EDA Credit Assignment

Problem Statement

- This case study aims to identify patterns which indicate if a client has difficulty paying their instalments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending at a higher interest rate, etc.

- The company wants to understand the driving factors behind loan default.

  i.e. the variables which are strong indicators of default. The company can utilise this knowledge for its portfolio and risk assessment.

Notebook: Final Description

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 on whether the previous application had been Approved, Cancelled, Refused or Unused offer.

3. Final.ipynb file is a file worked on 2 files individually and merged.

Problems to Explain

- The results of univariate, bivariate analysis, etc.
- Handle missing values.
- If outlier, then why it is an outlier.
- Find the top 10 correlation for the Client (Defaulter & Non-Defaulter)

Overall Approach

1. Identifying the important variable.
2. Identifying the missing data and using appropriate method to deal with it.
3. Removing columns/or replacing it with an appropriate value.
4. Univariate and Bivariate analysis

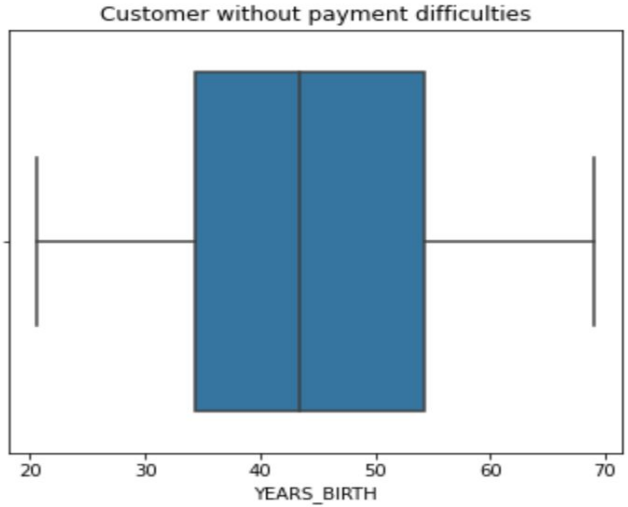
Results - Univariate Analysis for Numerical Variable.

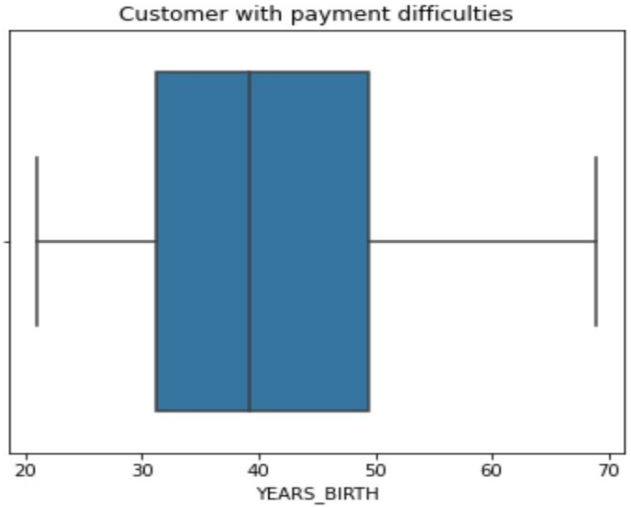
```
# YEARS_BIRTH

plt.figure(figsize=(13,5))

plt.subplot(1,2,1)
ax = sns.boxplot(target0['YEARS_BIRTH'])
plt.title('Customer without payment difficulties')

plt.subplot(1,2,2)
ax = sns.boxplot(target1['YEARS_BIRTH'])
plt.title('Customer with payment difficulties')
plt.show()
```





