Credit Card Approval Prediction

By: Randa Hesham

Agenda

01 Objectives

- Business Challenge
- Task Aim

02 Dataset

- Application records
- Credit records

03 EDA

- Data Glimpse
- Data cleaning
- Data visualization

04 Feature Engineering

- Create new features
- Outliers Analysis
- Vintage Analysis

05 Data Preparation

- Feature Scaling
- Handling Imbalanced data

06 Modeling & Results

Objectives

Business Challenge

- Banks receive a lot of applications for issuance of credit cards.
- Many of them rejected for many reasons:
  - High-loan balances
  - Low-income levels
  - Too many inquiries on an individual's credit report.
- Manually analyzing these applications is error-prone and a time-consuming process.

Task Aim

- Build a machine learning model to predict if an applicant is 'good' or 'bad' client.
- The definition of 'good' or 'bad' is not given.
- Use some technique, such as vintage analysis to construct you label. Also, handle the imbalanced data.

Dataset

- **application_record.csv**
  contains appliers personal information, which you could use as features for predicting.

  - It has 18 columns [9 categorical vars, 9 numerical vars]

- **credit_record.csv**
  Users' behaviors of credit card.

  - It has 3 columns [1categorical var, 2 numerical vars]

EDA [Data Glimpse]

- **application_record metadata**
  - Number of datapoints for application records: 438557
  - Number of unique clients in dataset: 438510

  This means it has duplicates.

- **credit_record metadata**
  - Number of datapoints for credit records: 1048575
  - Number of unique clients in dataset: 45985

  It means that there are repeating entries for different monthly values and status.

- There are fewer customers than applications in the credit record dataset. The intersection is 36,457 customers.

Note: There is a big difference between the two unique numbers of the two tables. It's obvious that there are less customers than applications.

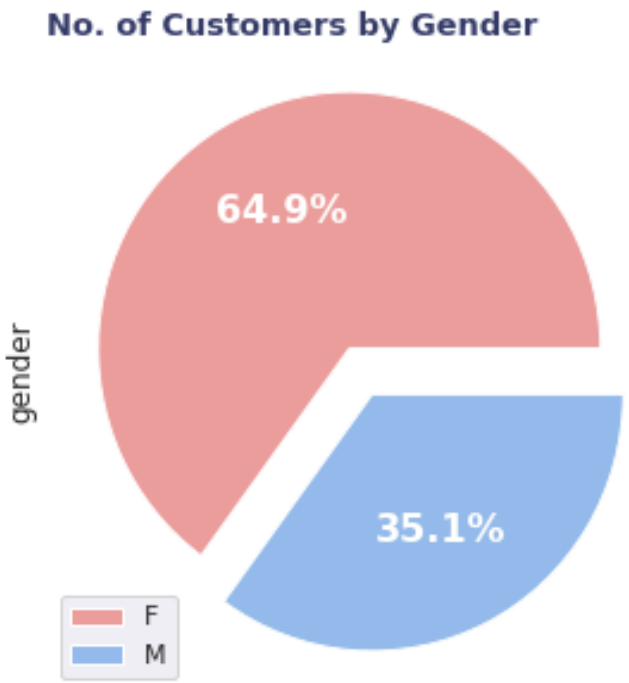
EDA [Data Cleaning]

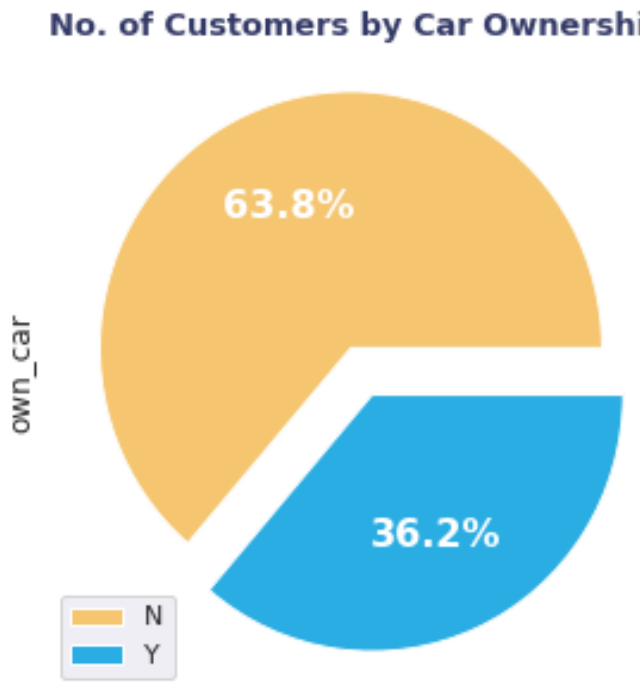
- **Data cleansing has been done through the following steps:**

