**Team 10 - AAS end module exam**

**Team members:**

220940325067 Shruti Kishor Patil

220940325006 Ajay Prakash Fatpure

220940325057 Ruchira Rajesh Bhargave

220940325032 Gaurav Rajan Bopardikar

220940325068 Shubham Shrivastava

**Data reading and understanding:**

Application\_data.csv

* There are 307511 rows and 122 columns in this dataset
* There are 122 columns with different data types like float, integer and object with 307511 rows
* Describe command *“df.describe()*” gives the statistical information of the dataset such as mean, count, min and max values, standard deviation and IQR ranges.
* df.info(all) #gives all information about the dataset like column name and data types

Previous\_application.csv

* There 1670214 rows and 37 columns in this dataset
* There are 37 columns with different data types like float, integer and object with 307511 rows
* Describe command *“preapdf.describe()*” gives the statistical information of the dataset such as mean, count, min and max values, standard deviation and IQR ranges.

**Cleaning dataset:**

→ Dropping null values:

1. Checking for columns whose null values are greater than 40% and dropping them
2. There are 49 such columns
3. After dropping them we have 73 columns remaining

→ Dropping irrelevant columns

* Notice that there are columns having 40% and more missing values. When dealing with columns, you have two simple choices - either delete or retain the column. If you retain the column, you'll have to treat (i.e. delete or impute) the rows having missing values.
* If you delete the missing rows, you lose data. If you impute, you introduce bias.
* Apart from the number of missing values, the decision to delete or retain a variable depends on various other factors, such as:
* The analysis task at hand, The usefulness of the variable (based on your understanding of the problem), The total size of available data (if you have enough, you can afford to throw away some of it) etc. Thus, for this exercise, let's remove the columns having more than missing values and which are not necessary for our analysis.
* For column AMT\_REQ\_CREDIT\_BUREAU\_HOUR, We can see that most of the values are equal to 0. So removing mode will give output as 0
* Same like this AMT\_REQ\_CREDIT\_BUREAU\_DAY, df.AMT\_REQ\_CREDIT\_BUREAU\_WEEK, df.AMT\_REQ\_CREDIT\_BUREAU\_MON, df.AMT\_REQ\_CREDIT\_BUREAU\_QRT, df.AMT\_REQ\_CREDIT\_BUREAU\_YEAR has the mode as 0, and hence the null values for these columns are imputed with mean
* Now we check for the columns having more than 0% of null values.
* From the columns dictionary we can conclude that only 'OCCUPATION\_TYPE', 'EXT\_SOURCE\_3 looks relevant to the TARGET column.
* We found that occupation type has 31% of null values
* It is observed that the majority of occupation types are for laborers.
* For EXT\_SOURCE\_3, It is better we don't replace any value for EXT\_SOURCE as the missing percentage is very high.
* For AMT\_GOODS\_PRICE, We can see that mode and median are almost the same i.e 45000. Mean and median are very close to each other. Hence, we can use it for imputation. The null values are filled with mode.
* Checking the NAME\_TYPE\_SUITE:
  + NAME\_TYPE\_SUIT is categorical type of data, to check the count in each category we have used the count function which gives the conclusion that unaccompanied has the highest count.
  + We can observe that 81% of the times the loan applicant was Unaccompanied So we can impute the missing value by unaccompanied.
* For, OBS\_30\_CNT\_SOCIAL\_CIRCLE,
  + Median and mode are same i.e '0'
  + Till 50th percentile all the values are 0, above it there are two outliers
  + Mean and mode are closer, so we can use it for imputation (with 0).

# 

# **Treating errors in Data types and data**

To analyze data types and values of other columns:

* The df.info command will give the detailed information about the data types of the column.
* While analyzing the **gender column** we observed that there are some null values. As per the analysis the count percentage is around 65% for the female loan applicants. Hence **we can impute** the NA data with 'F'.
* There are some **columns** with the **negative values** which is not feasible as the number of days cannot be negative. By using the **abs function** we converted the negative data into positive/absolute form.
* While analyzing the column **ORGANIZATION\_TYPE**, there **18% values are null/NA**. We cannot impute by mean/mode because then the result may get biased. We created a new category as **“unknown”.**

# **Converting all the Days column to Year**

* As analysis of data in days will be inappropriate so lets convert it into years.
* Using command
* Days\_column = [column for column in df if column.startswith('DAYS')] df[Days\_column] = df[Days\_column]/365 df[Days\_column].describe()
* After that we are renaming the column using the command df.rename.

