Home Credit Scorecard Model

Created By : Vicky Jodie

LIST OF CONTENTS :

1. Background
2. Problem Statement
3. Exploratory Data Analysis (EDA)
4. Data Pre-processing
5. Model Evaluation
6. Conclusion

Link to sourcecode : https://github.com/vickyjodie/VIX_HCI

1. Background

1. Background

Home Credit is currently using various statistical methods and Machine Learning to make credit score predictions. At this time, I was asked to find the maximum potential from the data owned by Home Credit Indonesia. This is done with the aim of **ensuring that customers who are able to make repayments are not rejected when applying for loans, and loans can be provided with principal, maturity, and repayment calendars that will motivate customers to succeed.**

2. Problem Statement

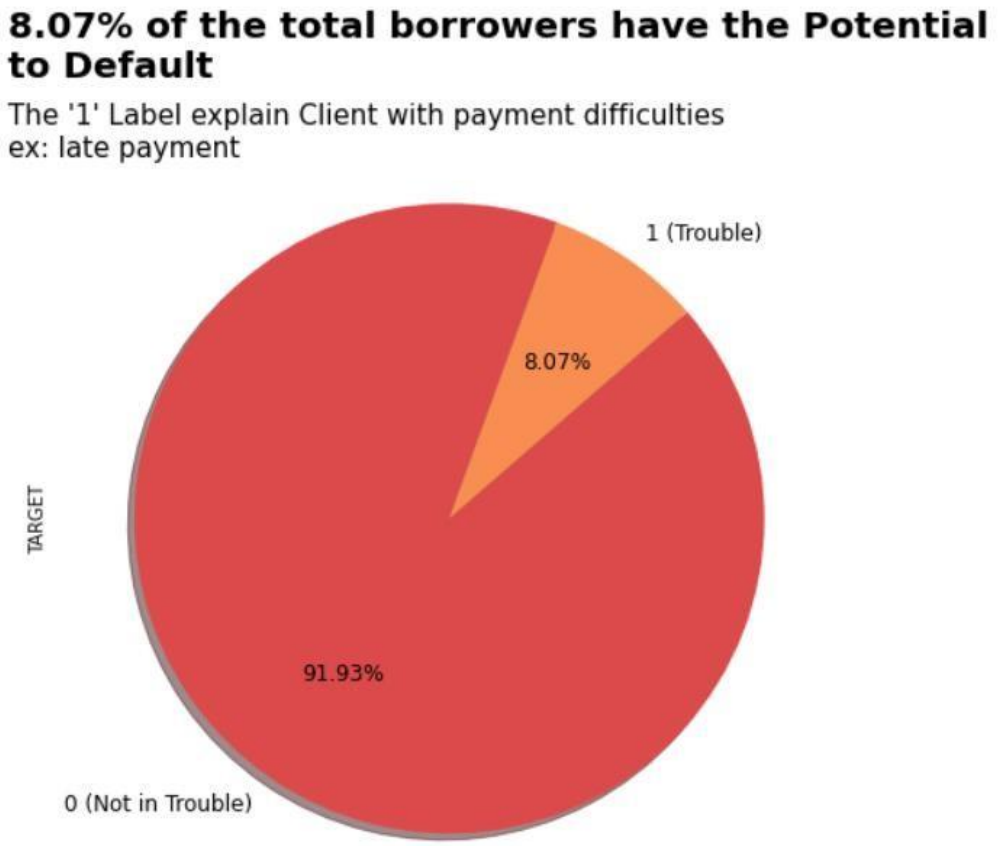
2. Problem Statement

Problem statement:
The loss generated by non-paying customers is quite high. **(8,07%)**

Goals:
Predicting potential customers who can borrow money with minimum risk

Objective:
Help minimize these losses by **using predictive models** to predict customers who are likely to default.
Use **AUC** and **Kolmogorov-Smirnov (KS)** with target:
- AUC : 0,7
- KS : 0,3

Business Metrics:
Lost Given Default (LGD)

