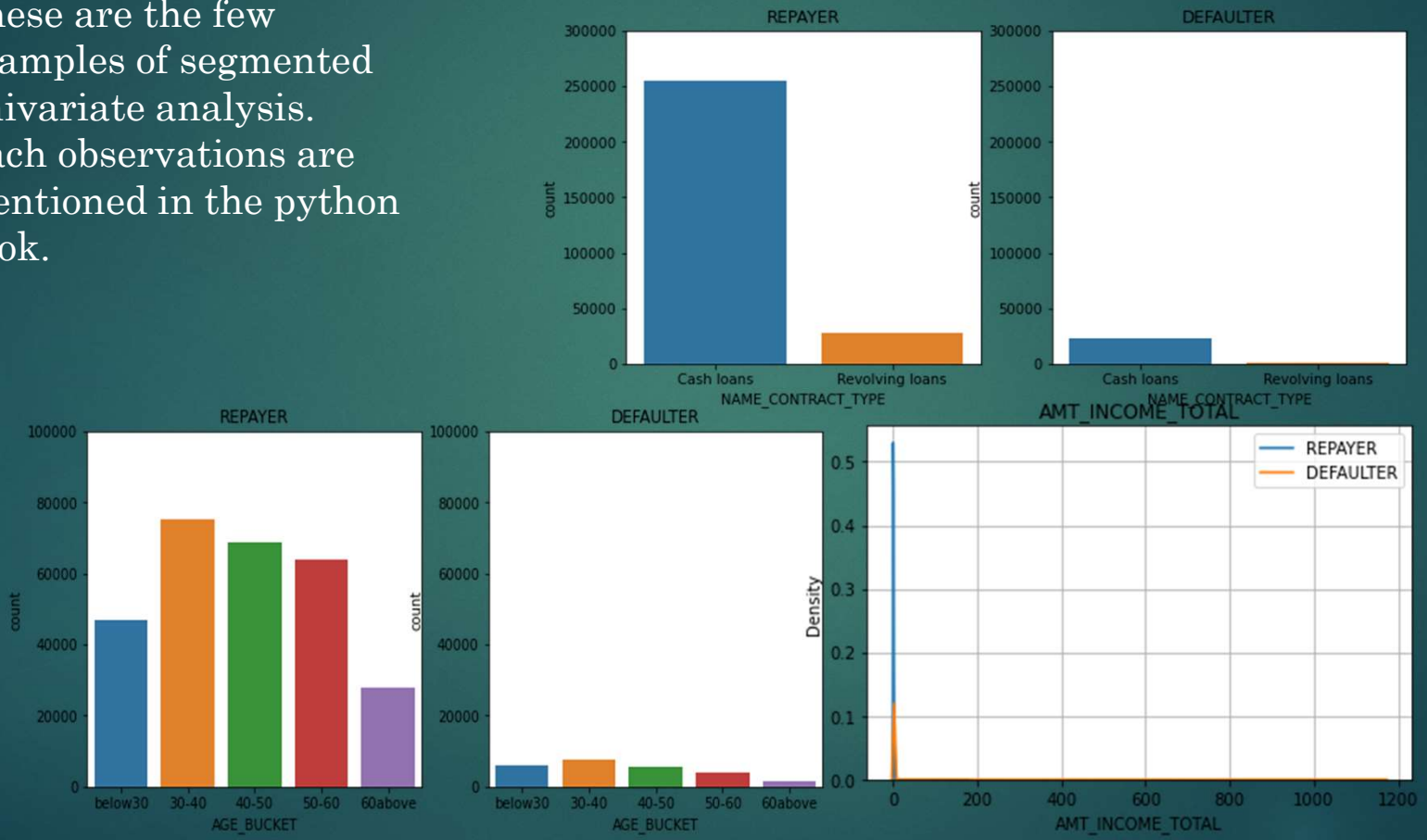


These are the few example of categorical and numerical univariate analysis, Details of each are mentioned in the python notebook.

