Credit EDA Assignment

This case study aims to identify patterns which indicate if a client has difficulty paying their installments

ABHIJITH CV

26th-30th may 2022

- **Problem statement**
- 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 Analyse the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.
- **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 (too 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.
- **overall approach of the analysis**
- The files are captured, understood, prepared for analysis (cleaned, processed) and analyses via different methods (statistical summaries and plotting). Then some conclusions are drawn based on the results.

Index

1.Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Libraries are imported

2.Reading the Data set

Data Understanding

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.

Data sets are loaded

3. Data cleaning

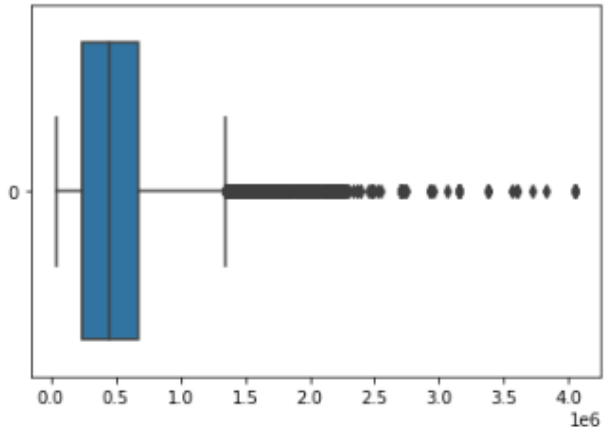
3.1.9 AMT_GOODS_PRICE

Goods price of good that client asked for (if applicable) on the previous application

In [30]:
```
application_data.AMT_GOODS_PRICE.head(20)
```

...

In [31]:
```
# AMT_GOODS_PRICE is a continuous variable.
sns.boxplot(data = application_data.AMT_GOODS_PRICE,orient='h') # Box plot to check the presence of outliers
plt.show()
```



In [32]:
```
print(' AMT_GOODS_PRICE is a numerical feature and has many outliers so Replace missing values with median \n median:',applicatic
```

```
AMT_GOODS_PRICE is a numerical feature and has many outliers so Replace missing values with median
median: 450000.0
```

To fill the missing values with median we can use the below code.

- application_data.AMT_GOODS_PRICE.fillna(application_data.AMT_GOODS_PRICE.median(),inplace = True)

This is an example of an approach took in data cleaning.

- **conversion to a suitable and convenient format**

3.2.2 creating new feature (age)

In [43]:
```
application_data['age'] = abs(application_data.DAYS_BIRTH/365.25)
```

In [44]:
```
application_data.age.describe() # Sanity check
```

Out[44]:
```
count    307511.000000
mean         43.906900
std          11.947950
min          20.503765
25%          33.984942
50%          43.121150
75%          53.886379
max          69.073238
Name: age, dtype: float64
```

Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)

3.2.3 FLAG_OWN_CAR converted to a suitable and convenient format

1 if the client owns a car 0 if not

In [45]:
```
application_data['FLAG_OWN_CAR'] = application_data.FLAG_OWN_CAR.map(lambda x:1 if x=='Y' else 0)
```

3.2.3 FLAG_OWN_REALTY converted to a suitable and convenient format

1 if the client owns a house or flat 0 if not

In [46]:
```
application_data['FLAG_OWN_REALTY'] = application_data.FLAG_OWN_REALTY.map(lambda x:1 if x=='Y' else 0)
```

Different datatype conversions are done for convenient use

3.3 Binning

3.3.1 Binning age

Making an ordinal categorical column with age feature

The column is categorised as follows

- 0-20 very_young
- 20-30 very_adult
- 30-40 adult
- 40-50 elderly
- 50-70 senior_citizen
- 70-200 old_senior_citizen

Features are binned appropriatel

In [47]:
```
buckets = [0,20,30,40,50,70,200]
names = ['very_young','young_adult','adult','elderly','senior_citizen ','old_senior_citizen ']
```

In [48]:
```
application_data['AGE_CATEGORY'] = pd.cut(x=application_data['age'],bins = buckets,labels = names)
```

In [49]:
```
application_data['AGE_CATEGORY']
```

Out[49]:
```
0             young_adult
1                 elderly
2          senior_citizen
3          senior_citizen
4          senior_citizen
                ...
307506        young_adult
307507     senior_citizen
307508            elderly
307509              adult
307510            elderly
Name: AGE_CATEGORY, Length: 307511, dtype: category
Categories (6, object): ['very_young' < 'young_adult' < 'adult' < 'elderly' < 'senior_citizen ' < 'old_senior_citizen ']
```

4.Splitting the data frame

- Helps us to do side by side comparison

In [56]:
```
application_data.shape
```

Out[56]: (307511, 76)

In [57]:
```
df1 = application_data[application_data.TARGET == 1] # All Target variable is 1
df1                                                  #This df represents client with payment difficulties
```

...

In [58]:
```
df0 = application_data[application_data.TARGET == 0] # All Target variable is 0
df0                                                  #This df represents client's without payment difficulties
```

...

In [59]:
```
(24825 + 282686) # sanity check : sum of all the rows of df's add back upto the number of original rows
```

Out[59]: 307511

The data frame is split into two

5. Application_data (Analysis) ¶

5.1 Univariate Analysis

In [60]:
```
application_data.head()
```

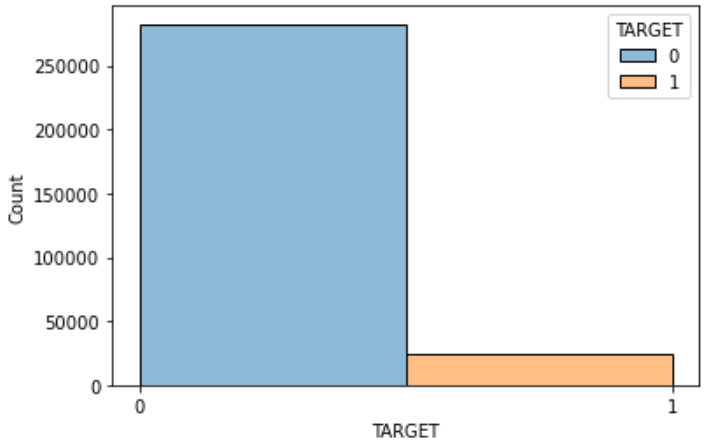
Out[60]:

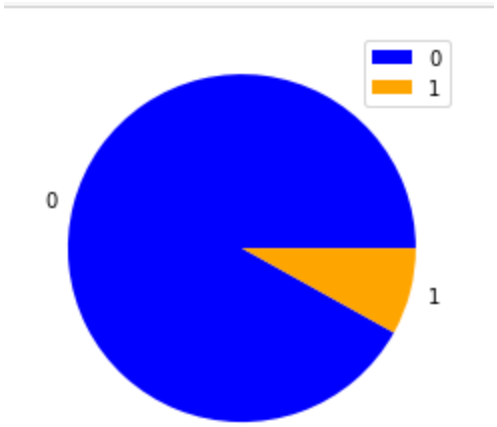
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREI
0	100002	1	Cash loans	M	0	1	0	202500.0	40659
1	100003	0	Cash loans	F	0	0	0	270000.0	129350
2	100004	0	Revolving loans	M	1	1	0	67500.0	13500
3	100006	0	Cash loans	F	0	1	0	135000.0	31268
4	100007	0	Cash loans	M	0	1	0	121500.0	51300

5.1.1 TARGET

Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)