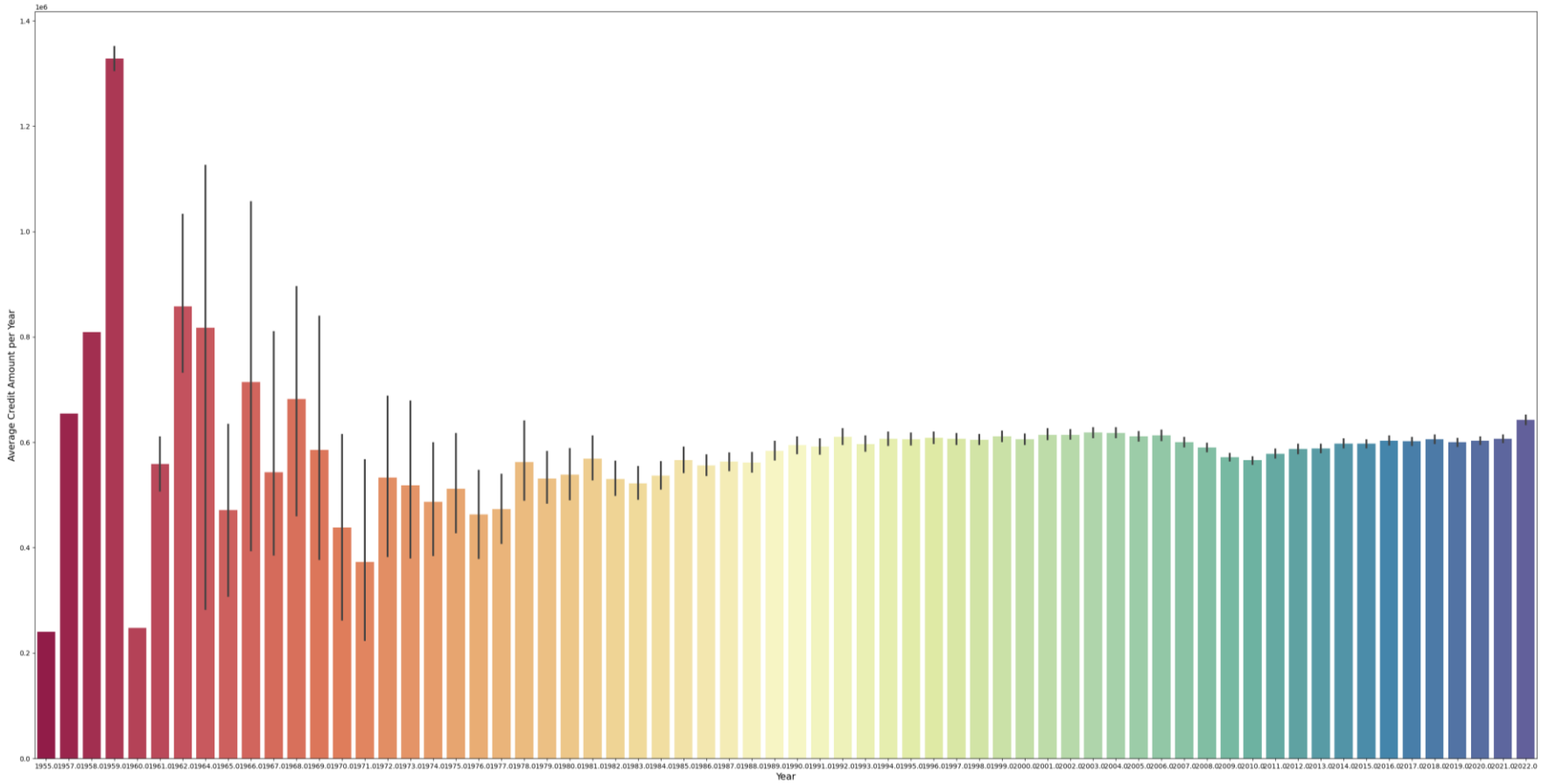


3. Exploratory Data Analysis (EDA)

3. Exploratory Data Analysis (EDA)

Generally, the nominal loan lending to customers is stagnant

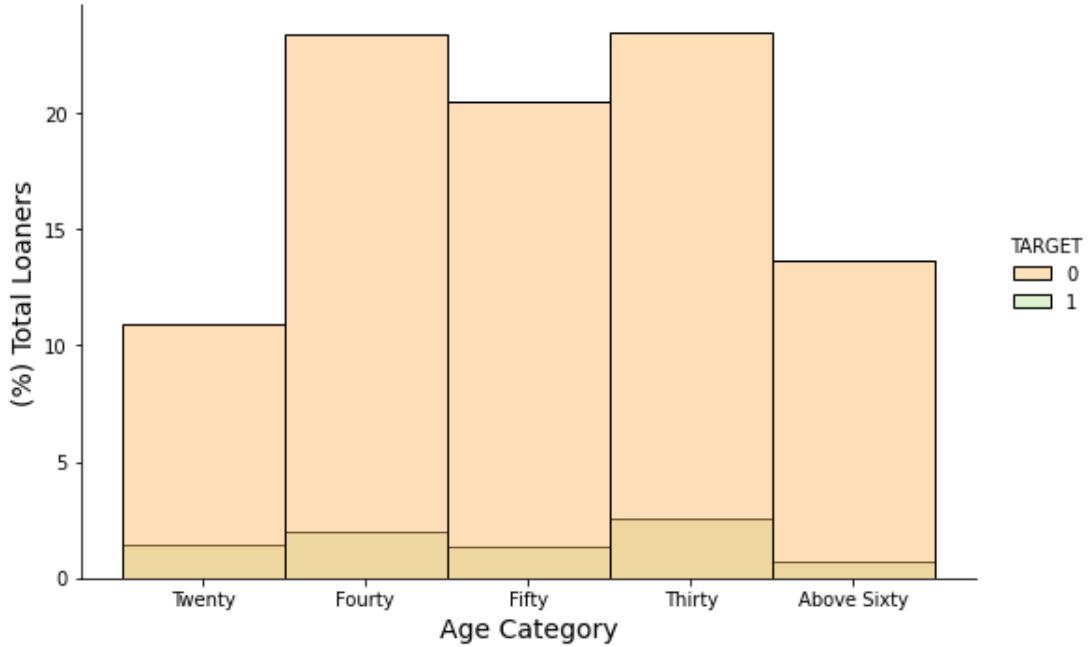
The nominal credit chart has tended to stagnate since 1978 and is expected to do so until 2021. It appears to have increased by 2022. The company can investigate this further.



3. Exploratory Data Analysis (EDA)

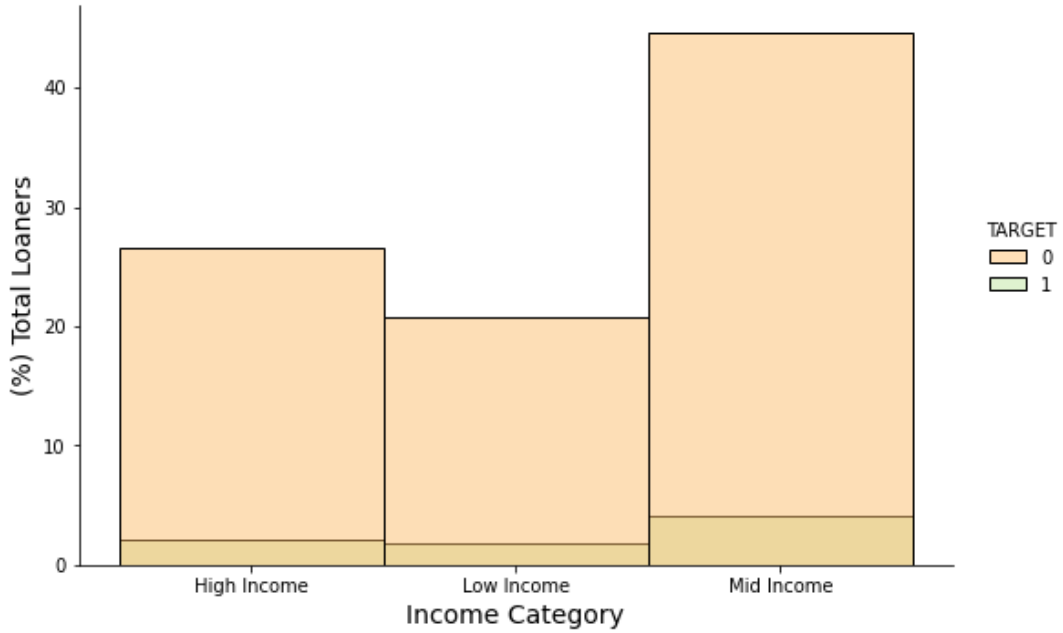
3% Loaners in Thirty more likely to default then others

