CREDIT EDA CASE STUDY

BY

SELVI DURAISAMY

Steps performed for Case Study

- Study the data (Both the application and previous application data)
- Analyze each feature , data distribution , outliers any
- Data cleaning activity –
  - check the Null % and exclude more than 40 % null column values
  - For the remaining columns impute the missing values by the standard approach learnt
  - Check and format the data to appropriate datatypes , absolute values
- Analysis –
  - Univariate
    - Categorical feature
    - Numerical feature
  - Bivariate
    - Categorical vs Numerical feature
    - Numerical va Numerical
  - Co-orelation between numerical features
  - Segmented univariate/bivariate analysis
- Conclusion

DATA UNDERSTANDING

```
# get the count of no of rows / columns in dataframe
df.shape
```

589] ✓ 0.0s Pytho

```
(307511, 122)
```

```
#identify the null count of columns
null_perc=df.isnull().mean() * 100
missing_value_df = pd.DataFrame({'column_name': df.columns, 'No of Nulls':df.isnull().sum(),'percent_missing': null_perc})
```

590] ✓ 0.3s Pytho

```
missing_value_df.sort_values('percent_missing', ascending=False, inplace=True)
missing_value_df.head()
```

591] ✓ 0.0s Pytho

	column_name	No of Nulls	percent_missing
COMMONAREA_AVG	COMMONAREA_AVG	214865	69.872297
COMMONAREA_MODE	COMMONAREA_MODE	214865	69.872297
COMMONAREA_MEDI	COMMONAREA_MEDI	214865	69.872297
NONLIVINGAPARTMENTS_MEDI	NONLIVINGAPARTMENTS_MEDI	213514	69.432963
NONLIVINGAPARTMENTS_MODE	NONLIVINGAPARTMENTS_MODE	213514	69.432963

DATA UNDERSTANDING

```
df.describe()
```

[597] ✓ 1.8s Pytho

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e+05	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e+05	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e+05	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e+04	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e+05	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e+05	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	

We see many negative values in Days_Birth,Days_Employed,Days_Registration,Days_ID_Publish,Days_Last_Phone_Change

DATA UNDERSTANDING

```python
#There are outliers in AMT_INCOME_TOTAL
```

✓ 0.0s

```python
df.groupby(["NAME_CONTRACT_TYPE","CODE_GENDER"]).count()
```

✓ 0.3s

		SK_ID_CURR	TARGET	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_
NAME_CONTRACT_TYPE	CODE_GENDER						
Cash loans	F	182800	182800	182800	182800	182800	
	M	95432	95432	95432	95432	95432	
Revolving loans	F	19648	19648	19648	19648	19648	
	M	9627	9627	9627	9627	9627	
	XNA	4	4	4	4	4	

```python
# Gender has outlier value "XNA"
```

✓ 0.0s

DATA CLEANING

```
#exclude the columns with missing value more than 40 %
df_filtered=df.loc[:, df.isnull().mean() * 100 <= 40]
```

[604] ✓ 0.2s

```
df_filtered.shape
```

[605] ✓ 0.0s

```
(307511, 73)
```

```
#Check for duplicates and remove them
df_filtered.drop_duplicates()
```

[606] ✓ 0.6s

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHI
0	100002	1	Cash loans	M	N	Y	
1	100003	0	Cash loans	F	N	N	

```
# for family members most of the families have 2 kids so lets go with mode
df_filtered.CNT_FAM_MEMBERS.value_counts()
```

✓ 0.0s

DATA CLEANING

```
# To correct the outlier in CODE_GENDER, we will use mode to correct the outliers
df_filtered[df_filtered['CODE_GENDER']=='XNA'].shape
# There are 4 records with XNA value

df_filtered.loc[df_filtered['CODE_GENDER']=='XNA','CODE_GENDER']='F'
df_filtered['CODE_GENDER'].value_counts()
```

[508] ✓ 0.0s

```
CODE_GENDER
F    202452
M    105059
Name: count, dtype: int64
```

```
# since we have various organization type distributed across, will not be able to impute any value for this feature.
# So can exclude this "XNA" orgnization type

df_filtered=df_filtered.drop(df_filtered.loc[df_filtered['ORGANIZATION_TYPE']=='XNA'].index)
df_filtered[df_filtered['ORGANIZATION_TYPE']=='XNA'].shape
```

[509] ✓ 0.1s

```
#correct the negative columns

cols=['DAYS_BIRTH','DAYS_EMPLOYED','DAYS_REGISTRATION','DAYS_ID_PUBLISH','DAYS_LAST_PHONE_CHANGE']
df_filtered[cols]=df_filtered[cols].abs()
```