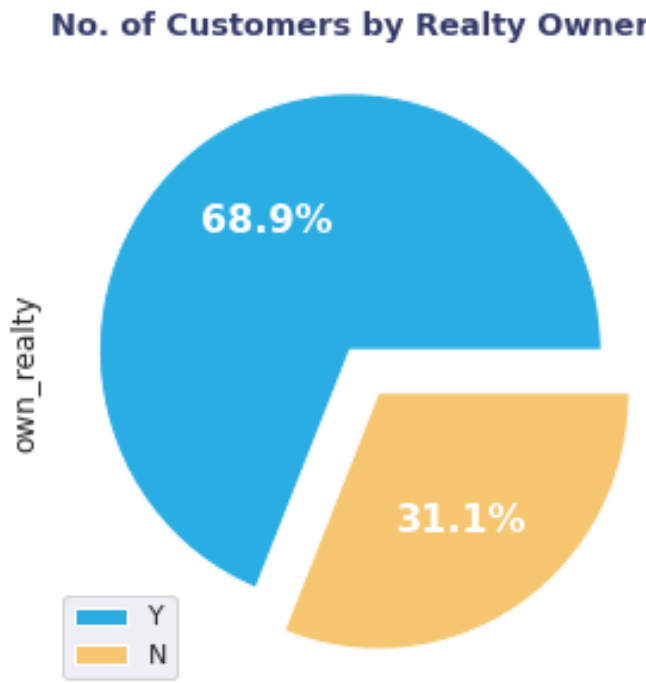
  - Handling Missing data
    - Occupation Type is the only variable that has NaN, so we fill those values with “Others”.
  - Removing duplicate and Unnecessary Data.
  - Dropping unnecessary features such as FlLAG_MOBILE
  - Renaming columns for simplicity’s sake
  - Adding customer’s age and work experience features .
  - Converting non-numeric featurs to numeric.

EDA [Data Visualization]

Customer distribution by gender, car and realty ownership:

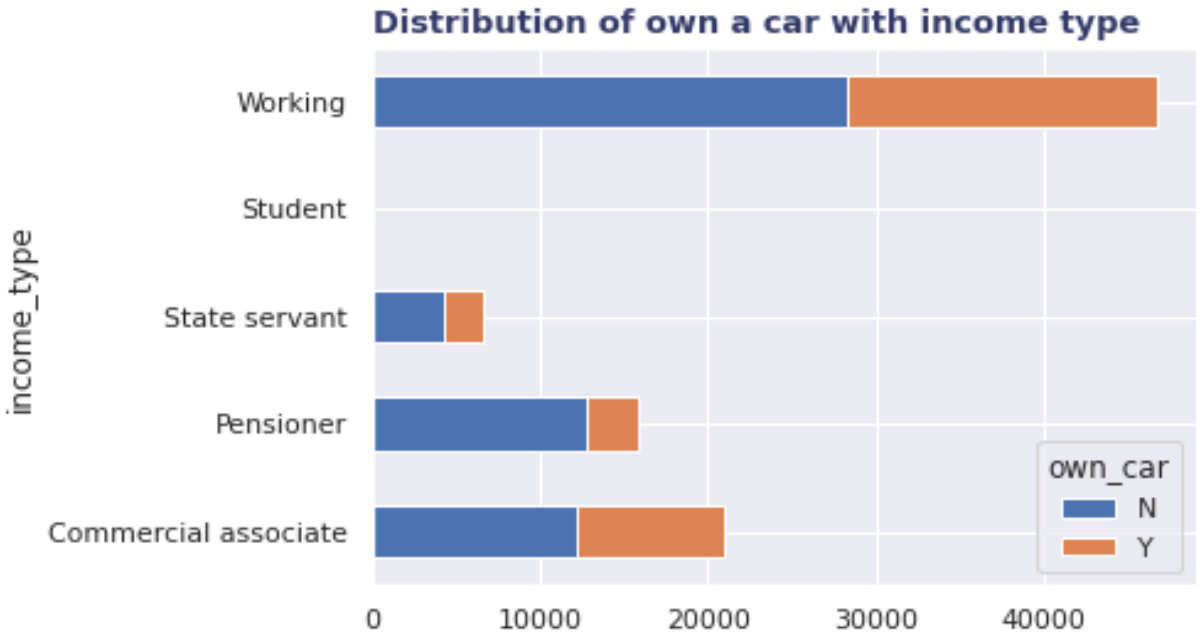




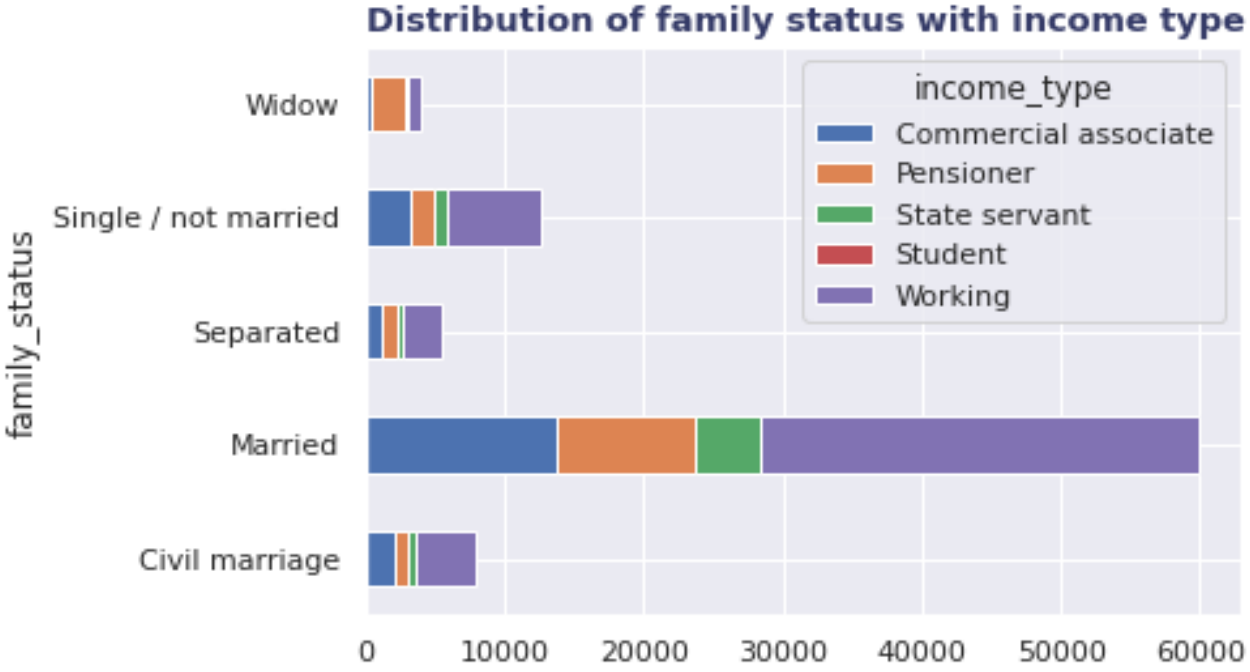


- Gender of the customers are 64% females and 35% males.
- 36% of customers own cars
- 68% of customers own a realty

EDA [Data Visualization]