In the graphic image below, Every Age Category is have
Potential to default



4% Loaners in Mid Income more likely to default then others

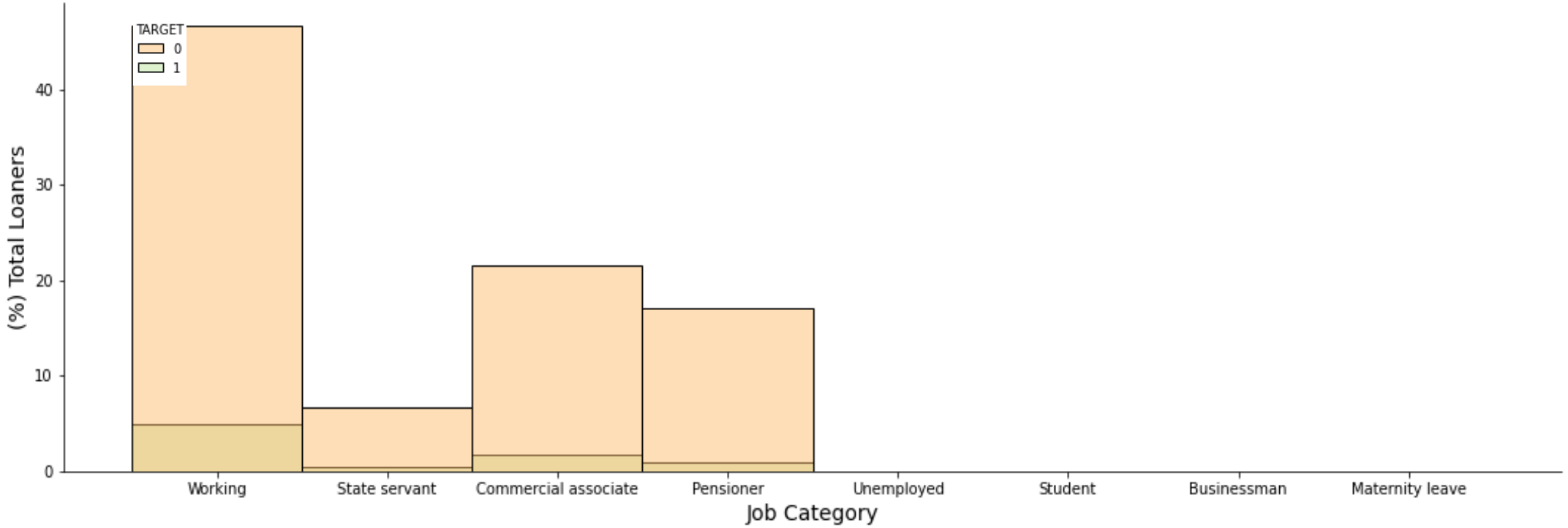
In the graphic image below, Every Income Category is have
Potential to default



3. Exploratory Data Analysis (EDA)

7% Loaners in State Servant more likely to default then others

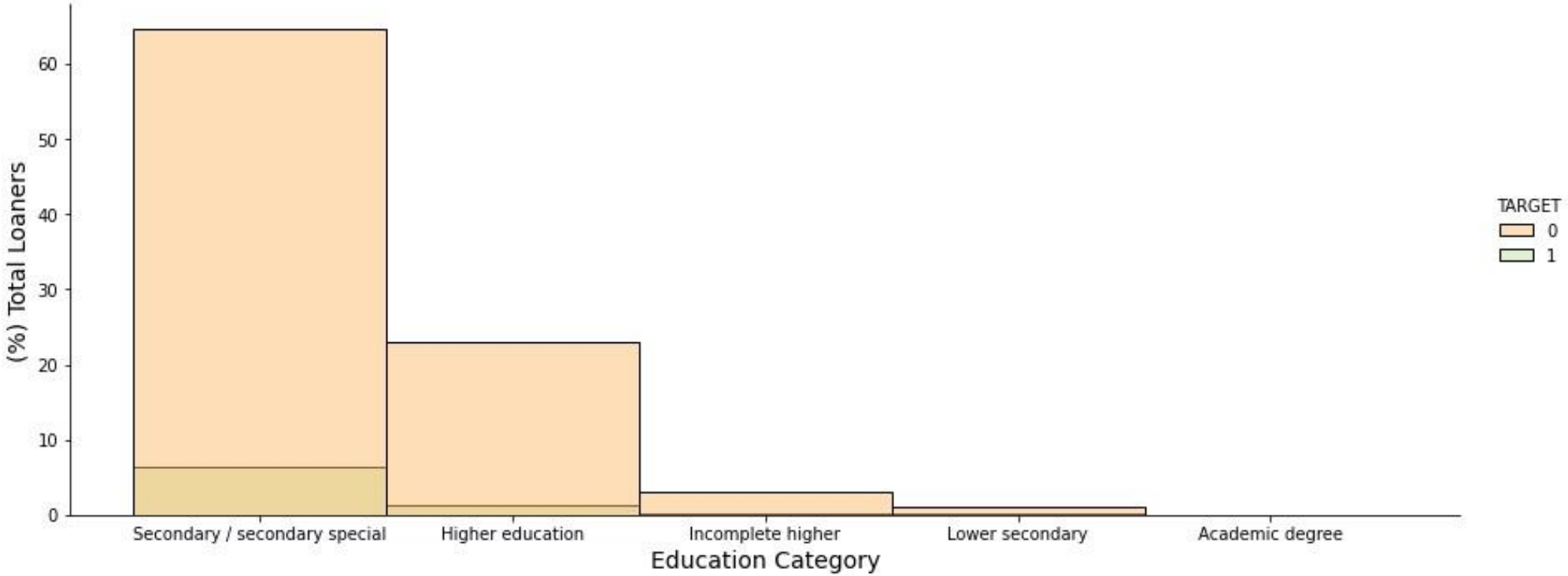
In the graphic image below, Some of loaner by job category is have
Potential to default