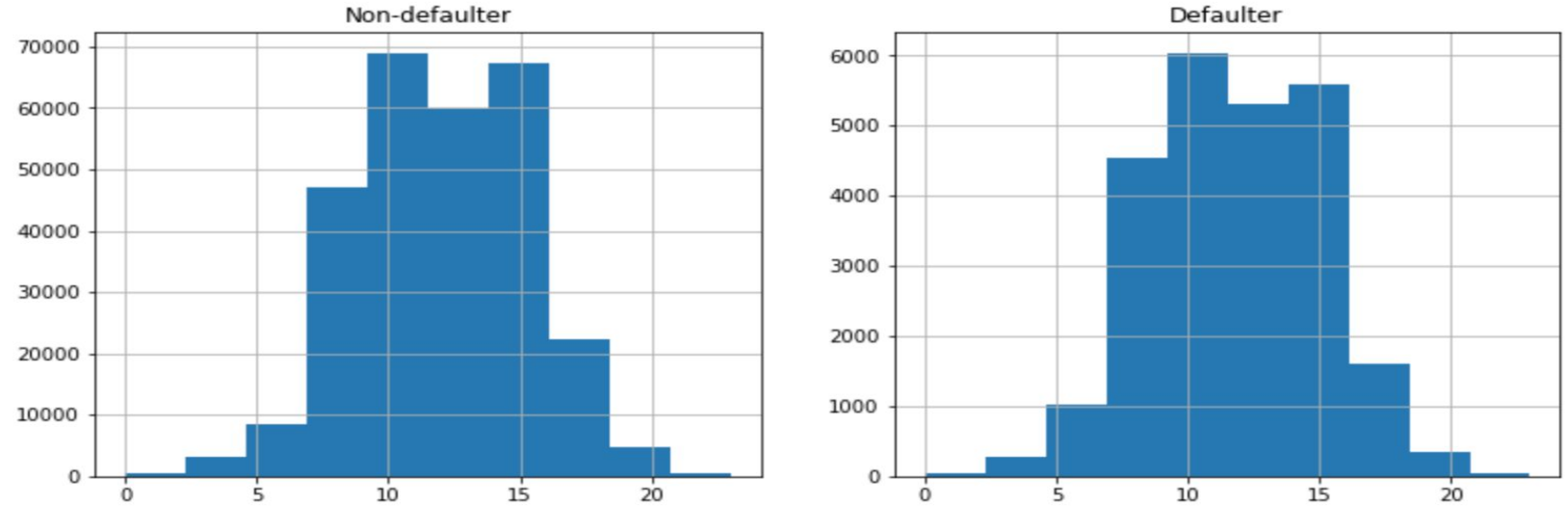
Results- Univariate Analysis for Categorical Variable**.**

```
fig = plt.figure(figsize=(13,5))

ax1 = fig.add_subplot(1, 2, 1, title="Non-defaulter")
ax2 = fig.add_subplot(1, 2, 2, title="Defaulter")

appl_df[appl_df["TARGET"] == 0]["HOUR_APPR_PROCESS_START"].hist(bins=10, ax=ax1)
appl_df[appl_df["TARGET"] == 1]["HOUR_APPR_PROCESS_START"].hist(bins=10, ax=ax2)

plt.show()
```



- In the first figure, we have plotted a box plot w.r.t. Target variable as important variable showing customer or clients with and without payment difficulties between 20 to 70 years.

- In the second figure , we have plotted a histogram keeping Target variable as primary concern showing approximately at what hour did the client apply for the loan.

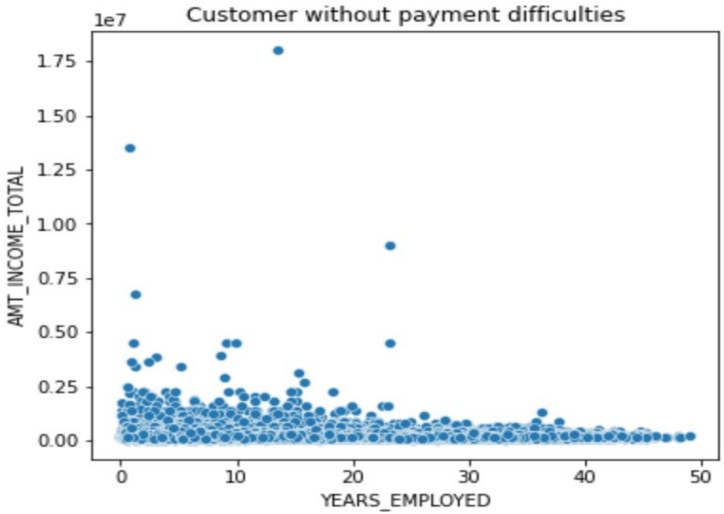
Results- Numerical-Numerical Bivariate Analysis