5.1.1.1 Imbalance check

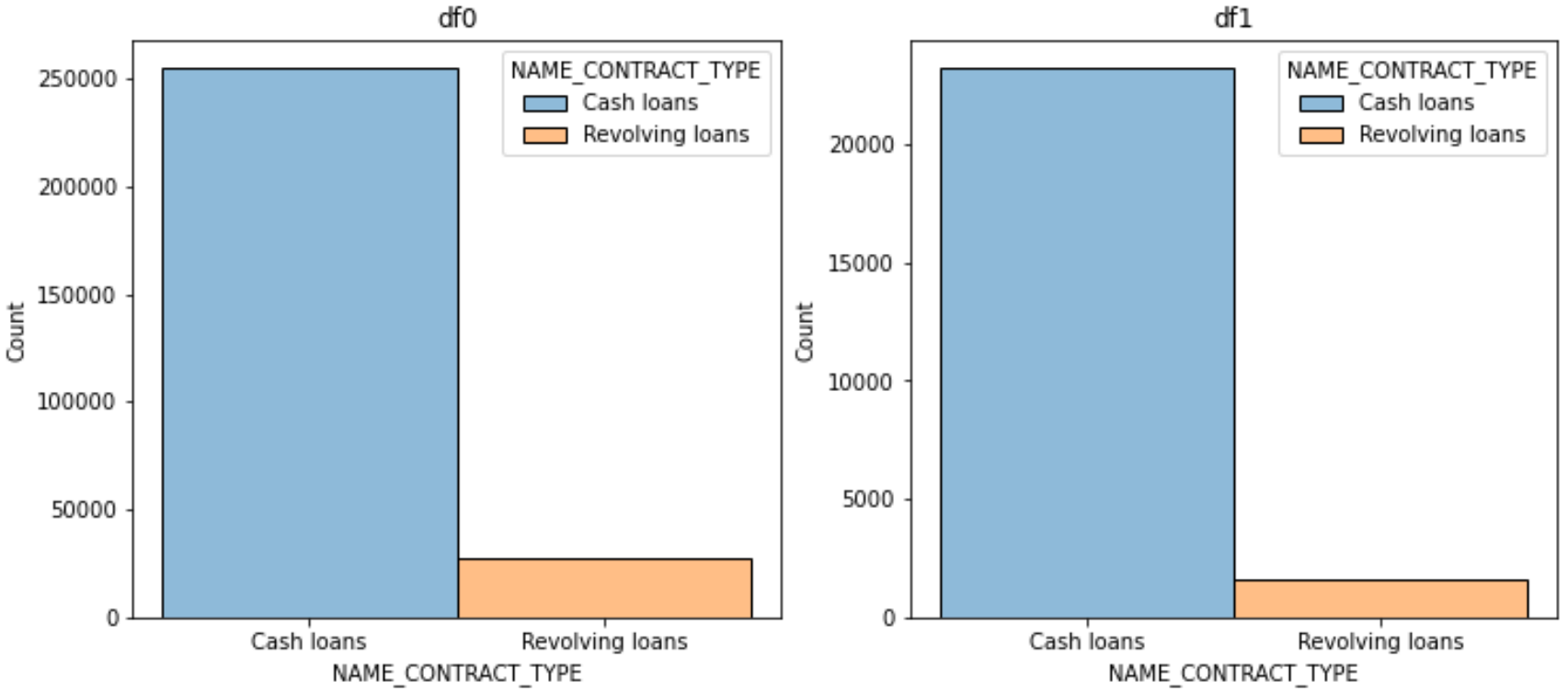




- 92% of people didn't default whereas to 8% defaulted.
- From the above statistics it is clear that the data is imbalanced

5.1.2 NAME_CONTRACT_TYPE

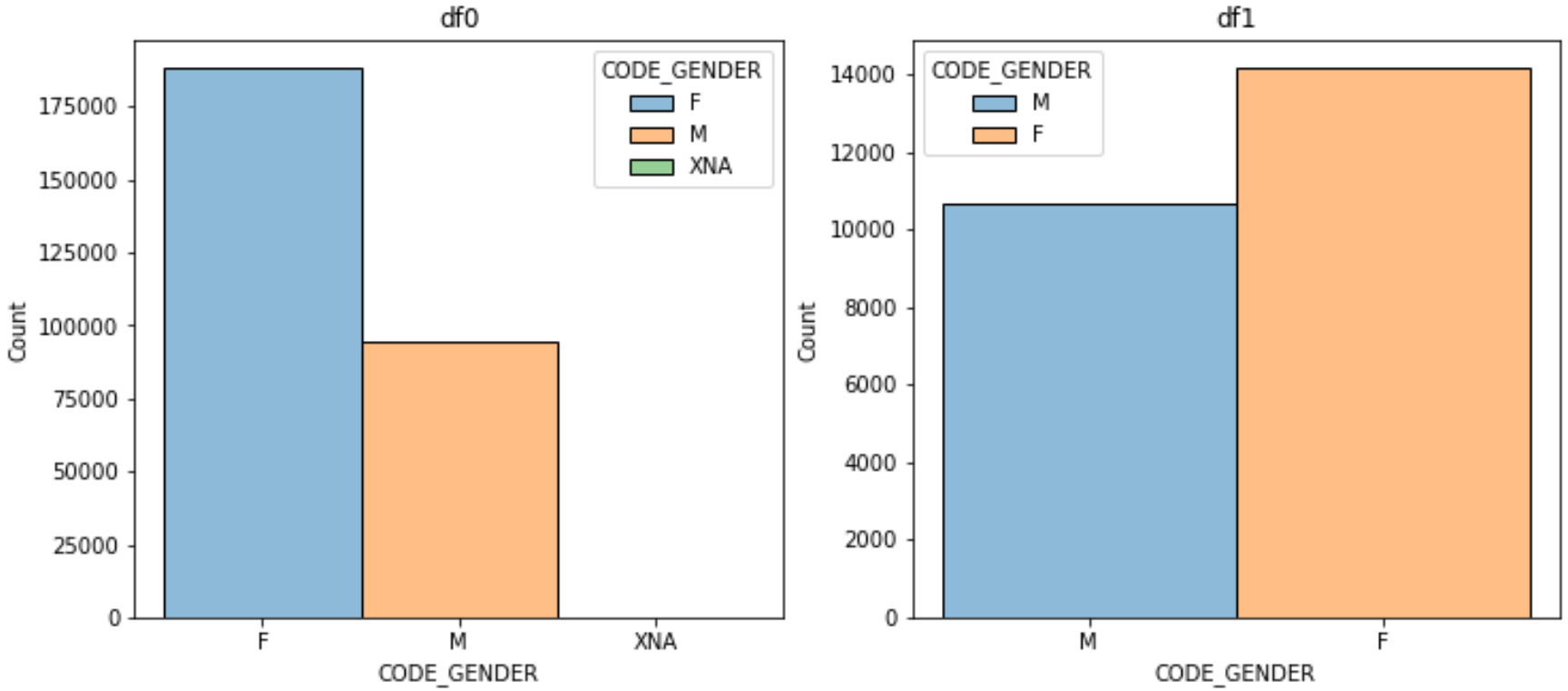
Identification if loan is cash or revolving



- **Non defaulters took 90 % of Cash loans and 10 % Revolving loans whereas defaulters took 93 % of Cash loans and 6 % Revolving loans**
- **We can see that there is a 3% increase in Cash loans and 3 % decrease in Revolving loans taken by the defaulters compared to non-defaulters**