✓ 0.0s

DATA CLEANING

```
# there are different categories of occupation type , so will not be able to use mean or median here ,
# so lets impute by naming it as seperate category
df_filtered.OCCUPATION_TYPE.fillna('UnKnown',inplace=True)
```

✓ 0.0s

```
# external source is the value for the customer from outside which cannot be calculated or so
# so lets consider  mean for the missing value
df_filtered.EXT_SOURCE_3.value_counts()
```

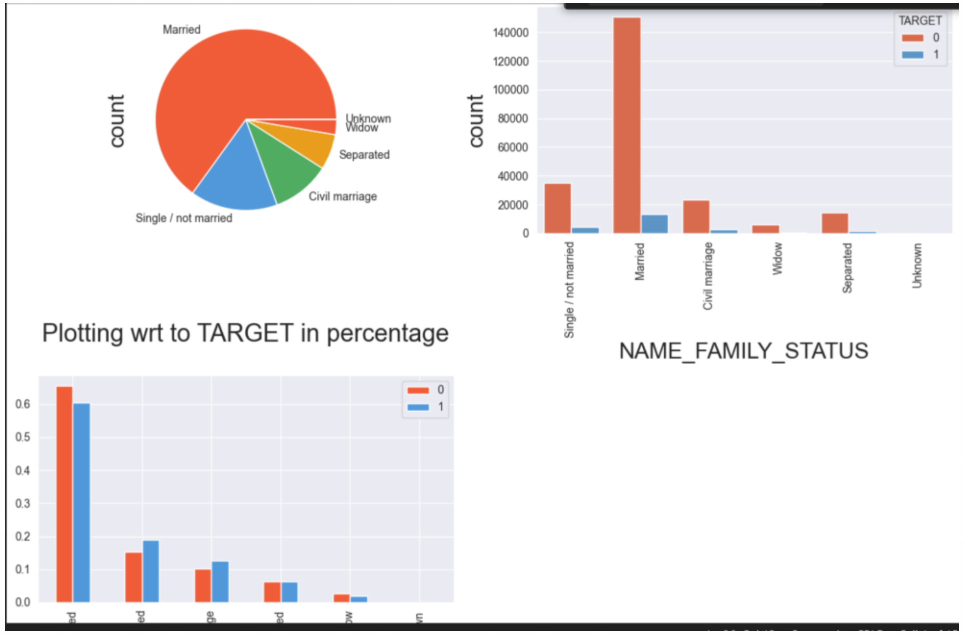
✓ 0.0s

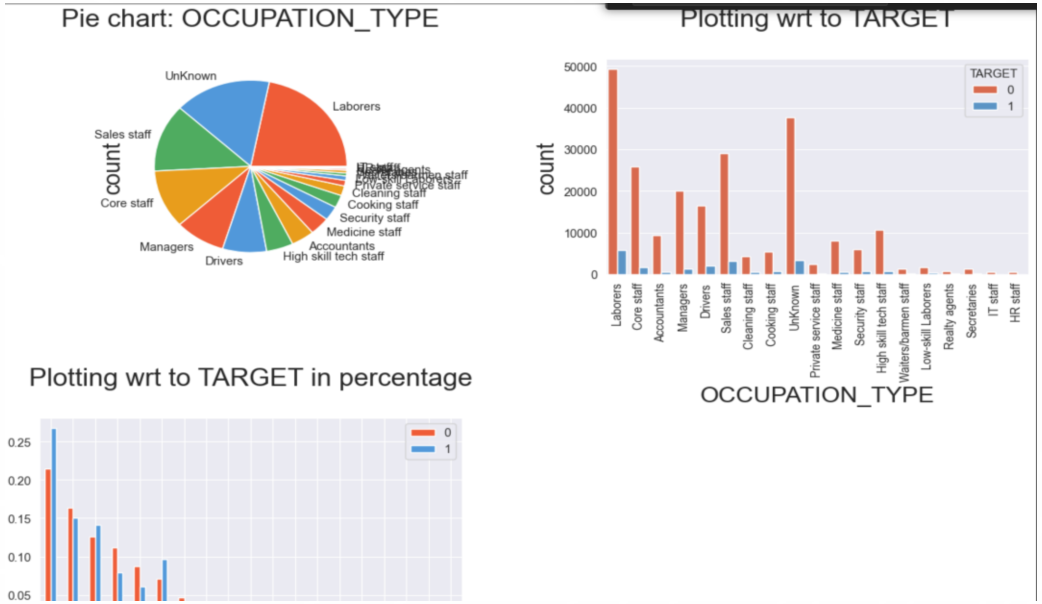
```
# most of them have not enquried and only very countable entries have made enquries, so lets use mode
for col in ['AMT_REQ_CREDIT_BUREAU_HOUR','AMT_REQ_CREDIT_BUREAU_DAY','AMT_REQ_CREDIT_BUREAU_WEEK','AMT_REQ_CREDI
    df_filtered[col].fillna(df_filtered[col].mode()[0],inplace=True)
```