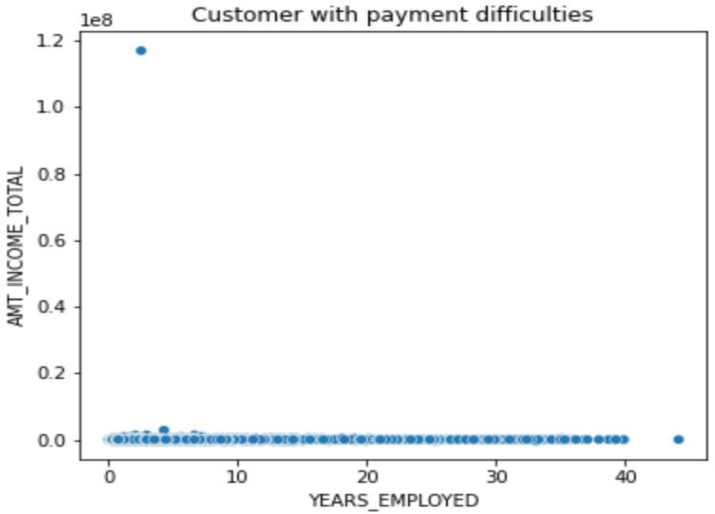
```
# YEARS_EMPLOYED & AMT_INCOME_TOTAL

plt.figure(figsize=(13,5))

plt.subplot(1,2,1)
ax = sns.scatterplot(data=target0[target0['YEARS_EMPLOYED']<1000], x='YEARS_EMPLOYED',y='AMT_INCOME_TOTAL')
plt.title('Customer without payment difficulties')

plt.subplot(1,2,2)
ax = sns.scatterplot(data=target1[target1['YEARS_EMPLOYED']<1000], x='YEARS_EMPLOYED',y='AMT_INCOME_TOTAL')
plt.title('Customer with payment difficulties')
plt.show()
```





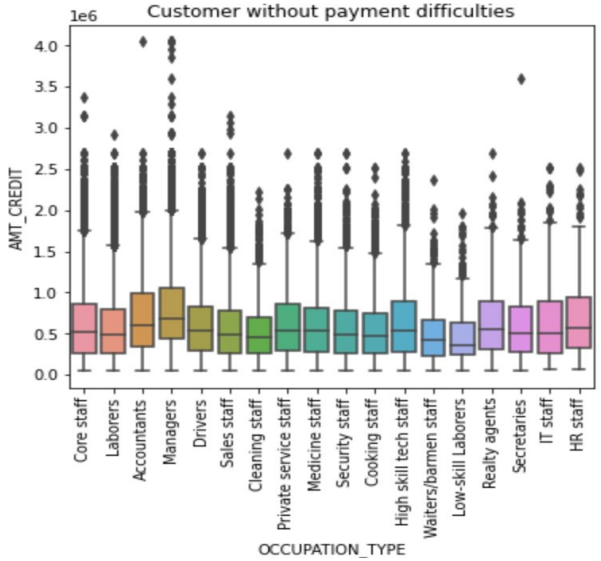
Results- Numerical - Categorical Bivariate Analysis