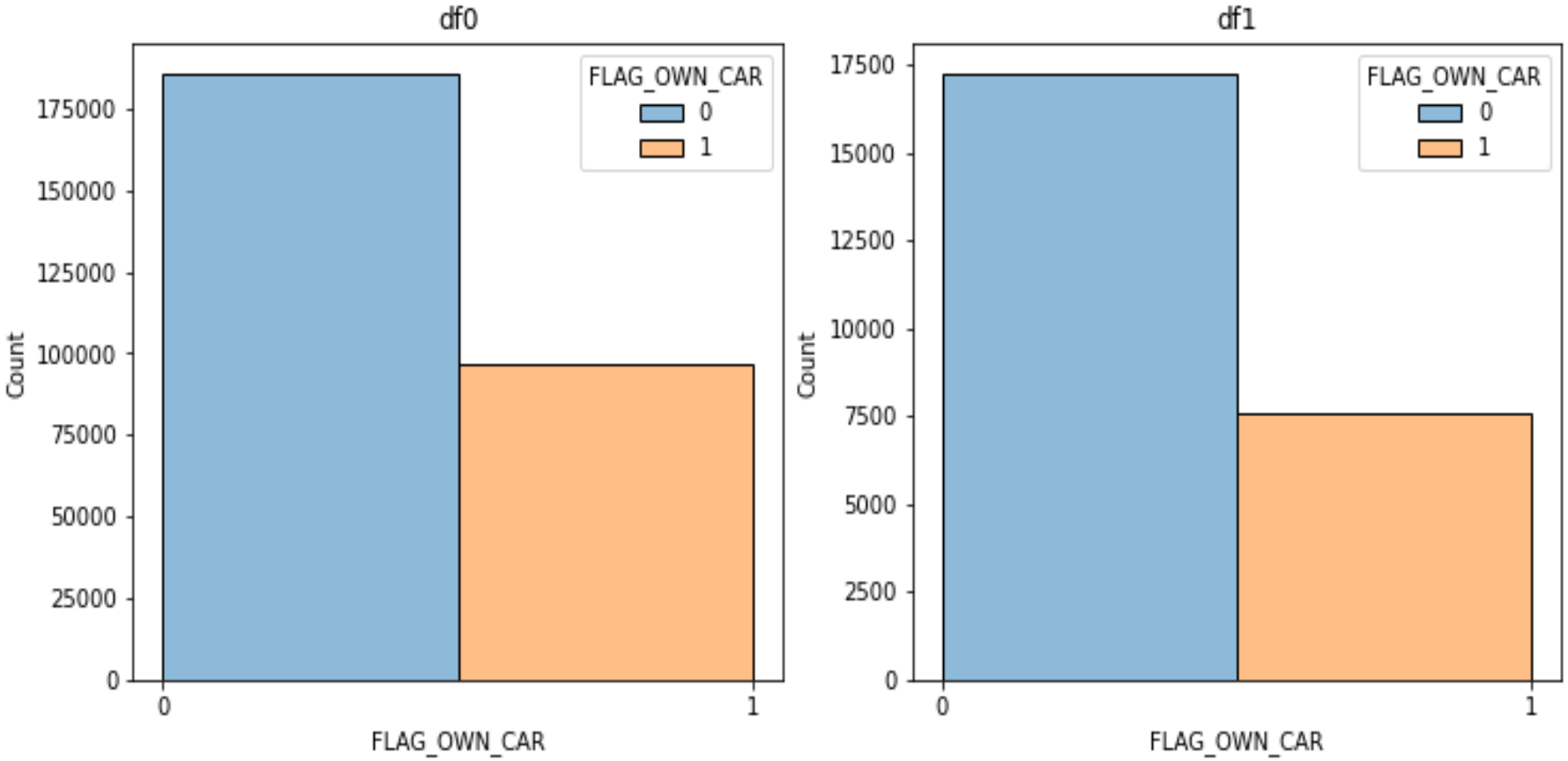
5.2.3 CODE_GENDER

Gender of the client



- **57 % of defaulters are females.**
- **But we have to remember, 65 % of loans were took by females .Thus they have more defaulters.**
- **This increase in defaulters in females is later explained in bi-variant analysis.**

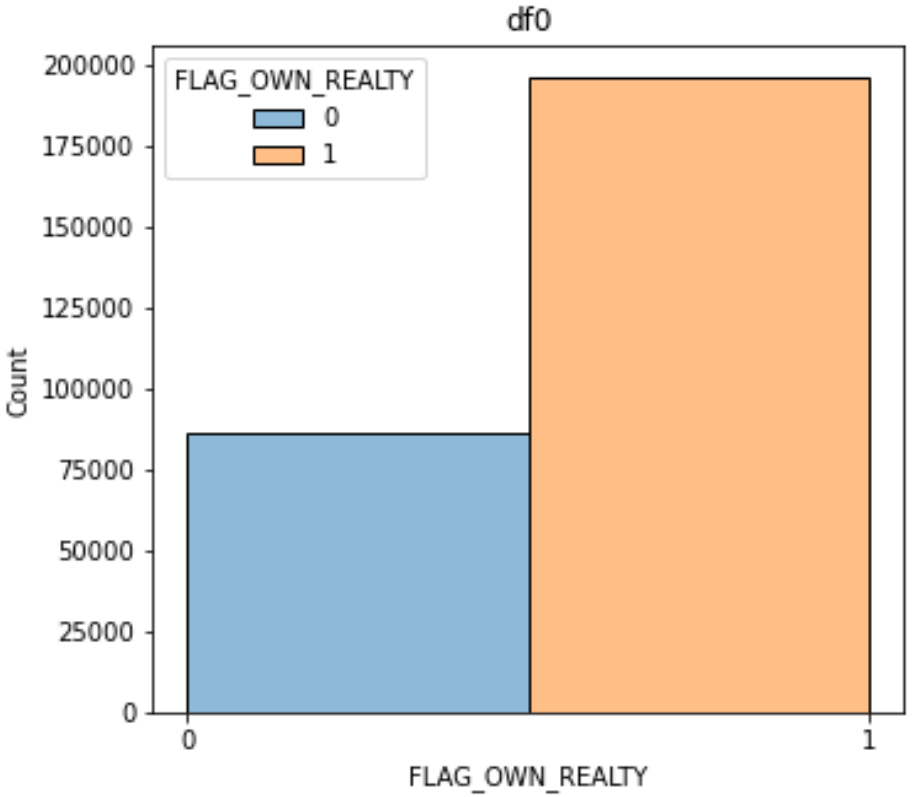
5.2.4 FLAG_OWN_CAR

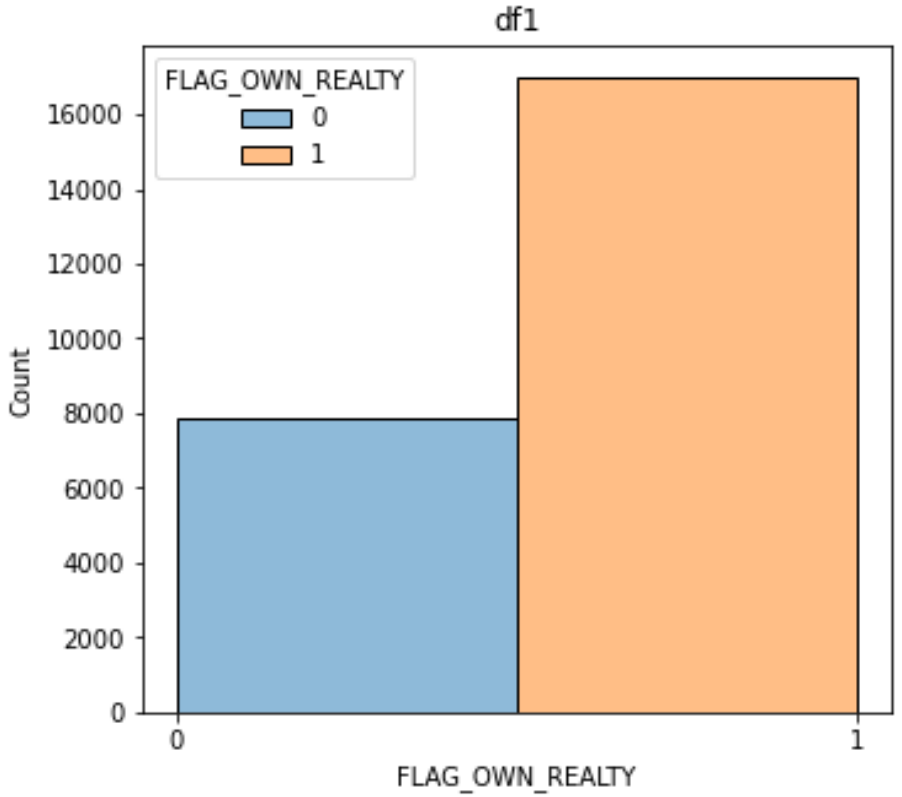


- **34 % of non-defaulters own a car while only 30 % of defaulters own it.**
- **But there are about 2 times more people that does not own a car compared to people that own a car.**