✓ 0.0s

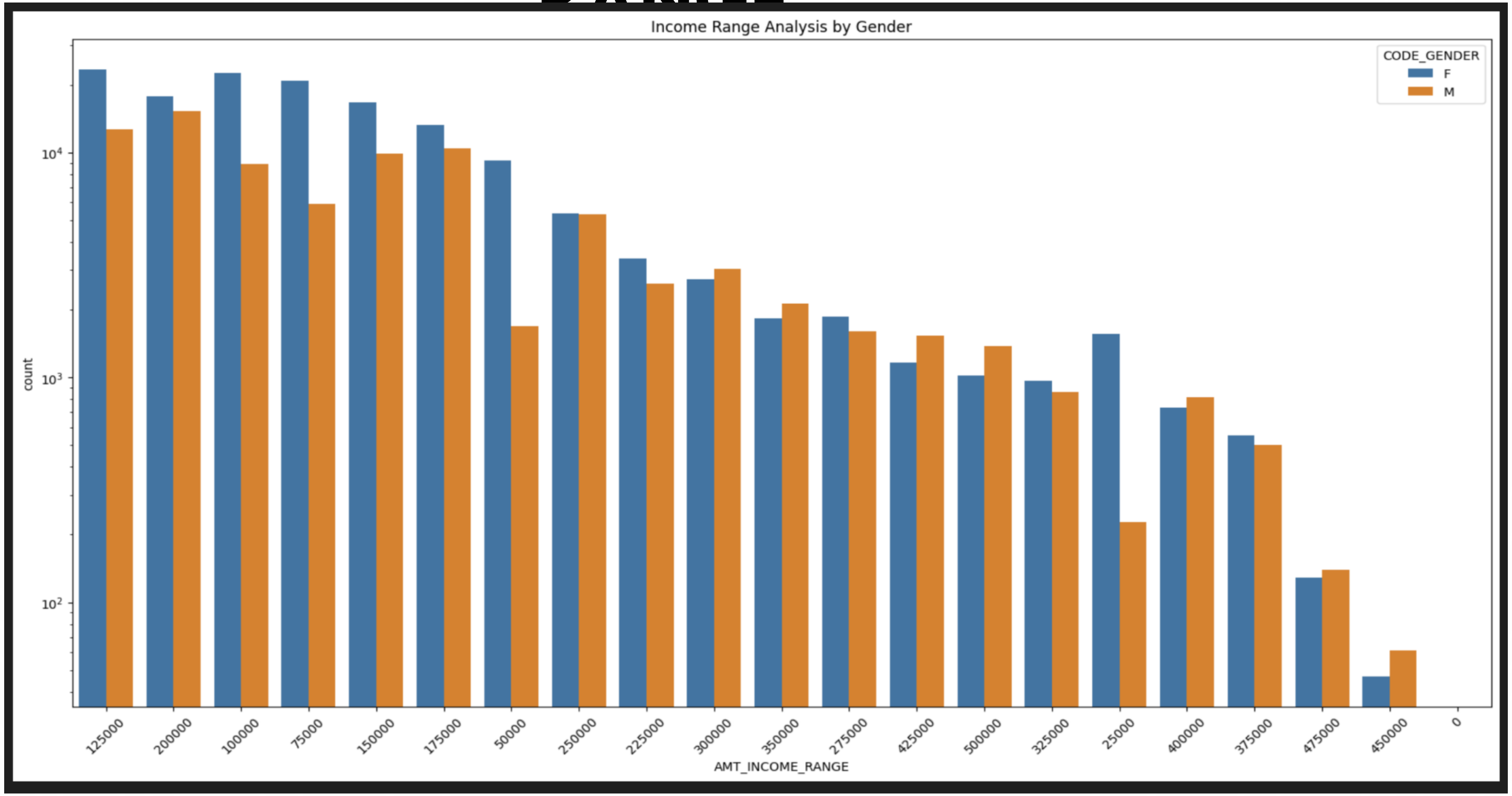
```
# it is the asset or price of the good for which the loan is sanctioned , so most of the time AMT_credit is equal to Amt_Good
# the same is confirmed by mode as well
df_filtered.AMT_GOODS_PRICE.fillna(df_filtered.AMT_CREDIT,inplace=True)
df_filtered.eval('Credit_to_Goods_Ratio = AMT_CREDIT / AMT_GOODS_PRICE', inplace=True)
```