3. Exploratory Data Analysis (EDA)

6% Loaners in Secondary / secondary special more likely to default then others

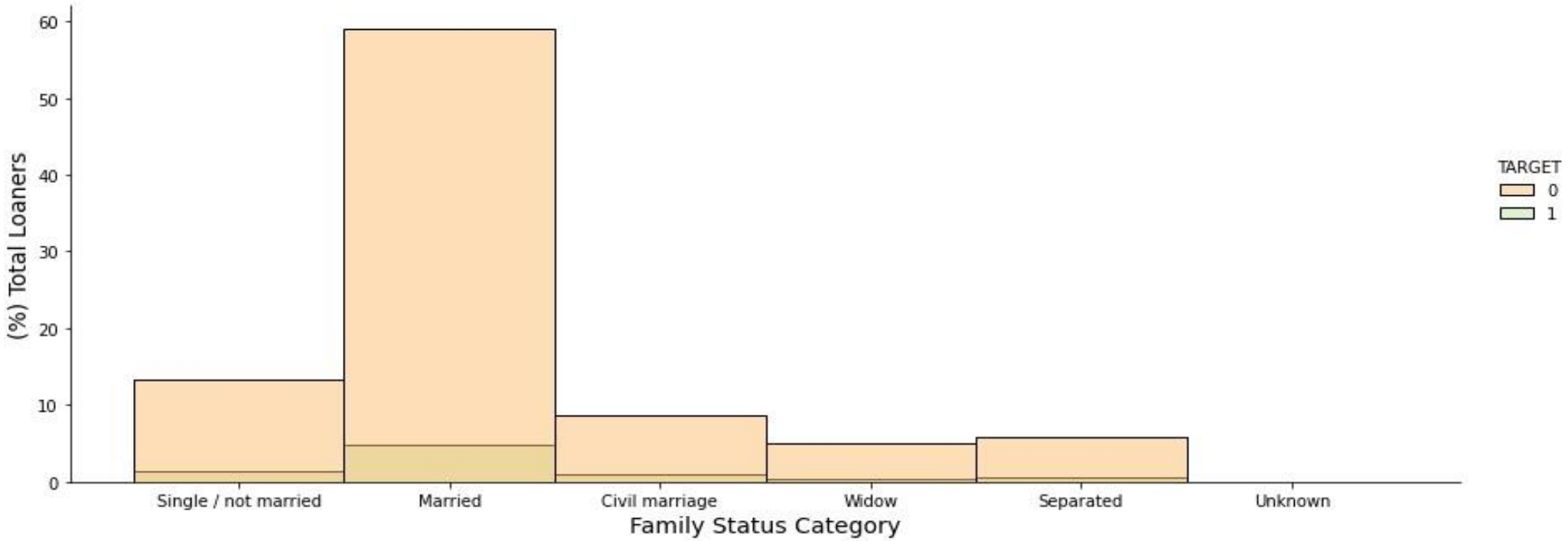
In the graphic image below, Some of loaner by edu category is have
Potential to default



3. Exploratory Data Analysis (EDA)

Loaner who is married is more likely to default then others

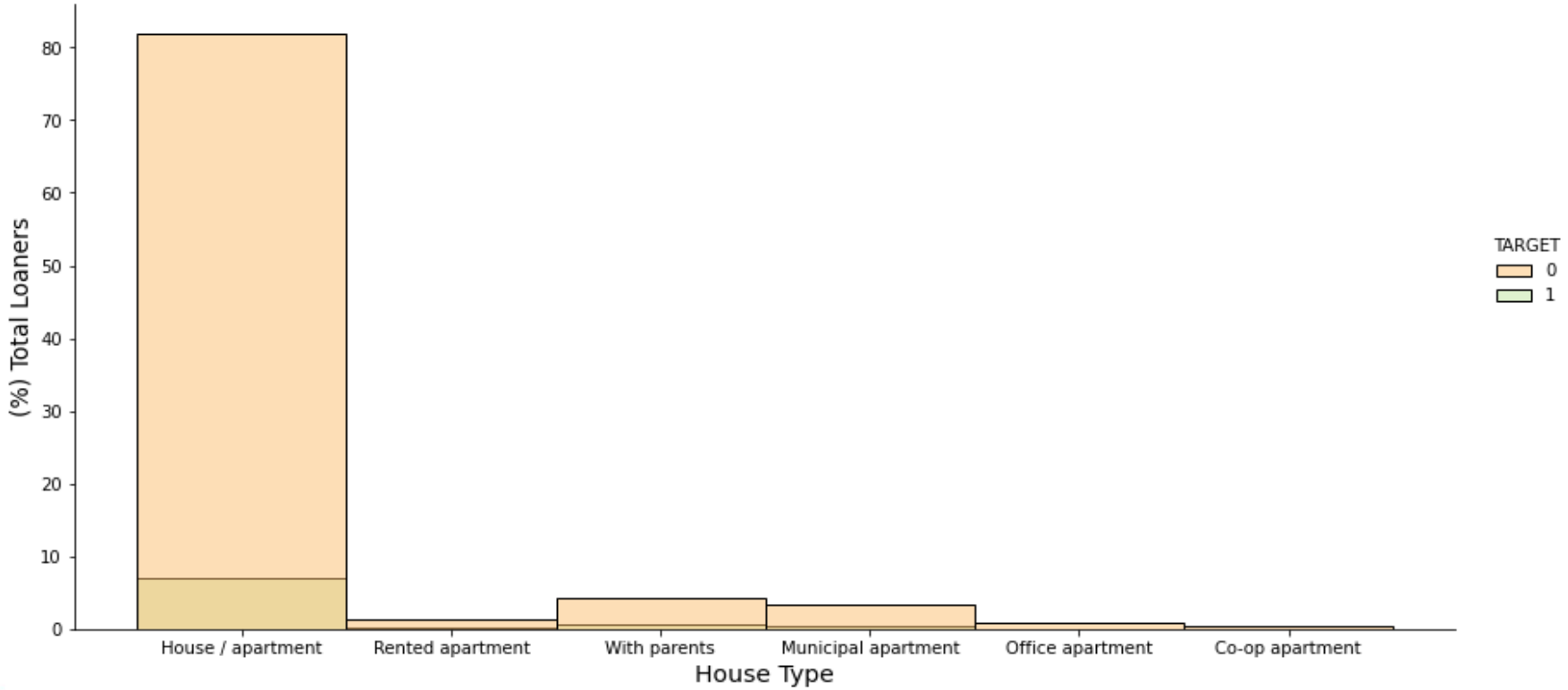
In the graphic image below, Some of loaner by family status category is have Potential to default