5.2.5 FLAG_OWN_REALTY

Flag if client owns a house or flat

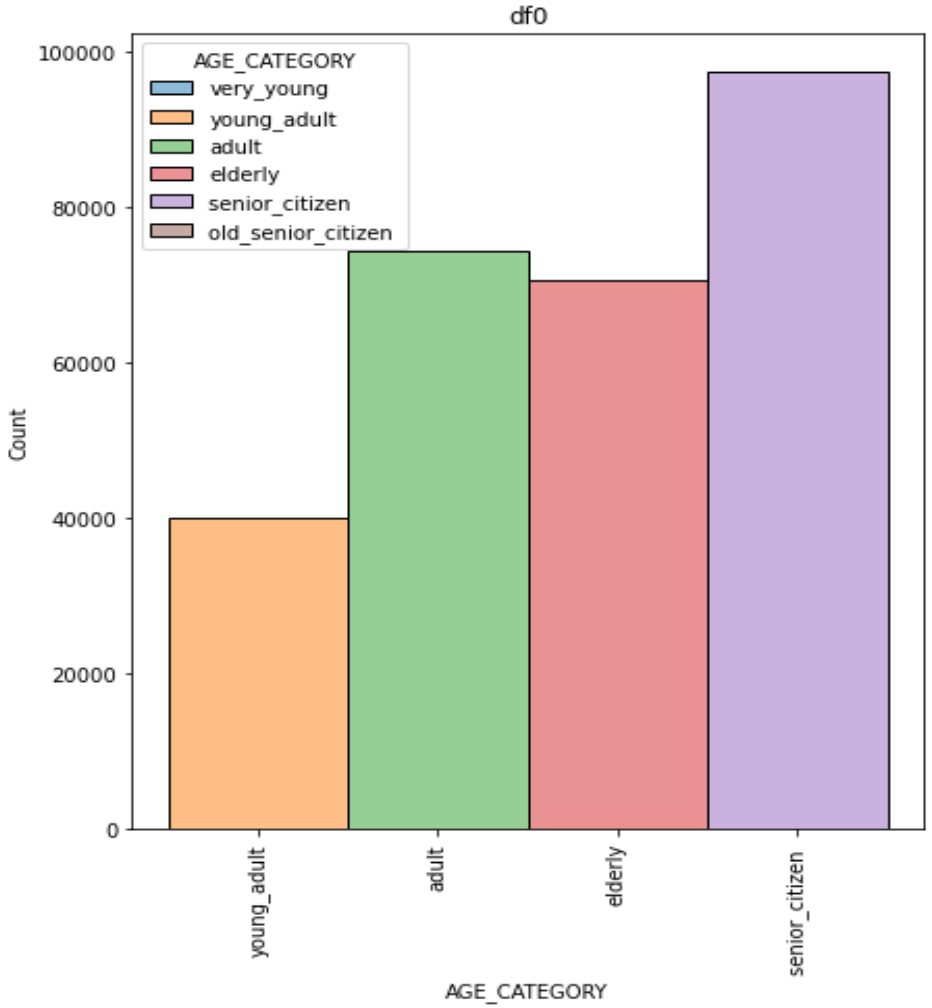


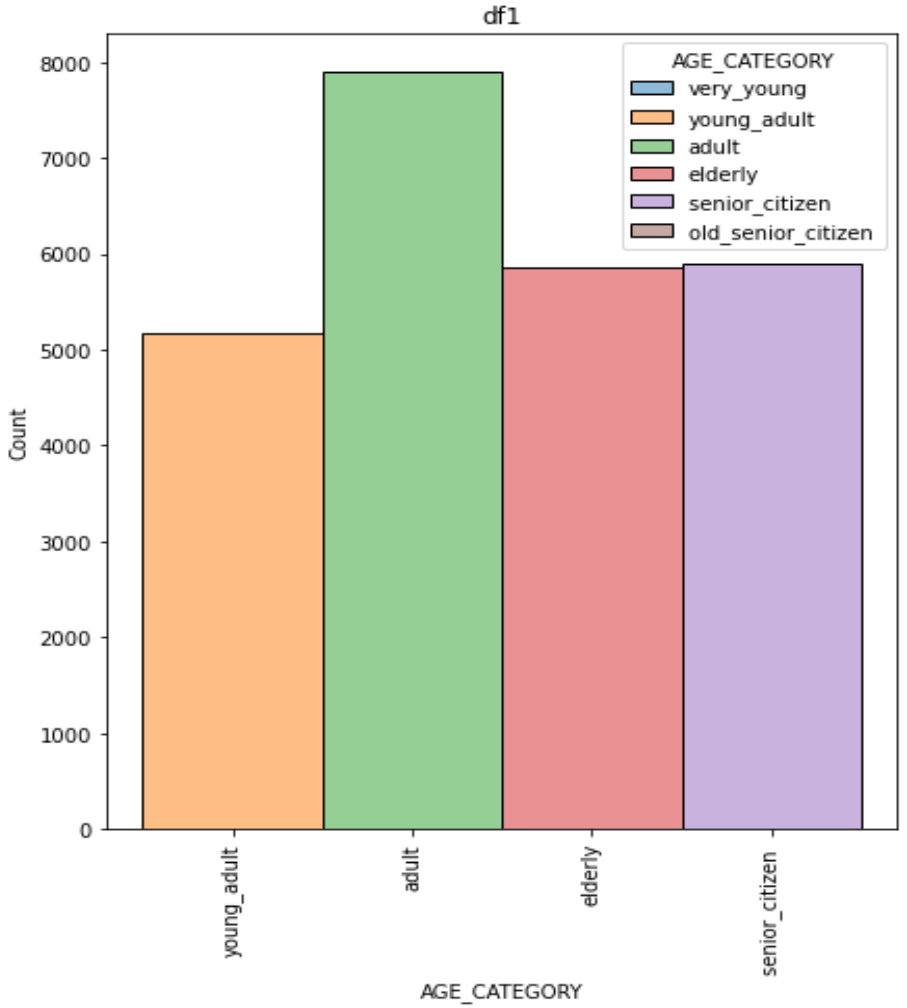


- **distribution of defaulters and non-defaulters are almost same for FLAG_OWN_REALTY feature**

5.2.8 AGE_CATEGORY

Type of organization where client works

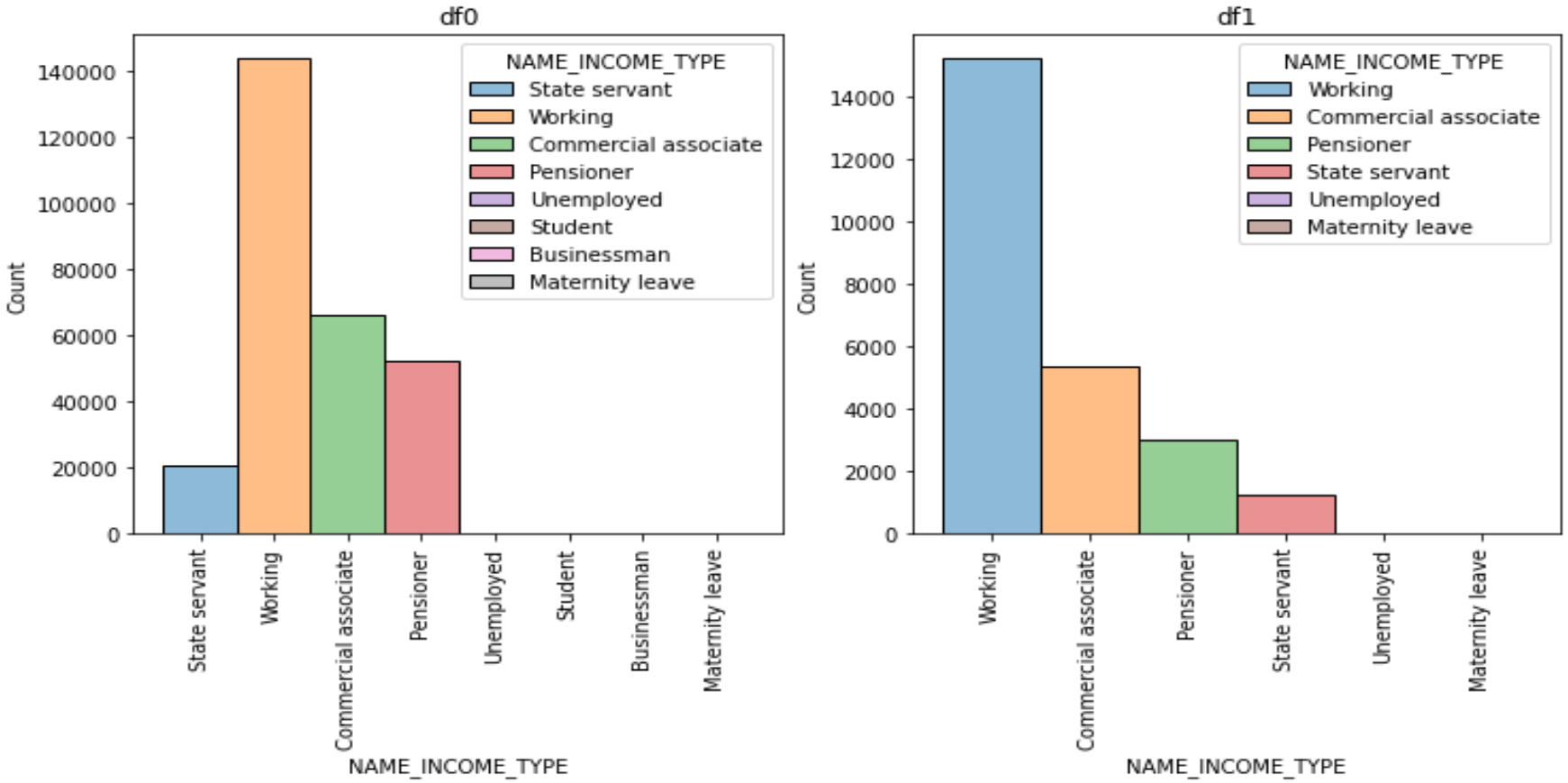




- **Old senior citizen's (70+) do not take loans**
- **for non-defaulter's senior citizens are found to take more loans**
- **For defaulter's adults are found to take more loans**