DISTRIBUTION OF OCCUPATION TYPE



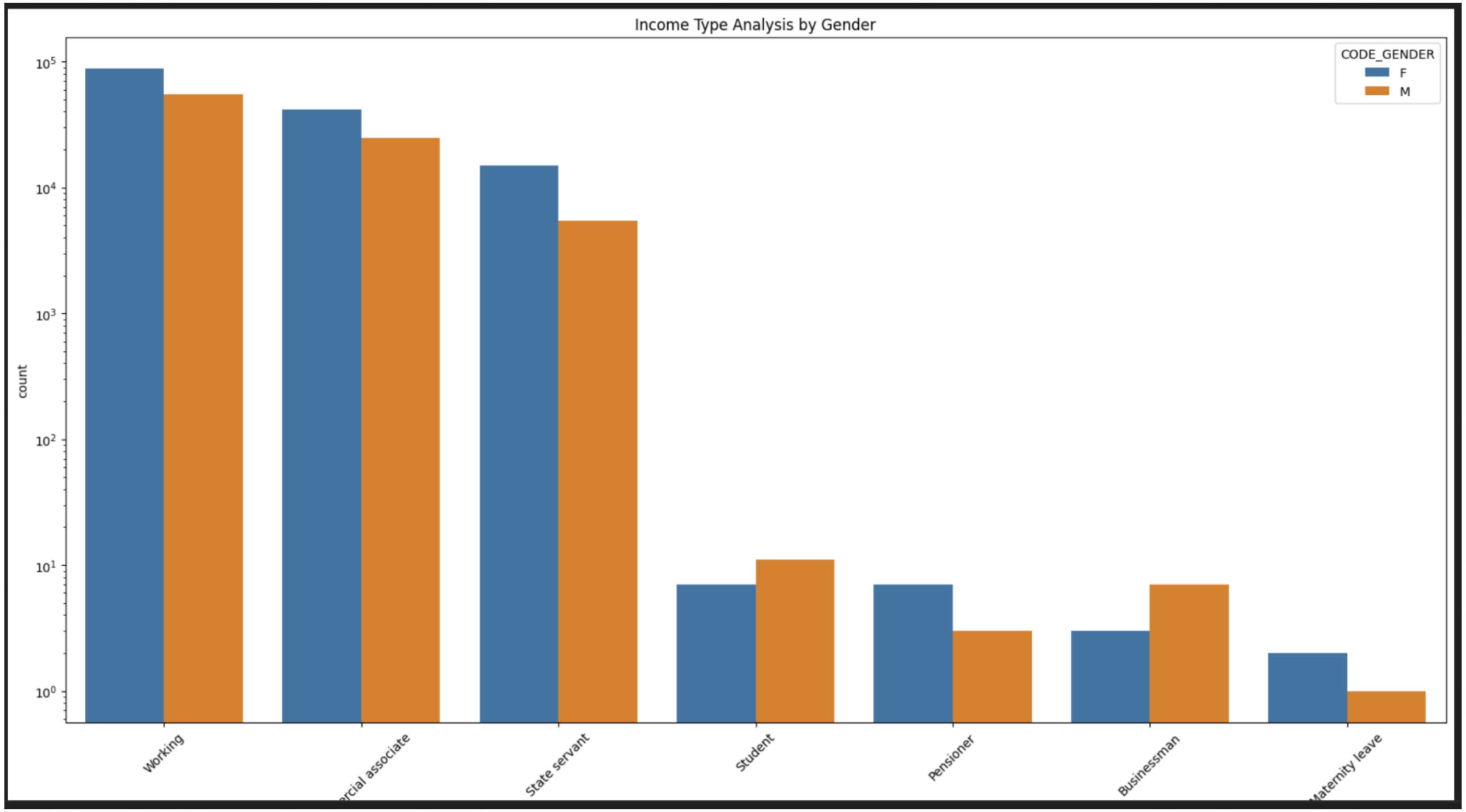


DISTRIBUTION OF INCOME RANGE



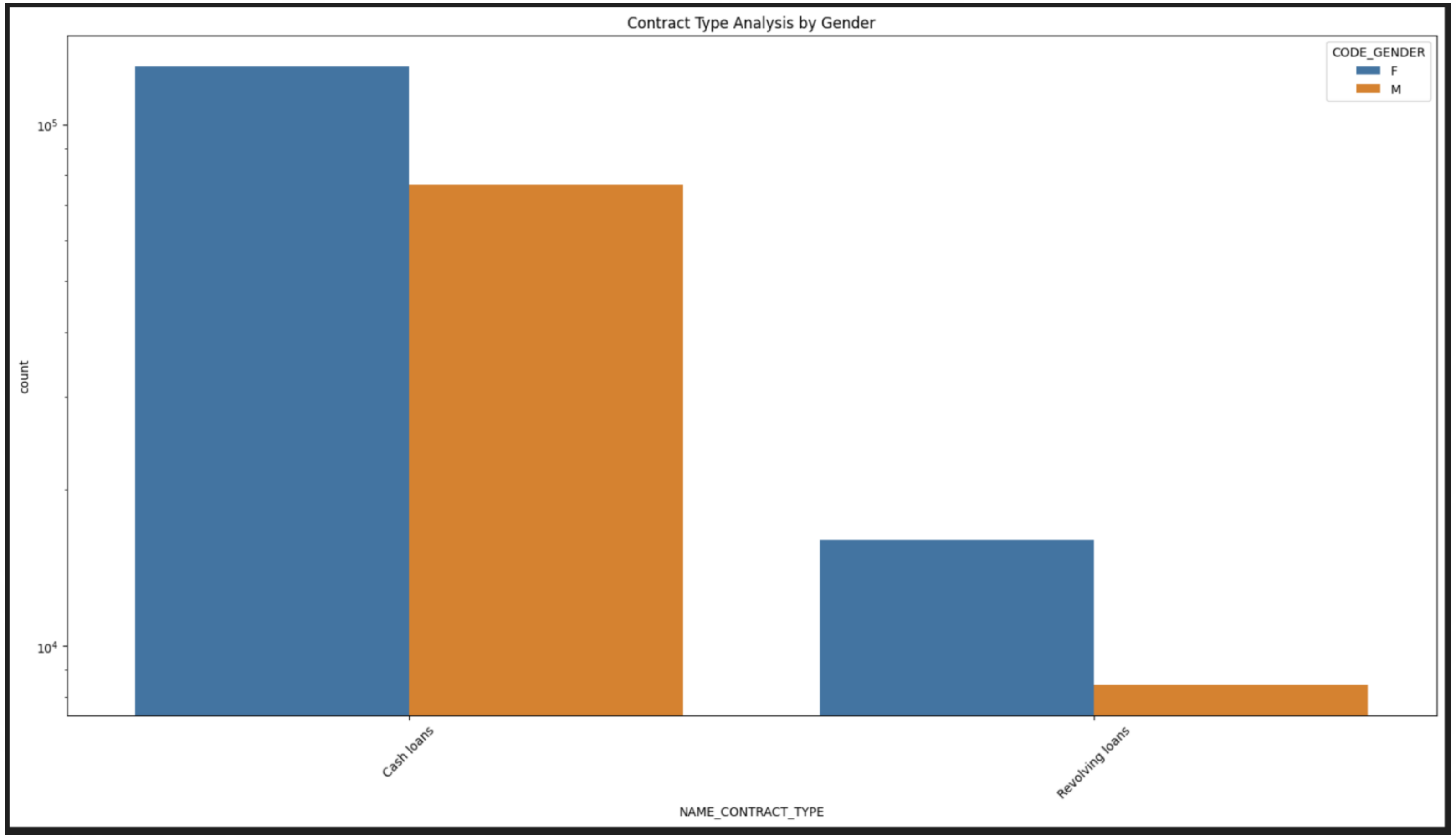
1. Female counts are higher than male
2. Females have more credit range than males
3. Income Range above 400000 is less

DISTRIBUTION OF INCOME TYPE



1. For income type ‘working’, ’commercial associate’, and ‘State Servant’ the number of credits are higher than others.
2. For this Females are having more number of credits than male.

DISTRIBUTION OF CONTRACT TYPE



1. For contract type ‘cash loans’ is having higher number of credits than ‘Revolving loans’ contract type.
2. For this also Female is leading for applying credits.

Conclusion

1. The banks should target following contract type for successful payments.
    a. Students
    b. Pensioner
    b. Businessman

2. ‘Working’ income type should be avoided as they are having most number of unsuccessful payments as well as "laborers" .

3. ‘Repair’ is having higher number of unsuccessful payments on time.

4. "Co-op apartment type" are ones facing high payment difficulties while "office apartment" type loan is good and high in successful payments

5. High percentage of "secondary/special education" type face difficulties in re-paying

6. With respect to gender Male is more defaulter than female