3. Exploratory Data Analysis (EDA)

Loaner who have house is more likely to default then others

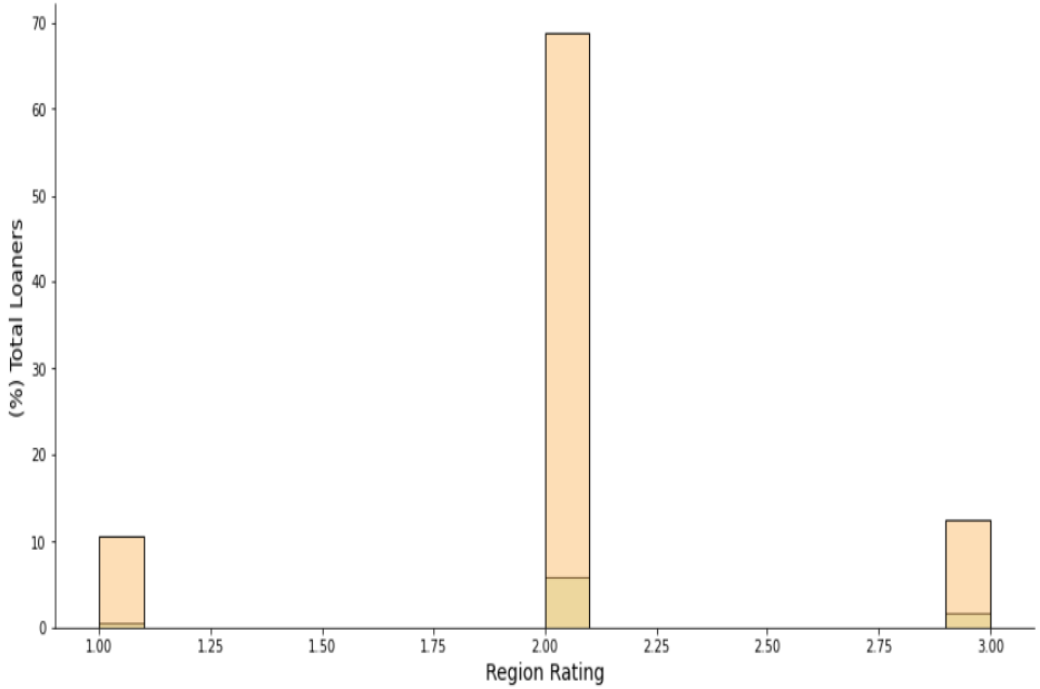
In the graphic image below, Some of loaner who have house (7%) is have Potential to default



3. Exploratory Data Analysis (EDA)

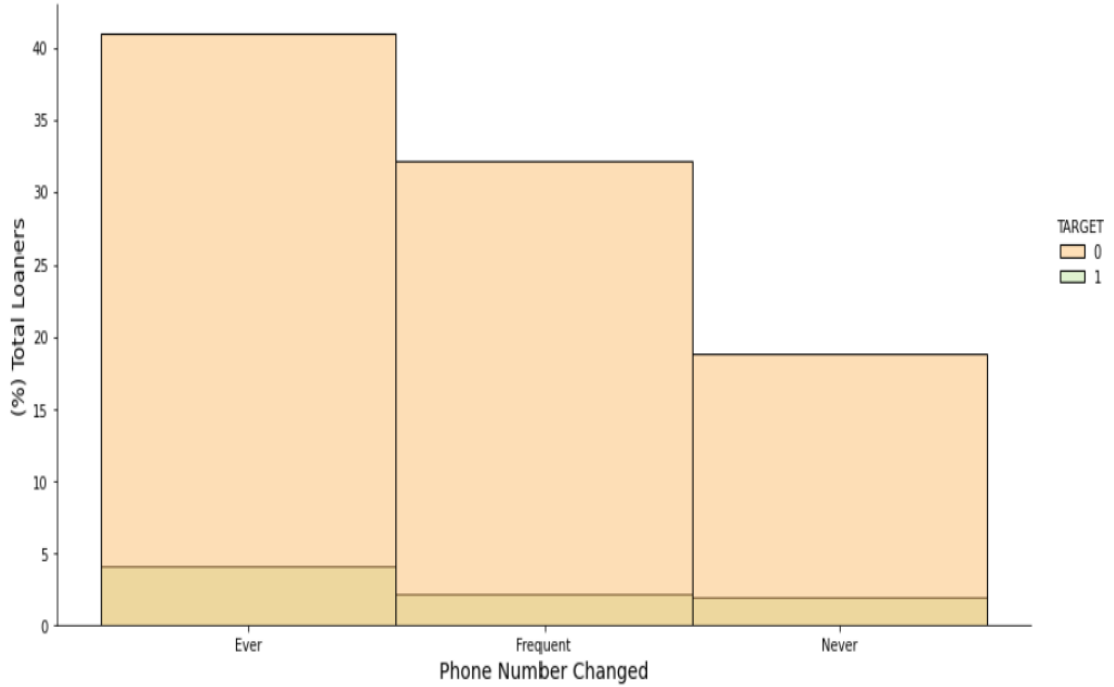
Loaner who in 2 more likely to default then others

In the graphic image below, Some of loaner with rating 2 (6%) is have more Potential to default



Loaner who ever change their phone more likely to default then others

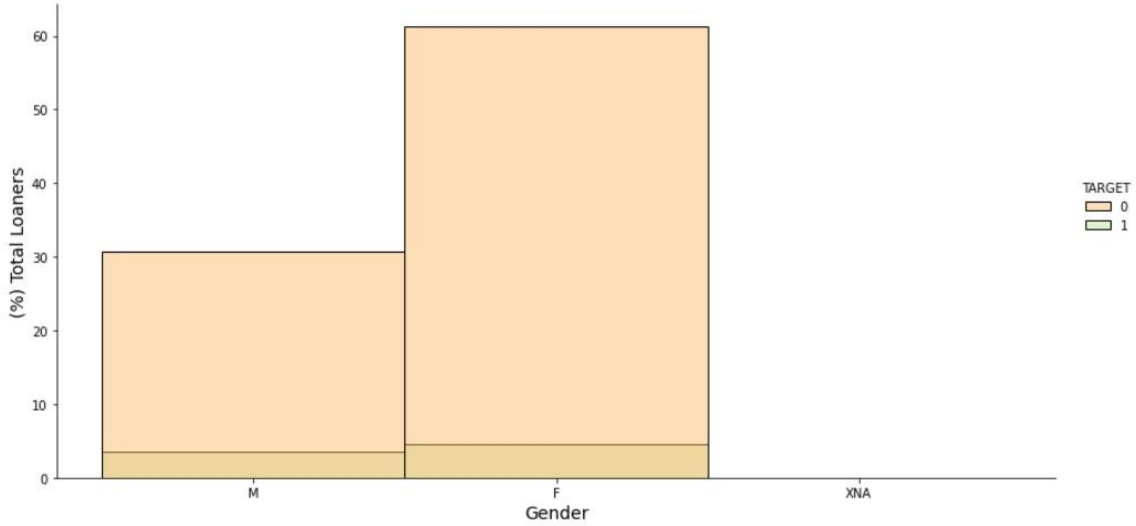
Some of loaner who ever change their phone (1-3 times) is have more Potential to default (4%)