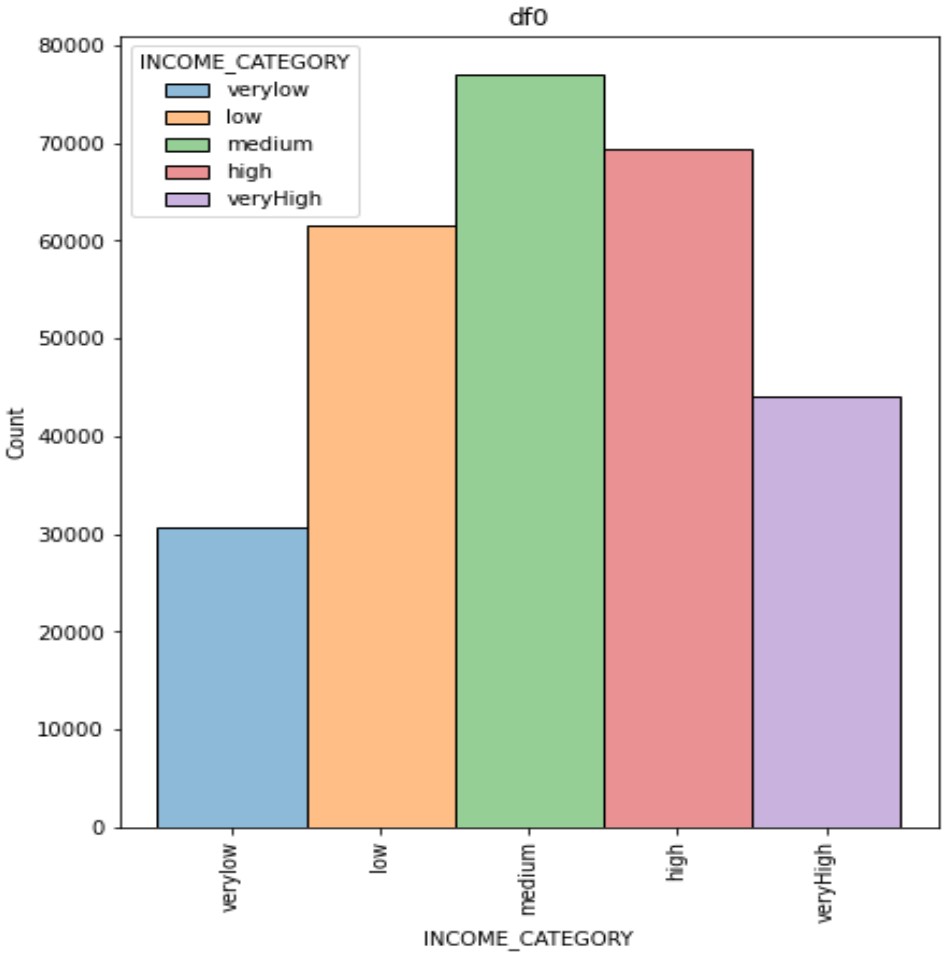
5.2.7 NAME_INCOME_TYPE

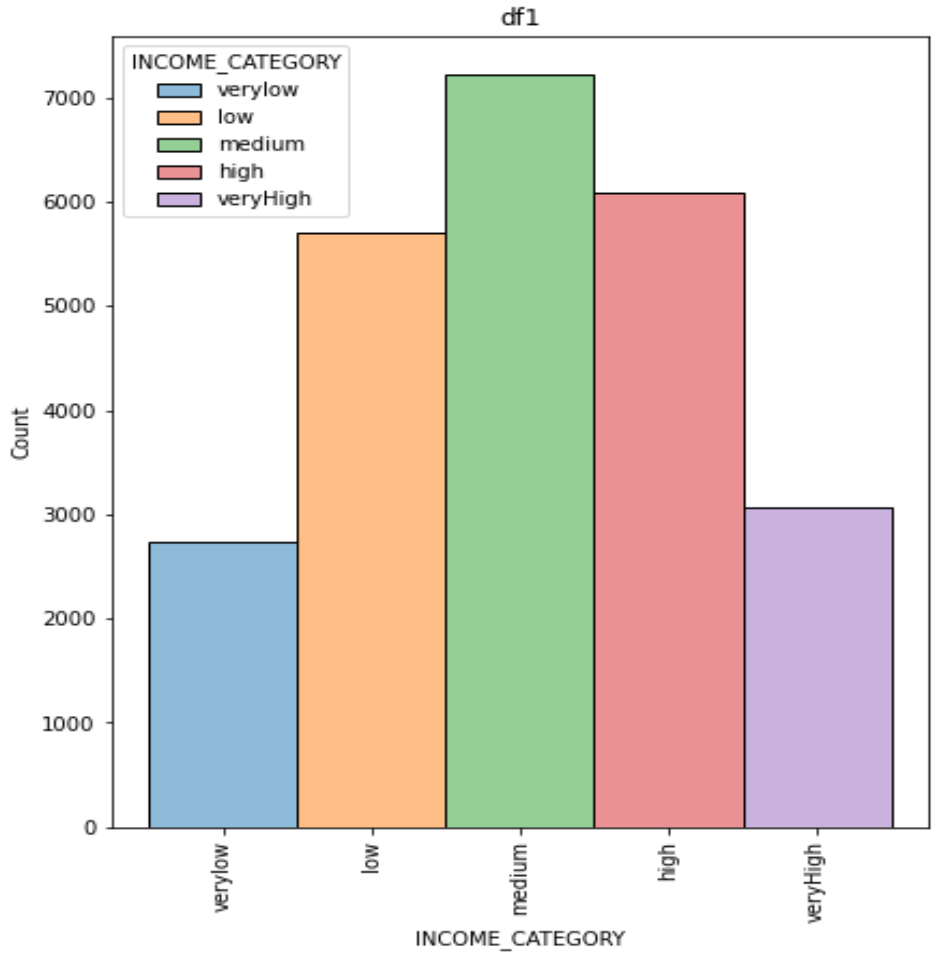
Clients income type (businessman, working, maternity leave,…)



- **Students and Businessman have 100 % non-default rate(But very few of them took loans).**
- **51 % of loans are given to Working people.**
- **23 % of loans are given to Commercial associate.**
- **18 % of loans are given to Pensioner.**

5.2.9 INCOME_CATEGORY

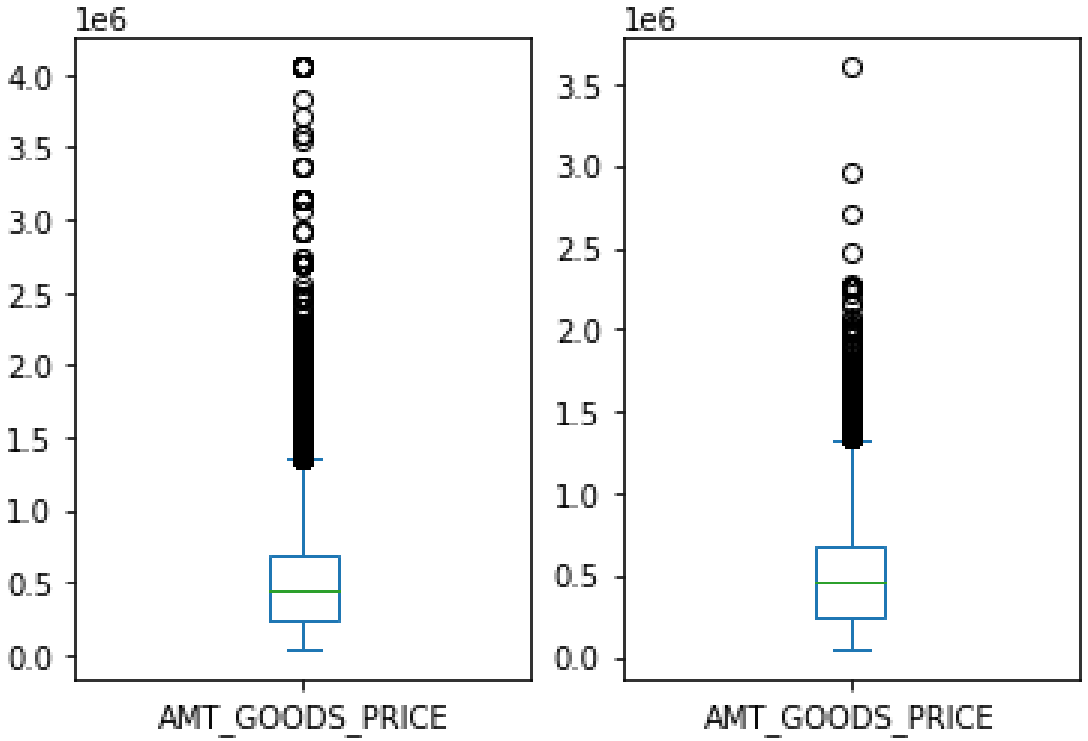




- **The salary distribution of defaulter's and defaulter's are very close**
- **This means we cannot directly predict if a person will be defaulter or not, just by looking at their salary**