SEGMENTED UNIVARIATE ANALYSIS

This means we have segmented the dataset based on the target variable and performed the univariate analysis on each segment.

These are the few examples of segmented univariate analysis. Each observations are mentioned in the python book.

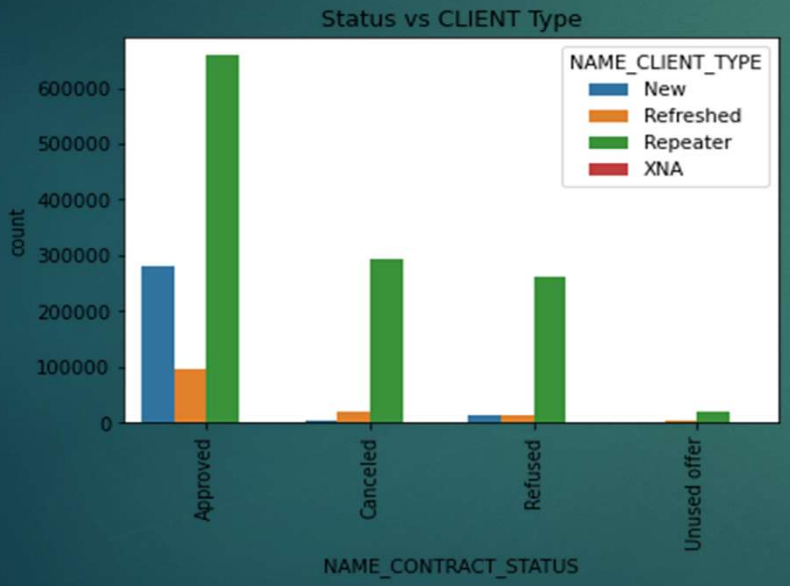


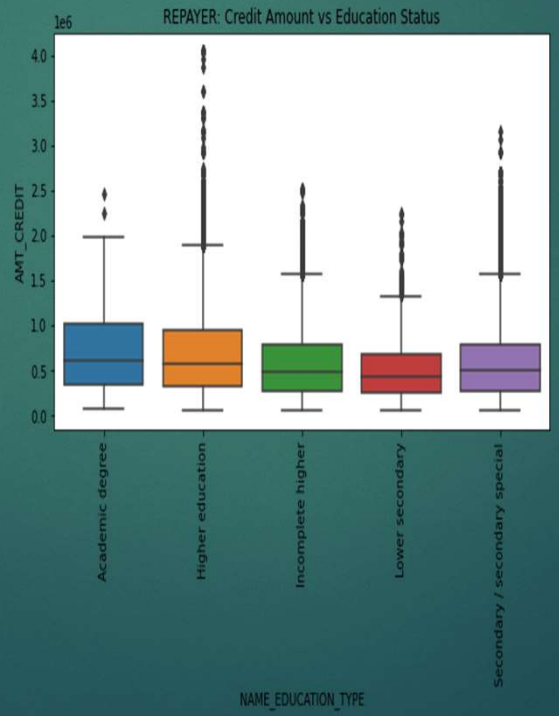
BIVARIATE ANALYSIS
FOR BOTH DATASETS

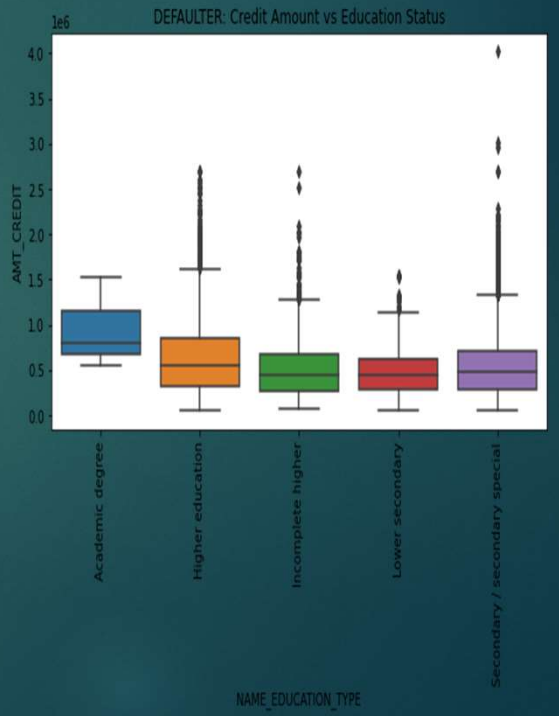
Bivariate analysis refer to analyzing with two variable at a time with the target variable.
It has three forms:

- NUMERICAL AND CATEGORICAL ANALYSIS
- CATEGORICAL AND CATEGORICAL ANALYSIS
- NUMERICAL AND NUMERICAL ANALYSIS

Below are some examples :







MERGING TWO DATASETS

This is where we merge two of the datasets (application_data.csv & previous_application_data.csv) on the basis of a common element present in the two datasets. We can merge the entire dataset, or we can select which columns to be merged for the analysis.
This is done as follows:

Merged Data Analysis

```
merged_df=pd.merge(app_df[["SK_ID_CURR","TARGET"]],
                   prev_df[["SK_ID_CURR","NAME_CONTRACT_
                   ,on="SK_ID_CURR",how="inner")
```

Next we have again created two separate data frame of repayers and defaulters.

```
merged_repayers=merged_df[merged_df["TARGET"]==0]
```

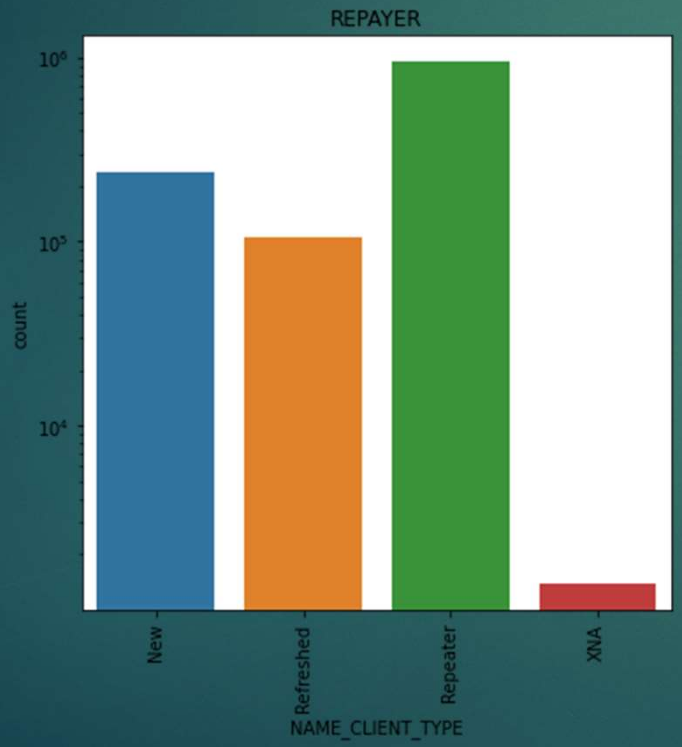
REPAYERS

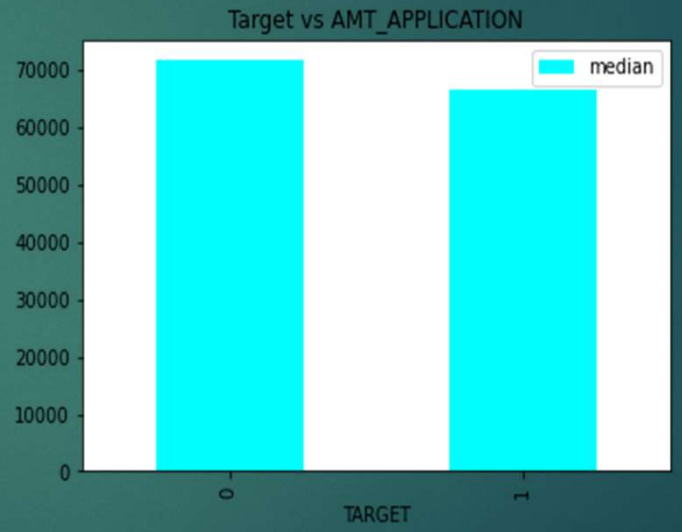
```
merged_defaulters=merged_df[merged_df["TARGET"]==1]
```

DEFAULTERS

MERGED DATA ANALYSIS

After separating the merged data frame, i have shown a few analysis with the target variable(only repayers).
Below are the some examples, observations of each mentioned in details in the python notebook.