3. Exploratory Data Analysis (EDA)

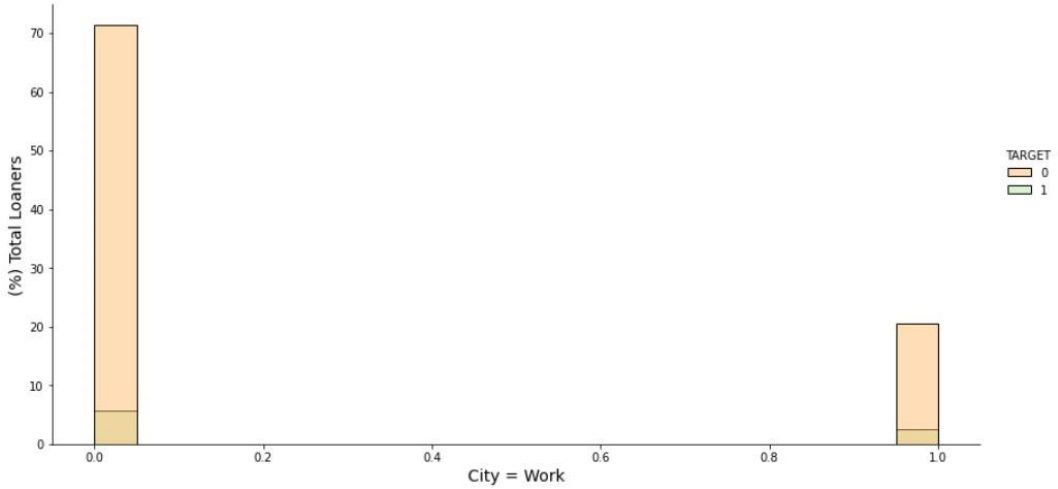
Female Loaners is have more potential to default

14170 Female Loaners is have more Potential to default (5%) then the Male Loaners



Loaners who have an address that is not the same as their place of work is have more potential to default

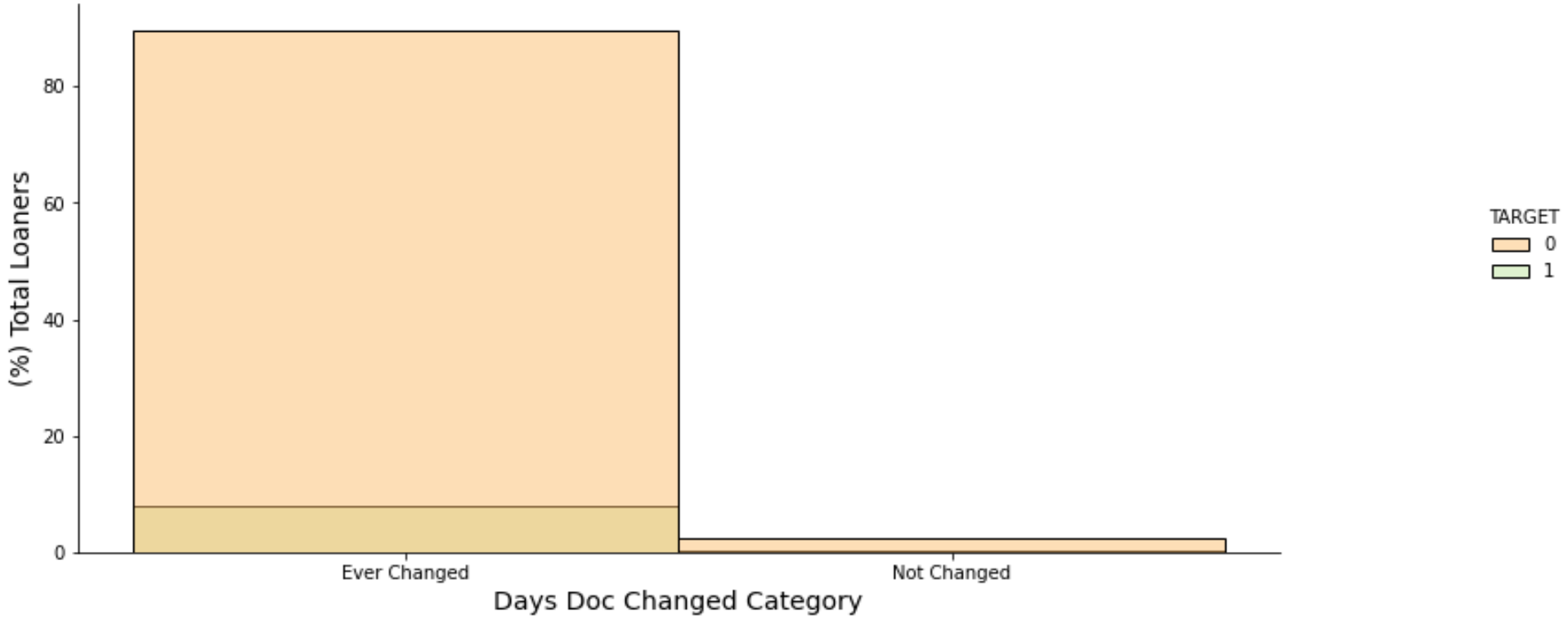
17305 Loaners who have an address that is not the same as their place of work is have more Potential to default (6%)



3. Exploratory Data Analysis (EDA)

Loaner who changed the document in 'Ever Changed' more likely to default then others

24035 Total Loaners who ever changed the identity Document
have more Potential to default (8%)



4. Data Pre-processing

4. Data Pre-processing

Drop Missing Value (<68.4% from Dataset)

	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4

Input Missing Value with Single Imputer

```
#create two DataFrames, one for each data type
data_numeric = df_1[nums]
data_categorical = pd.DataFrame(df_1[cats])


from sklearn.impute import SimpleImputer
imp = SimpleImputer(missing_values=np.nan, strategy='median', verbose=0)
data_numeric = pd.DataFrame(imp.fit_transform(data_numeric), columns = data_numeric.columns)

cimp = SimpleImputer(missing_values=np.nan, strategy='most_frequent', verbose=0)
data_categorical = pd.DataFrame(cimp.fit_transform(data_categorical), columns = data_categorical.columns)


#you could do something like one-hot-encoding of data_categorical here

#join the two masked dataframes back together
df_1 = pd.concat([data_numeric, data_categorical], axis = 1)
```