5.2.10 AMT_GOODS_PRICE (continuous numerical)¶

For consumer loans it is the price of the goods for which the loan is given



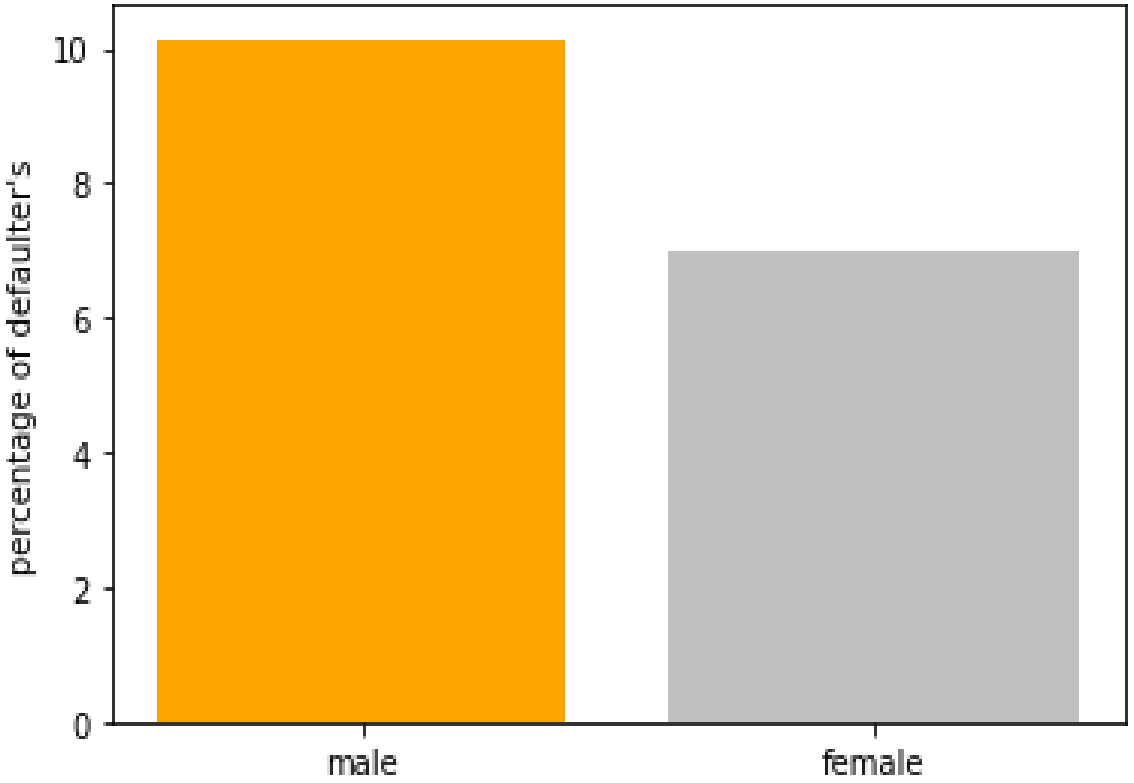
Many outliers are present in AMT_GOODS_PRICE feature for both the data frames

5.2 Bivariate analysis

5.2.1 CODE_GENDER vs TARGET

increase in defaulters in females(explanation).

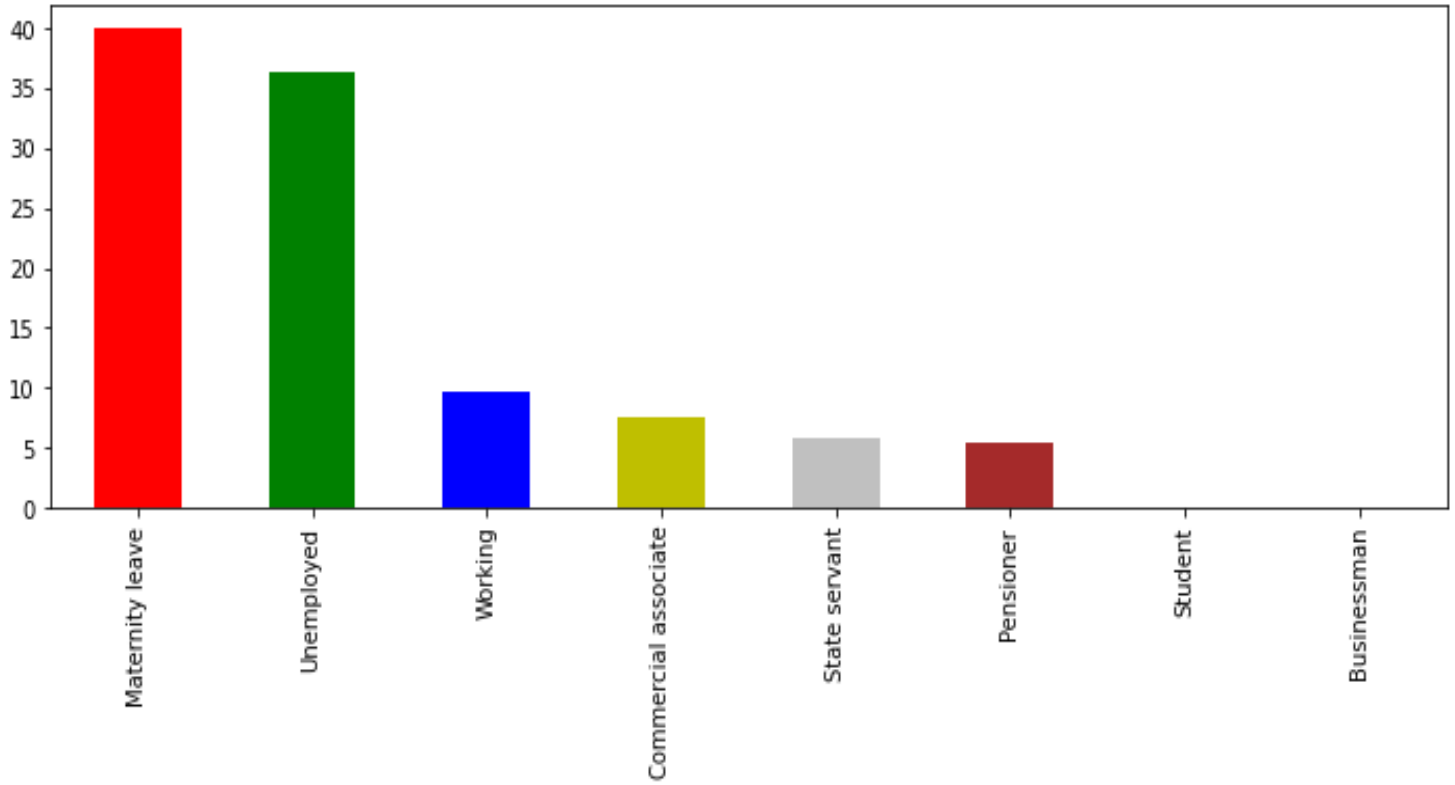
57 % of defaulters are females. But we have to remember, 65 % of loans were took by females.



•From the chart it is clear that the percentage of male defaulters are high compared to females.