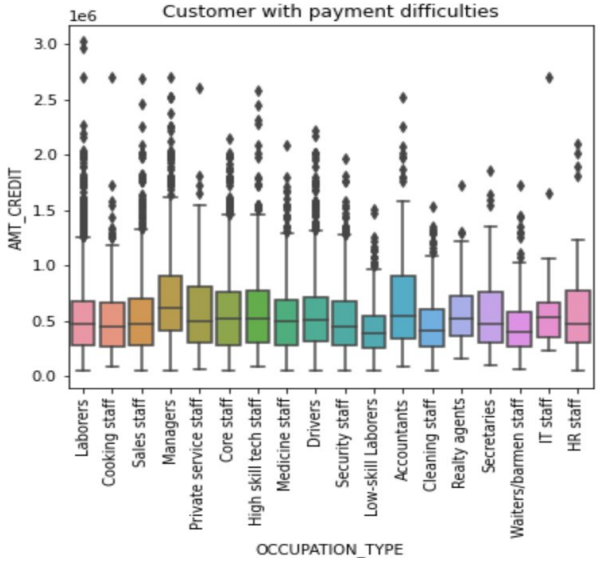
```
# AMT_CREDIT & OCCUPATION_TYPE

plt.figure(figsize=(13,5))

plt.subplot(1,2,1)
ax = sns.boxplot(data=target0,y='AMT_CREDIT',x='OCCUPATION_TYPE')
plt.title('Customer without payment difficulties')
plt.xticks(rotation=90)

plt.subplot(1,2,2)
ax = sns.boxplot(data=target1,y='AMT_CREDIT',x='OCCUPATION_TYPE')
plt.title('Customer with payment difficulties')
plt.xticks(rotation=90)
plt.show()
```





- In the first figure , we have a scatter-plot presenting us with the data to determine whether or not two variables have a relationship or correlation.

  It helps us to determine whether there's a potential relationship between them or not.

- In the second figure we can see what kind of occupation does the client have compared to the range of the clients with and without payment difficulties.

  Boxplot provides a visual summary of the data enabling us to quickly identify mean values, the dispersion of the data set & the median .

Handling Missing Values

```
#Percentage of null values can be founded like this.
ndp = appl_df.isnull().sum()*100/len(appl_df)


#This code will present the percentage of null values more than or equal to 40.
mdc = ndp[ndp>=40]
mdc
```

```
OWN_CAR_AGE                    65.990810
EXT_SOURCE_1                   56.381073
APARTMENTS_AVG                 50.749729
BASEMENTAREA_AVG               58.515956
YEARS_BEGINEXPLUATATION_AVG    48.781019
YEARS_BUILD_AVG                66.497784
COMMONAREA_AVG                 69.872297
ELEVATORS_AVG                  53.295980
ENTRANCES_AVG                  50.348768
FLOORSMAX_AVG                  49.760822
FLOORSMIN_AVG                  67.848630
LANDAREA_AVG                   59.376738
LIVINGAPARTMENTS_AVG           68.354953
```

```
#Lets drop all the unwanted columns

droop = ['FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE','FLAG_PHONE','FLAG_EMAIL','DAYS_LAST_PHONE_

appl_df.drop(columns=droop, inplace=True)
```

- In the above figures ,to handle the missing values we counted the percentage of null values.