4. Data Pre-processing

Feature Encoding

```
cats_le = df_for_model[['NAME_CONTRACT_TYPE','CODE_GENDER','FLAG_OWN_CAR',
                        'FLAG_OWN_REALTY','EMERGENCYSTATE_MODE',
                        'NAME_EDUCATION_TYPE','AMT_INCOME_CAT',
                        'DAYS_ID_PUBLISH_CAT', 'PHONE_CHANGE_CAT']]
cats_ohe = df_for_model[['NAME_TYPE_SUITE','NAME_INCOME_TYPE',
                         'NAME_FAMILY_STATUS','OCCUPATION_TYPE',
                         'WEEKDAY_APPR_PROCESS_START','ORGANIZATION_TYPE',
                         'WALLSMATERIAL_MODE','HOUSETYPE_MODE',
                         'AGE_CATEGORY','NAME_HOUSING_TYPE']]
```

```
for i in cats_le:
    df_for_model[i]= LabelEncoder().fit_transform(df_for_model[i])
for cat in cats_ohe:
    onehots = pd.get_dummies(df_for_model[cat], prefix=cat)
    df_for_model = df_for_model.join(onehots)
```

MinMaxScaler

```
from sklearn.preprocessing import MinMaxScaler
for i in numss:
    df_for_model[i] = MinMaxScaler().fit_transform(df_for_model[i])
```

Feature Selection using SelectKBest

```
x_k = df_for_model[[col for col in df_for_model.columns if col not in ['TARGET']]]
y_k = df_for_model['TARGET']

print("Feature data dimension: ", x_k.shape)
```

```
Feature data dimension:  (307511, 236)
```

```
select = SelectKBest(score_func=f_regression, k=15)
z = select.fit_transform(x_k, y_k)

print("After selecting best 15 features:", z.shape)
```

```
After selecting best 15 features: (307511, 15)

Selected best 15:

['DAYS_BIRTH' 'DAYS_ID_PUBLISH' 'REGION_RATING_CLIENT'
 'REGION_RATING_CLIENT_W_CITY' 'REG_CITY_NOT_WORK_CITY' 'EXT_SOURCE_1'
 'EXT_SOURCE_2' 'EXT_SOURCE_3' 'DAYS_LAST_PHONE_CHANGE' 'AGE'
 'DAYS_LAST_PHONE_CHANGE_FIXED' 'DAYS_ID_PUBLISH_FIXED' 'CODE_GENDER'
 'NAME_EDUCATION_TYPE' 'NAME_INCOME_TYPE_Working']
```

5. Model Evaluation

5. Model Evaluation

Train Test Split & Handling Imbalance Data

Feature :

'DAYS_BIRTH' 'DAYS_ID_PUBLISH'
'REGION_RATING_CLIENT'
'REGION_RATING_CLIENT_W_CITY'
'REG_CITY_NOT_WORK_CITY'
'EXT_SOURCE_1'
'EXT_SOURCE_2' 'EXT_SOURCE_3'
'DAYS_LAST_PHONE_CHANGE'
'CODE_GENDER'
'NAME_EDUCATION_TYPE'
'NAME_INCOME_TYPE_Working'

Target :

'TARGET'