5.2.3 NAME_INCOME_TYPE vs TARGET

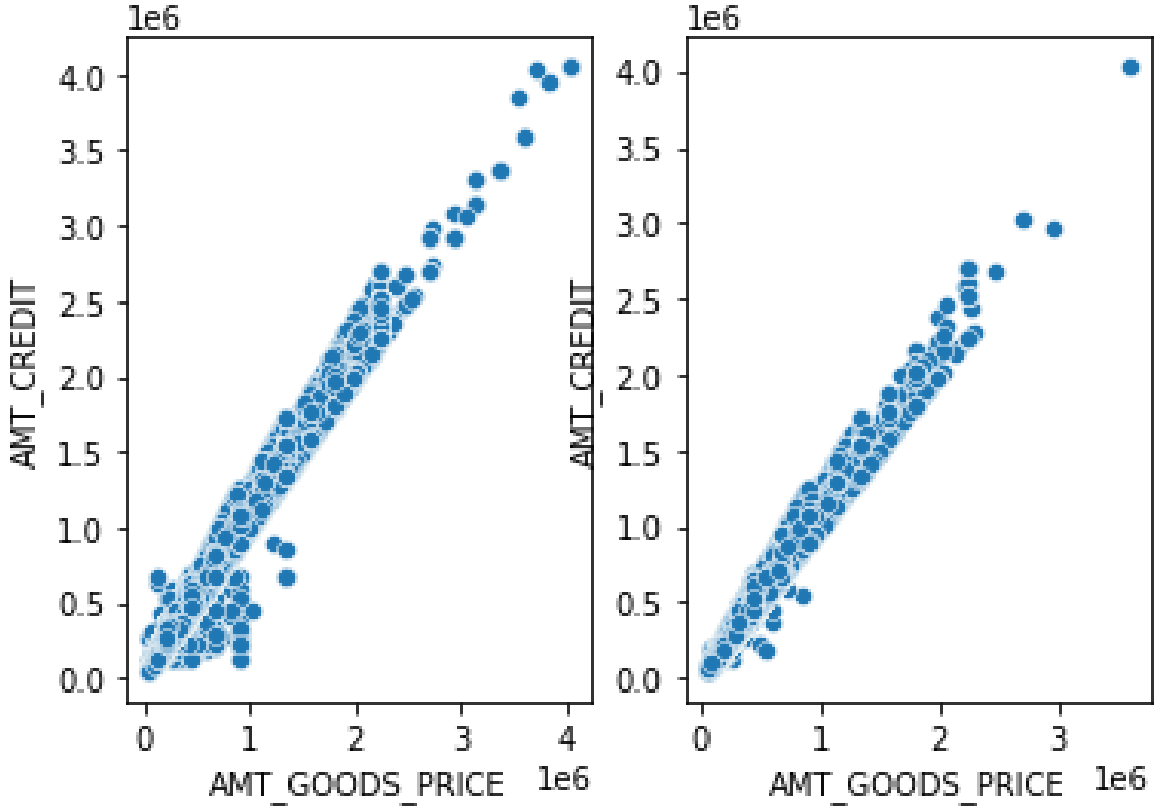
Client income type (businessman ,working ,maternity leave.)



- **Unemployed and Maternity leave loan takes have the highest percentage of defaulters in their group respectively.**
- **More loans can be given to students and Businessman**

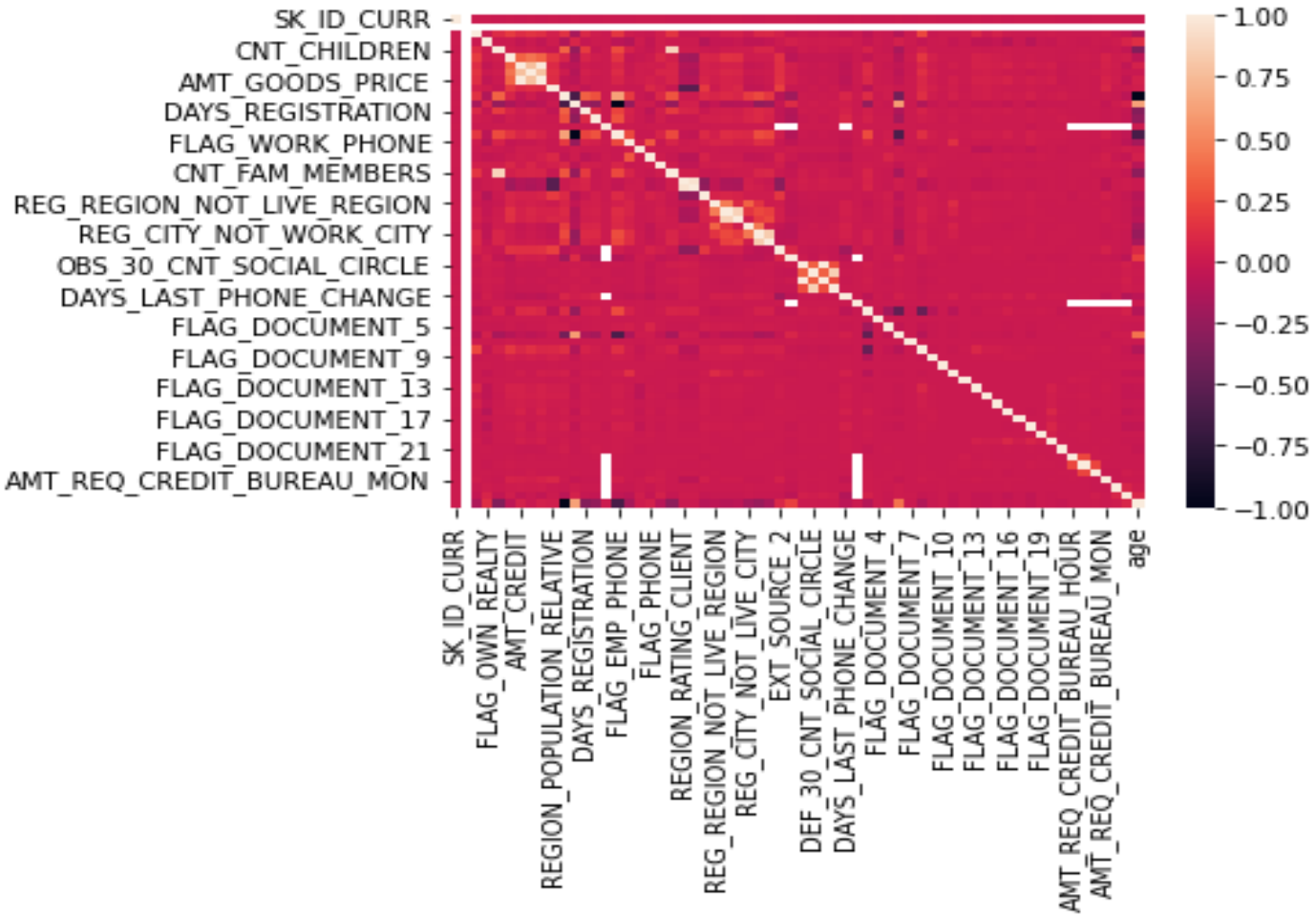
5.2.4 AMT_GOODS_PRICE vs AMT_CREDIT

AMT GOODS PRICE: For consumer loans it is the price of the goods for which the loan is given AMT CREDIT : Credit amount of the loan



- **Shows strong liner relation**

5.3 Correlation



5.3.1 Correlation for defaulter's

	Column1	Column2	Correlation
0	age	DAYS_BIRTH	-1.000000
1	FLAG_EMP_PHONE	DAYS_EMPLOYED	-0.999702
2	FLAG_EMP_PHONE	FLAG_DOCUMENT_6	-0.617421
3	FLAG_EMP_PHONE	age	-0.578519
4	DAYS_BIRTH	DAYS_EMPLOYED	-0.575097
5	FLAG_DOCUMENT_8	FLAG_DOCUMENT_3	-0.528927
6	FLAG_DOCUMENT_6	FLAG_DOCUMENT_3	-0.475807
7	REGION_POPULATION_RELATIVE	REGION_RATING_CLIENT_W_CITY	-0.446977
8	REGION_POPULATION_RELATIVE	REGION_RATING_CLIENT	-0.443236
9	DAYS_BIRTH	FLAG_DOCUMENT_6	-0.387422

•Top 10 Negative correlation

	Column1	Column2	Correlation
0	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998269
1	AMT_CREDIT	AMT_GOODS_PRICE	0.983103
2	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.956637
3	CNT_FAM_MEMBERS	CNT_CHILDREN	0.885484
4	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.868994
5	LIVE_REGION_NOT_WORK_REGION	REG_REGION_NOT_WORK_REGION	0.847885
6	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.778540
7	AMT_GOODS_PRICE	AMT_ANNUITY	0.752699
8	AMT_ANNUITY	AMT_CREDIT	0.752195
9	DAYS_EMPLOYED	FLAG_DOCUMENT_6	0.617307

•Top 10 Positive correlation

5.3.2 Correlation for non-defaulter's

	Column1	Column2	Correlation
0	age	DAYS_BIRTH	-1.000000
1	FLAG_EMP_PHONE	DAYS_EMPLOYED	-0.999758
2	FLAG_EMP_PHONE	age	-0.622073
3	DAYS_EMPLOYED	DAYS_BIRTH	-0.618048
4	FLAG_DOCUMENT_6	FLAG_EMP_PHONE	-0.596060
5	REGION_POPULATION_RELATIVE	REGION_RATING_CLIENT	-0.539005
6	REGION_RATING_CLIENT_W_CITY	REGION_POPULATION_RELATIVE	-0.537301
7	FLAG_DOCUMENT_3	FLAG_DOCUMENT_6	-0.486422
8	FLAG_DOCUMENT_3	FLAG_DOCUMENT_8	-0.461071
9	FLAG_DOCUMENT_6	DAYS_BIRTH	-0.407936