# **Data cleaning conclusion:**

The data is cleaned where null values and columns with single values and duplicate values were handled. We calculated the percentage of null values. The columns with more than 30% of null values were dropped. Likewise we also dropped columns having missing values less than 30%.

We also checked for the columns which can have the possibility of duplicates and fortunately there were no duplicate data for ID’s.

The values which were negative were converted into positive variables according to their interpretation like employed days and registration days were changed to positive .

# **Dealing with Outliers**

* Box plot of the AMT\_INCOME\_TOTAL to get the idea of outlier. We can check the value using command sns.boxplot(df.AMT\_INCOME\_TOTAL) plt.show().
* After that we are checking the spread of AMT\_INCOME\_TOTAL using command :-

df.AMT\_INCOME\_TOTAL.quantile(q=[0.25,0.5,0.75,0.95,0.99,1])

* After checking the spread we can see a large value. We can check the value using command :-

Max\_Annual\_income =df[df.AMT\_INCOME\_TOTAL == df.AMT\_INCOME\_TOTAL.max()]

* From above two results we can infer that this is an outlier since the person is a laborer and the target Variable is 1. We will remove this row as we don't want this data to impact our analysis

df3 = df[df.index!=12840]

* After this we are finding outlier on WORK\_EXPERIENCE column using command:-

sns.boxplot(df3.WORK\_EXPERIENCE)

plt.show()

* In the above boxplot we can see work\_experience as more than 1000 which is not practically possible.
* We can see the value using command df3[df3.WORK\_EXPERIENCE>999]
* From the box plot we can see the values above 1000 WORK\_EXPERIENCE are outliers.

After that Finding outlier on AMT\_ANNUITY column using command :-

sns.boxplot(df3['AMT\_ANNUITY'])

plt.show()

* In the above boxplot we can observe there is one value which is above 2500000. We can check that value using command :-

df3[df3.AMT\_ANNUITY>250000]

* By observing the AMT\_CREDIT and AMT\_INCOME\_TOTAL we can say that AMT\_ANNUITY is not a outlier. After that Finding outlier on YEARS\_REGISTRATION column using command:-

sns.boxplot(df3['YEARS\_REGISTRATION'])

plt.show().

* We can check the value where YEARS\_REGISTRATION is greater than 65 using command:-

df3[df3.YEARS\_REGISTRATION>65]

These values can be considered as outliers.

* From boxplot we can observe that there is one value which is above 2500000.

We can check that value using command dataframe[dataframe.AMT\_ANNUALLY>250000].

By observing the values of AMT\_CREDIT and AMT\_INCOME we can conclude that AMT\_ANNUALLY is not an outlier.

* We try to find outliers of YEARS\_REGISTRATION.

Using boxplot we get an upper fence which is 65. So values greater than 65 can be considered as outliers of YEARS\_REGISTRATION.

## **Binning the values for AMT\_INCOME\_RANGE**

* Using binning we can convert numeric values to categorical or to sample numeric values. Binning makes it easier to analyze continuous variables.
* So here we are binning the column AMT\_INCOME\_RANGE. For this we created 5 categories as

0-0.2 (values) - Extremely Low (labels)

0.2-0.5 - Low

0.5-0.8 - Medium

0.8-0.95 - High

0.9 -1 - Extremely High

For binning we use the command:

dataframe['AMT\_INCOME\_RANGE'] = pd.qcut(dataframe.AMT\_INCOME\_TOTAL, q=[0, 0.2, 0.5, 0.8, 0.95, 1], labels=['EXTREMLY\_LOW', 'LOW', "MEDIUM", 'HIGH', 'EXTREMLY\_HIGH'])

## **Binning the values for AMT\_CREDIT\_RANGE**

* Now we are binning AMT\_CREDIT\_RANGE as per categories

0-0.2 (values) - Very Low (labels)