Imbalance using SMOTE

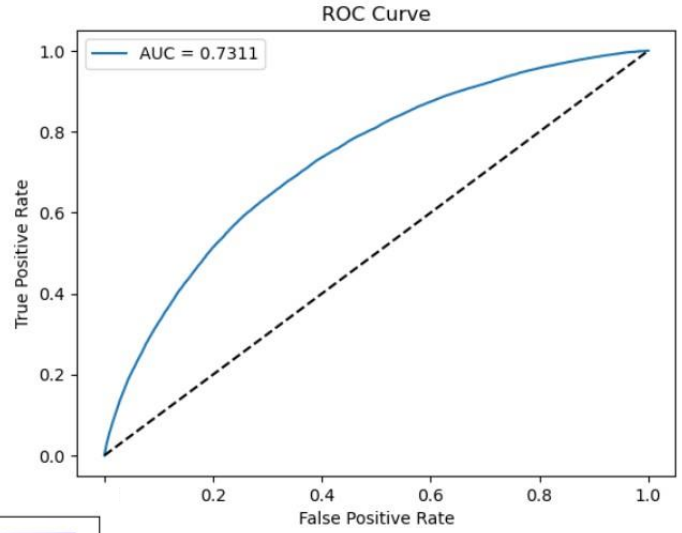
```
X, y = over_sampling.SMOTE(0.5).fit_resample(X, y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

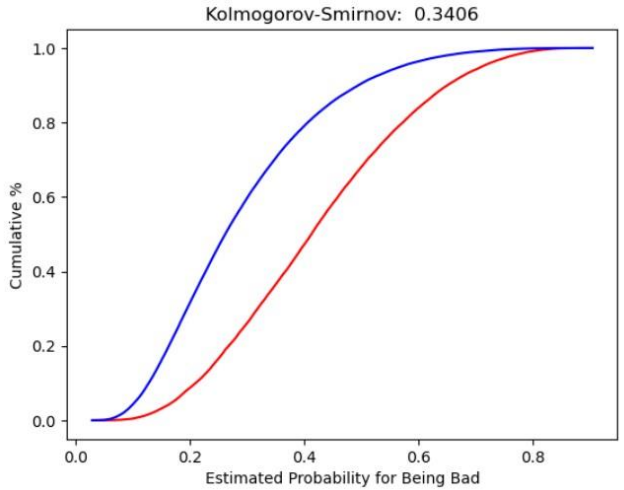
5. Model Evaluation

Model :
Logistic Regression

Model Evaluation:
AUC & KS

Evaluation	Model
	Logistic Regression
AUC (Train Set)	0.73
AUC (Test Set)	0.73
KS (Test Set)	0.3406





6. Conclusion

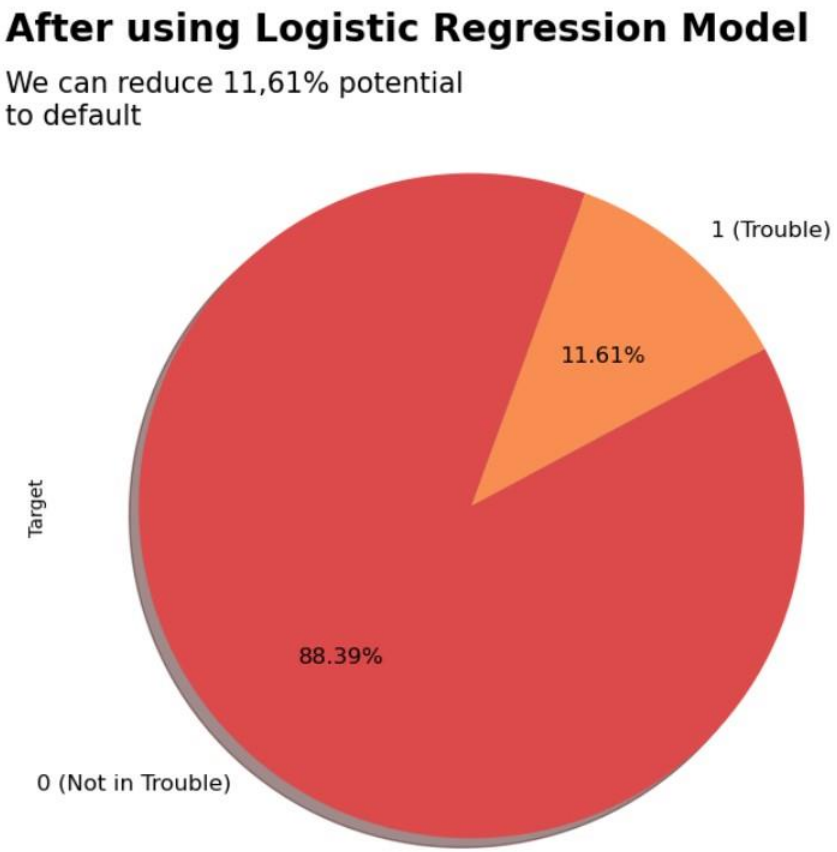
6. Conclusion

Before and After using model:

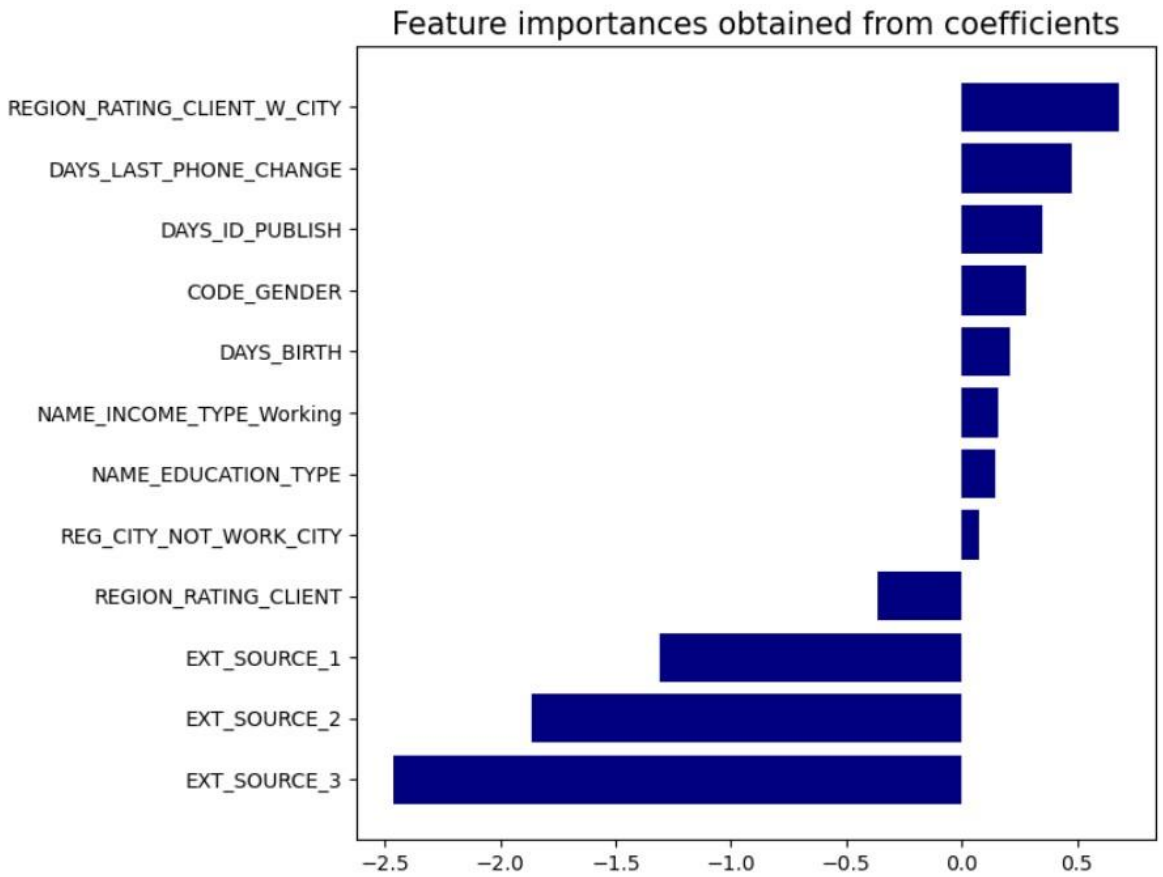
Before we use the model, we know that **8,07 %** of Customers indicated failure from data train.

We can predict that **11.61 %** of the Test Data will default based on our modeling. As a result, customers who are expected to fail to pay can have their **applications canceled**.

Loss generate (Before using Model) Using Previous Data	Loss generate (After using Model) Using Data Test
8,07% Default	11,61% Predicted Default



Feature Importance



Top 3 Feature Importance that give more impact is:

1. Region Rating Client W City
2. Days Last Phone Change
3. Days ID Publish

Insights

Credit Score Card with Logistic Regression Model (Based on Feature Importance and EDA):

1. The most impactful feature in the following model is **"Region Rating Client City."** According to EDA, the value in this column represents a rating of the city in which the customer resides. It should be noted that those who live in a "2" rating area are more vulnerable to credit default.

2. The **"Days Last Phone Change"** feature is the second most important. In EDA, we divide the customer's phone number changes into three categories: "Ever," "Once," and "Never." According to the EDA, people who ever changed their phone number are more likely to default.

3. One of the top three features in this dataset is **"Days ID Publish."** We categorize people into two groups: those who have not changed their identity documents (Not Changed) and those who have (Ever Changed). This needs to be looked into further because it could be a scam.

4. If some of the top three features are insufficient, features that are not in the top three become additional features.

Link to sourcecode : https://github.com/vickyjodie/VIX_HCI

THANK YOU!

Vicky Jodie
vickyjwang9696@gmail.com
linkedin.com/in/vicky-jodie

Let’s work together!