•Top 10 Negative correlation

	Column1	Column2	Correlation
0	OBS_60_CNT_SOCIAL_CIRCLE	OBS_30_CNT_SOCIAL_CIRCLE	0.998508
1	AMT_GOODS_PRICE	AMT_CREDIT	0.987250
2	REGION_RATING_CLIENT_W_CITY	REGION_RATING_CLIENT	0.950149
3	CNT_CHILDREN	CNT_FAM_MEMBERS	0.878571
4	REG_REGION_NOT_WORK_REGION	LIVE_REGION_NOT_WORK_REGION	0.861861
5	DEF_60_CNT_SOCIAL_CIRCLE	DEF_30_CNT_SOCIAL_CIRCLE	0.859332
6	LIVE_CITY_NOT_WORK_CITY	REG_CITY_NOT_WORK_CITY	0.830381
7	AMT_ANNUITY	AMT_GOODS_PRICE	0.776686
8	AMT_CREDIT	AMT_ANNUITY	0.771309
9	FLAG_EMP_PHONE	DAYS_BIRTH	0.622073

•Top 10 positive correlation

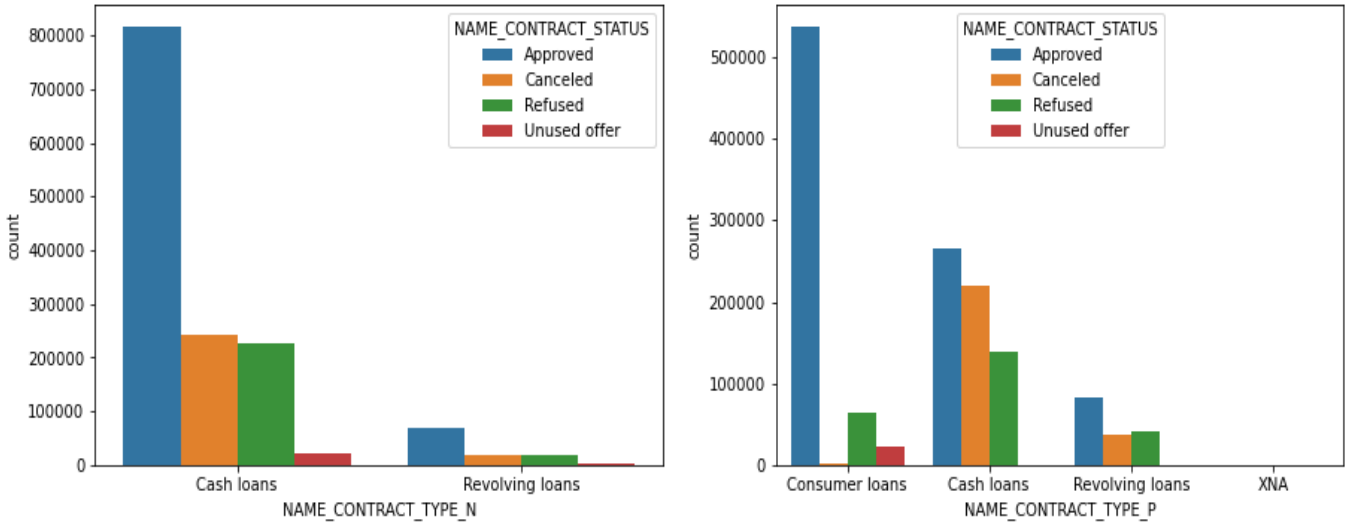
5.4 Merging new and old application

```
merged_data=pd.merge(application_data,previous_application,how='left',on='SK_ID_CURR',suffixes=('_N', '_P'))
merged_data.head()
```

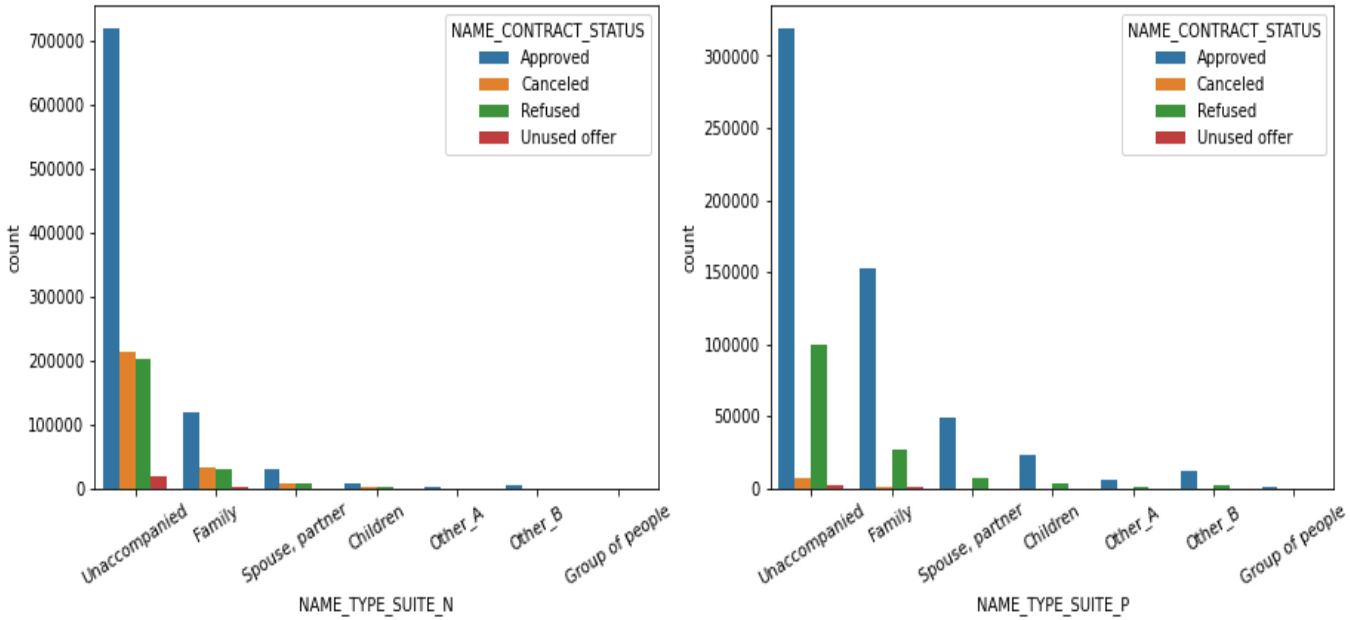
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_N	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
0	100002	1	Cash loans	M	0	1	0	202500.0
1	100003	0	Cash loans	F	0	0	0	270000.0
2	100003	0	Cash loans	F	0	0	0	270000.0
3	100003	0	Cash loans	F	0	0	0	270000.0
4	100004	0	Revolving loans	M	1	1	0	67500.0

Combining two data frames to do further analysis

'NAME_CONTRACT_TYPE_N **V/S** NAME_CONTRACT_STATUS'



NAME_CONTRACT_TYPE_P **V/S** NAME_CONTRACT_STATUS



- **For new application percentage of cash loans are high**
- **For previous applications percentage of consumer loans are high**
- **Both previous application and new application personals where unaccompanied at the time of loan application**
- **When with family previous application loans were very unlikely to get canceled.**

6. Conclusion

Imbalance.

- 92% of people didn't default whereas to 8% defaulted.
- From the above statistics it is clear that the data is imbalanced
- We can't predict if a person is a defaulter just by looking at their **salary.**The salary distribution of defaulter's and defaulter's are very close
- This means we cannot directly predict if a person will be defaulter or not, just by looking at their salary

Gender *discrepancy findings.*

- 57 % of defaulters are females.
- But we have to remember, 65 % of loans were took by females. Thus they have more defaulters.
- Also, from our analysis it is clear that percentage of male defaulters are high compared to females.

Does *owning a car make a difference ?*

- 34 % of non-defaulters own a car while only 30 % of defaulters own it.
- But there are about 2 times more people that does not own a car compared to people that own a car.
- Thus owning a car makes Not much difference

Give more loans to.

- Students and Businessman have 100 % nondefault rate(But very few of them took loans).

Loan analysis.

- Unemployed and Maternity leave loan takes have the highest percentage of defaulters in their group respectively.
- Old senior citizen's (70+) do not take loans
- for non-defaulter's senior citizens are found to take more loans
- for defaulter's adults are found to take more loans Features that can contribute in predicting target feature

Features that predict TARGET.

- NAME_EDUCATION_TYPE,AMT_INCOME_TOTAL,DAYS_BIRTH,AMT_CREDIT,DAYS_EMPLOYED,AMT_ANNUITY

7. References

upGrad **Python for Data Science/module-3**

Data Visualization in Python/module-5

upGrad **Exploratory Data Analysis/module-6**

Stack overflow

GeeksforGeeks

Kaggle data visualization course