0.2-0.5 - Low

0.5-0.8 - Medium

0.8-0.95 - High

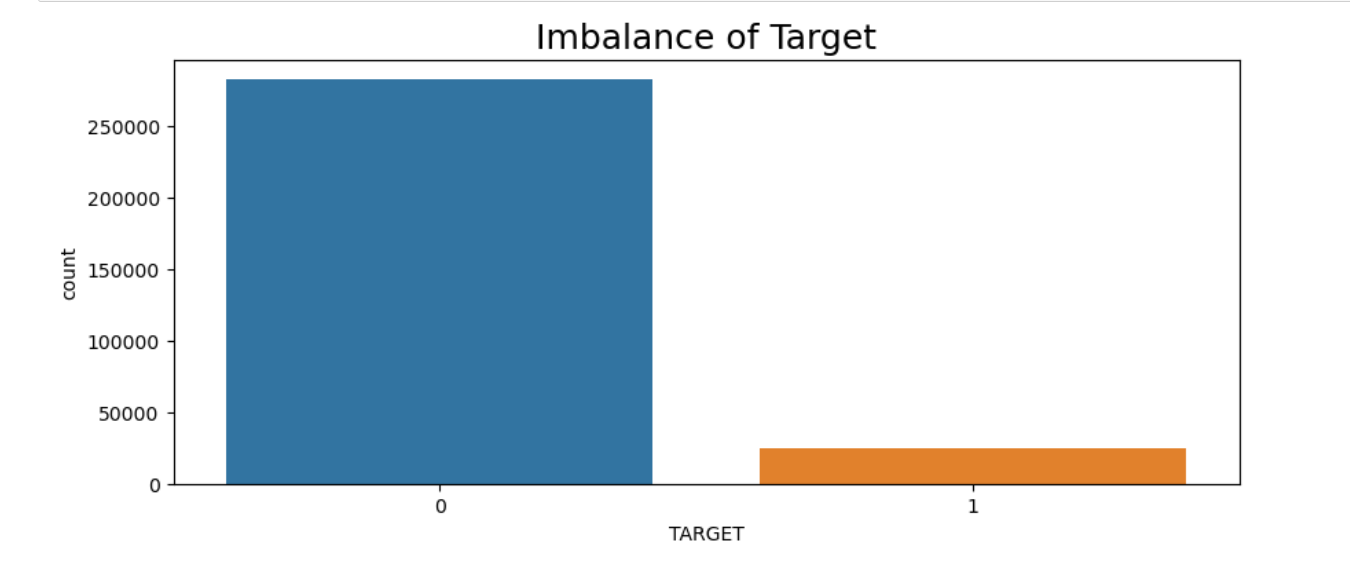
0.9 -1 - Very High

For binning AMT\_CREDIT\_RANGE, we use this command:

dataframe['AMT\_CREDIT\_RANGE'] = pd.qcut(dataframe.AMT\_CREDIT, q=[0, 0.2, 0.5, 0.8, 0.95, 1], labels=['VERY\_LOW', 'LOW', "MEDIUM", 'HIGH', 'VERY\_HIGH'])

# **Data Imbalance for target variable**

* Unbalanced dataset is one in which the target variable has more observations in one specific class than others. Using value\_counts() command, we could clearly see that there is imbalance in TARGET variable. So to support our inference, we plotted a bar graph.

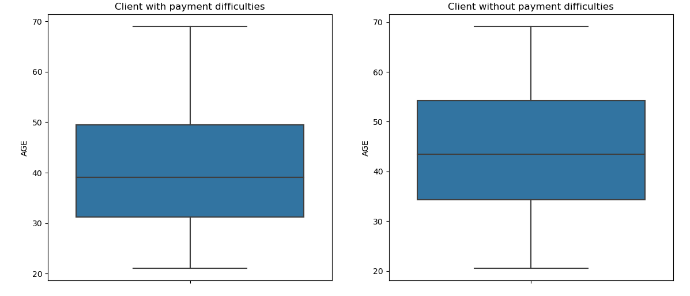


* We divided the dataset into two dataframe, one for client's with payment difficulties(1) which is 8% and other where clients are not facing difficulty(0) which is 91%.

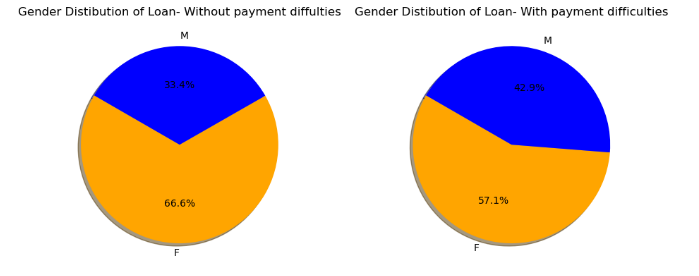
Using .shape command also we can see that there is an imbalance in the TARGET variable.