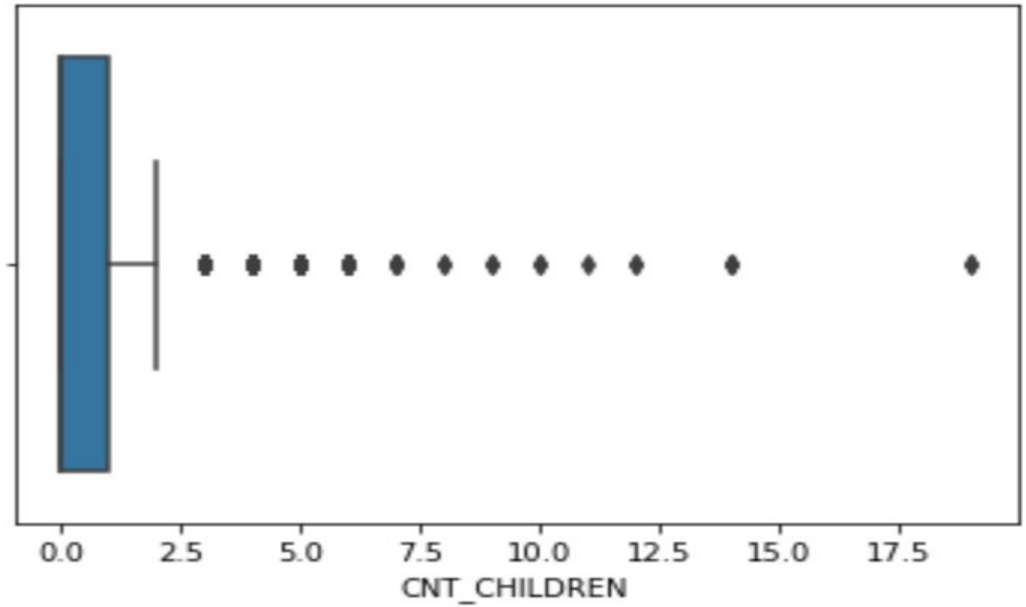
- In our case , we have dropped all the columns having percentage of missing values more than equal to 40 % .

- This helps us to focus on columns which have values to work on.

Outliers

```
#Using the column CNT_CHILDREN to check the outlier.

sns.boxplot(appl_df['CNT_CHILDREN'])
plt.show()
```



Outliers

- Outliers can be considered as missing values.

- The above figure, it is a boxplot of the CNT_CHILDREN .It shows the count of children in a family . we can see the upper extreme ranges till 2 which is the general case in real life.

- But we can see the count of children is as high as 19, which is not possible in general case scenario. Hence Outliers are present.

Top 10 High Correlation (Defaulters).

SK_ID_CURR	1.000000
OBS_30_CNT_SOCIAL_CIRCLE	OBS_60_CNT_SOCIAL_CIRCLE 0.998396
AMT_CREDIT	AMT_GOODS_PRICE 0.986238
REGION_RATING_CLIENT	REGION_RATING_CLIENT_W_CITY 0.949613
CNT_FAM_MEMBERS	CNT_CHILDREN 0.884883
LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION 0.876512
DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE 0.860864
LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY 0.827811
AMT_GOODS_PRICE	AMT_ANNUITY 0.758679
AMT_CREDIT	AMT_ANNUITY 0.752373
DAYS_BIRTH	FLAG_EMP_PHONE 0.542404
dtype: float64	

Top 10 High Correlation (Non-Defaulters).

SK_ID_CURR	SK_ID_CURR	1.000000
OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998690
AMT_CREDIT	AMT_GOODS_PRICE	0.989088
REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.952717
CNT_CHILDREN	CNT_FAM_MEMBERS	0.862515
DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.859640
FLAG_DOCUMENT_6	DAYS_EMPLOYED	0.828106
REG_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION	0.827431
REG_CITY_NOT_WORK_CITY	LIVE_CITY_NOT_WORK_CITY	0.816970
AMT_GOODS_PRICE	AMT_ANNUITY	0.808540
AMT_CREDIT	AMT_ANNUITY	0.803711

dtype: float64

Conclusion

From the analysis we have done, we can conclude that:-

- The applicants whose previous loans were approved are more likely to pay current loan in time, than the applicants whose previous loans were rejected.
- NAME_CONTRACT_STATUS is an important feature.
- 7% of the previously approved loan applicants, defaulted in current loan.
- 90 % of the previously refused loan applicants, were able to pay current loan
- Bank should get as much as clients from housing type ‘With parents’ as they are having least number of unsuccessful payments.