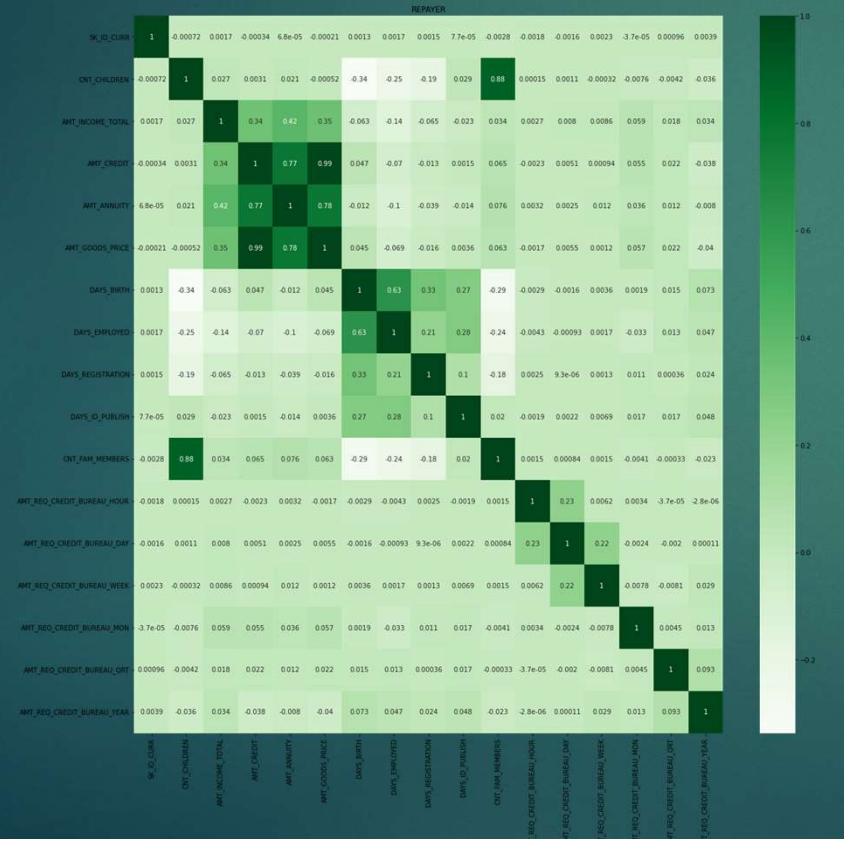


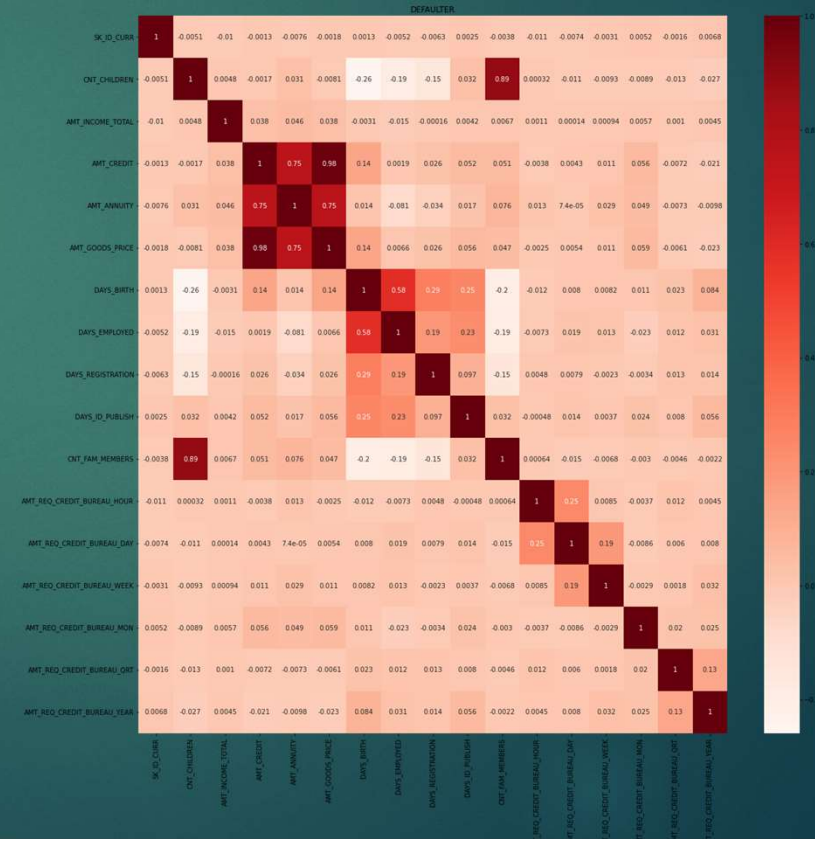
FINDING CORRELATION THROUGH HEATMAP

We have created two separate data frames to calculate the repayer's and defaulter's correlation as follows:

```
repayer_df=app_df.loc[app_df["TARGET"]==0, correlation]
defaulter_df=app_df.loc[app_df["TARGET"]==1, correlation]
```

REPAYERS CORRELATION





DEFAULTERS CORRELATION

TOP 10 HIGHEST CORRELATIONS

Variable 1	Variable 2	Correlation
AMT_CREDIT	AMT_GOODS_PRICE	0.99
CNT_CHILDREN	CNT_FAM_MEMBERS	0.88
AMT_ANNUITY	AMT_GOODS_PRICE	0.78
AMT_CREDIT	AMT_ANNUITY	0.77
DAYS_BIRTH	DAYS_EMPLOYED	0.63
AMT_ANNUITY	AMT_INCOME_TOTAL	0.42
AMT_INCOME_TOTAL	AMT_GOODS_PRICE	0.35
AMT_CREDIT	AMT_INCOME_TOTAL	0.34
DAYS_REGISTRATION	DAYS_BIRTH	0.33
DAYS_ID_PUBLISH	DAYS_EMPLOYED	0.28

INFERENCES FROM THE CORRELATION

REPAYERS:

Credit amount is highly correlated with

- Annuity amount
- Goods price
- Total income amount

There is a high correlation between

- Number of children and number of family members

We can also see that repayer have a high correlation with

- Days employed

DEFAULTERS:

- Loan annuity amount with credit resulted a little decrement when compared with the defaulters.(0.77 to 0.75)
- Annuity amount with total income amount resulted a high decline in defaulter. (0.42 to 0.096)
- Total income amount with goods price and credit amount also resulted a decline when compared with defaulters. (0.35 to 0.038)

CONCLUSION

DECISIVE FACTOR FOR REPAYER:

- NAME_EDUCATION_TYPE: Academic degree has less defaults.
- NAME_INCOME_TYPE: Student & businessman have no defaults.
- DAYS_BIRTH: People of age group above 50 are very unlikely to default.
- DAYS_EMPLOYED: People having experience over 40 years are very less in defaulters.
- CNT_CHILDREN: People having zero to two children tend to repay the loan.

DECISIVE FACTOR FOR DEFAULTER:

- CODE_GENDER: Men are relatively higher at default.
- NAME_FAMILY_STATUS: Civil Marriage or single people default more.
- NAME_EDUCATION_TYPE: People with lower secondary and secondary education default more.
- OCCUPATION_TYPE: Low-skilled workers, drivers, security staff, laborers have a huge default rate.

THANK YOU