**Conclusion(dataset 1):**

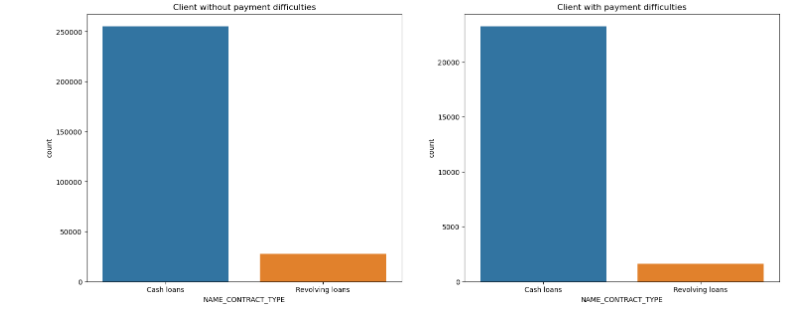
1. **By observing the boxplot, we can infer that Client with payment difficulties are in range of 31-49, whereas client with no payment difficulties are in range 34-54**

****

1. **If we consider the gender distribution, females approaches for loan in high percentage than the male.**

****

1. **Cash loans are the highest as compared to the recovered loans in both the cases of no payment and payment difficulties.**

****

**Univariate Analysis**

**Observation:**

→The most important column in deciding target variable could be Total income of applicant

* Total income is higher for client without payment difficulties as compared to client with payment difficulties

→The age group of applicants

* By observing the boxplot, we can infer that Client with payment difficulties are in range of 31-49, whereas
* client without payment difficulties are in range 34-54

**Categorical Variable**

* we observe that Females have more Loan payment difficulties as compared to Male's

→ Based on occupation type:

* We can clearly observe that laborers are the highest in both categories i.e clients with and without payment difficulties.

→ Based on name\_income\_type

* Pensioner and Govt Employees have better on-time payments.

**Bivariate Analysis**

→ Plotting for Income type across gender

* Female applicants have more difficulties in payment as compared to male applicant's
* Applicant's who are businessman and student's pay their loan on time although there count is low

→ Plotting for Income type across Education type

* Applicant's who have higher education have less difficulty in paying loan as compared to Secondary/secondary special.

**Cleaning previous\_app dataset**

* Dropping the columns with missing values more than 40%
* Checking out the values percentage in AMT\_APPLICATION
  + As we can see there are approx 23% values which are 0 in the AMT\_APPLICATION column. We need to remove it as it may affect the EDA process
  + Box plot observation for AMT\_APPLICATION: As we have seen there are no null values in AMT\_APPLICATION column but there are some values which are 0. we need to replace them with another value. There are some outliers also present which can affect the analysis. it is better to remove them.
  + Handling the outliers by taking the first 99th percentile values, we can remove those rows which have AMT\_APPLICATION value as 0.
* Checking out the outliers for AMT\_CREDIT in DataFrame using boxplot, we can observe that there are some outliers present in the data and also some null values present in data.
  + Outliers are handled by taking the 99th percentile of the data
  + Null values are imputed by median

# 

# **UNIVARIATE ANALYSIS(dataset2)**

**Univariate Analysis based on different columns:**

* data consists of only one variable. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes.
* As we made a pie chart based on the NAME\_CONTRACT\_TYPE , we can say that maximum customer approx 57% demands for Consumer Loans and only 7% customer demands for Revolving Loans which is the least in all.
* As per NAME\_CONTRACT\_STATUS we can conclude the out of all the loans applied for only 1% of the loans are canceled , refused percentage are 20% and maximum i.e 78% loans are approved.

# **Bi-VARIATE ANALYSIS**

* data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables.
* As per the bar graph plotted between two columns NAME\_CONTRACT\_STATUS vs NAME\_CLIENT\_TYPE, we can make inference that chances of approving or rejecting the load request is observed maximum in the regular/repeat customers.
* Other client types Loan refused rate for new clients or refreshed clients show the similar behavior.

# **MERGING BOTH THE DATAFRAMES**

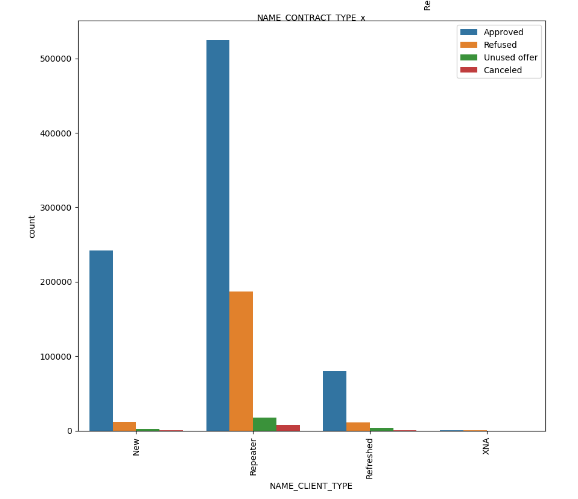
* Both the datasets are merged.
* Columns are checked and non-useful columns are dropped. We have total 38 columns and 1105806 rows in merged dataframe

Univariate analysis on categorical data:

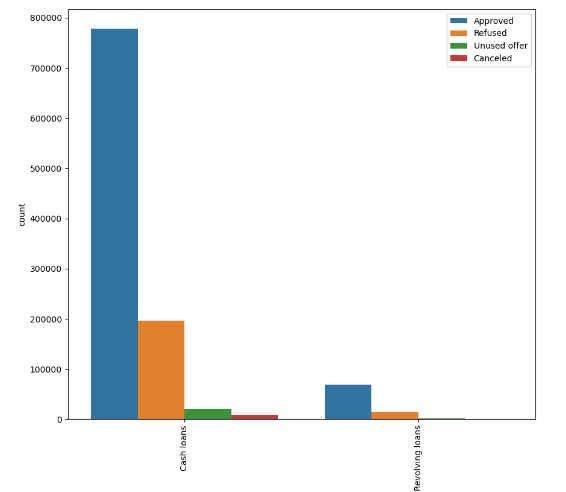
* "NAME\_CONTRACT\_TYPE\_x","NAME\_CONTRACT\_TYPE\_y","NAME\_CLIENT\_TYPE","NAME\_INCOME\_TYPE","NAME\_EDUCATION\_TYPE","OCCUPATION\_TYPE" are the columns with categorical data.
* **Observation:** Loan approval rates for Consumer Loans are much higher than any other loan. Banks like to give loans to the Repeaters. People with Secondary Education or more receive loan approval easily. Occupation\_type Laborers get more loans than others. Working class people receives more loan approvals than any other Income\_type

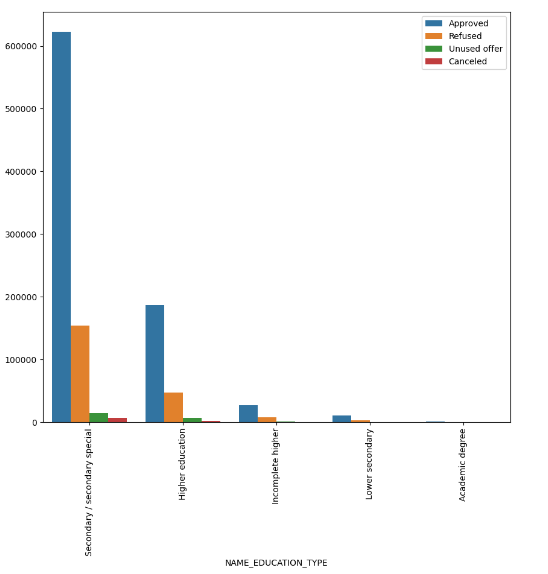
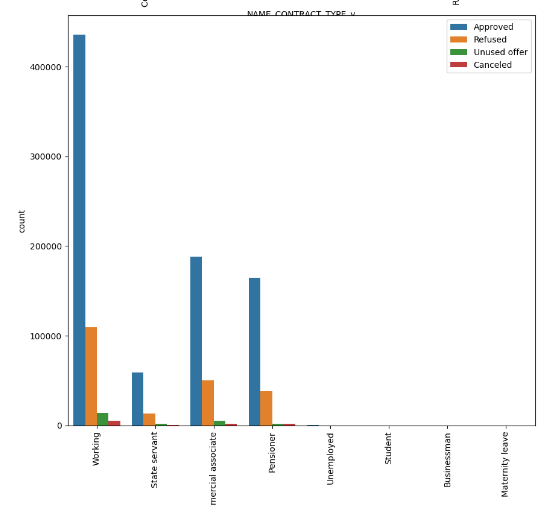
**Conclusion:**

1. Banks have the highest demand for loans from repeat customers.



1. Banks should focus on the cash loans as compared to the revolving loans as the number of applicants are more with respect to cash loans.



1. There is higher demand for loans from the working group and for higher education from the students
2. Banks should focus on providing low interest rates to the married and working class people since these two categories are the top most category which is facing the highest payment difficulties.

**Models that can be used:**

**Decision Tree**

Decision trees Can predict the output, with internal nodes representing the whole dataset and leaf nodes being the predictions.

**Random Forest**

Combination of various Decision trees can be used for prediction using Random Forest, However the computation time required to train such Random Forest Model is greater as compared to Decision tree but the prediction and the accuracy will be greater than decision tree.

**Binary Logistic Linear Regression**

As the output of the model is in 0 and 1 as repayer and Defaulter, Binary Regression can predict the output as either customer can be Repayer or defaulter based on its data given.