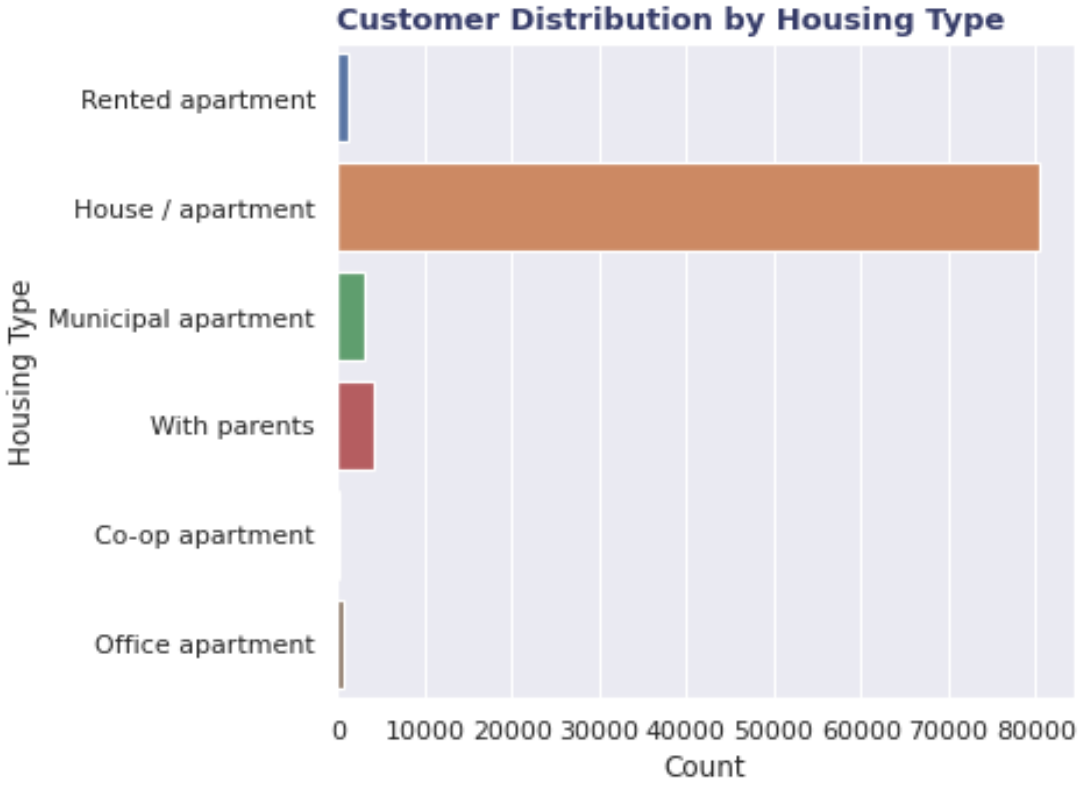


- About 50% are working customers and about 30% out of them does not own a car.
- About 20% are commercial associates and about 13% of them does not own a car.
- About 15% are pension and about 13% of them does not own a car.

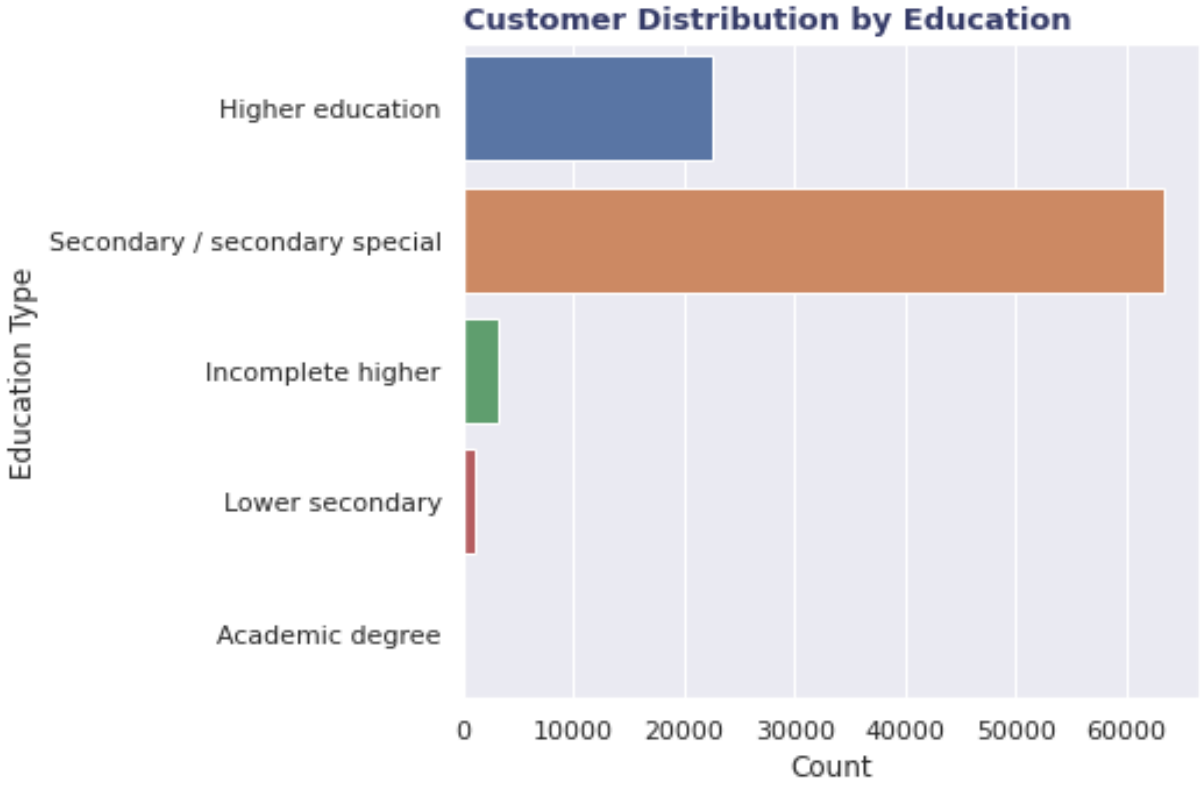


- About 60% are married and about 30% of them are working, about 15% of them are commercial associates, about 10% are pensioner and less than 10% are state servant.

EDA [Data Visualization]

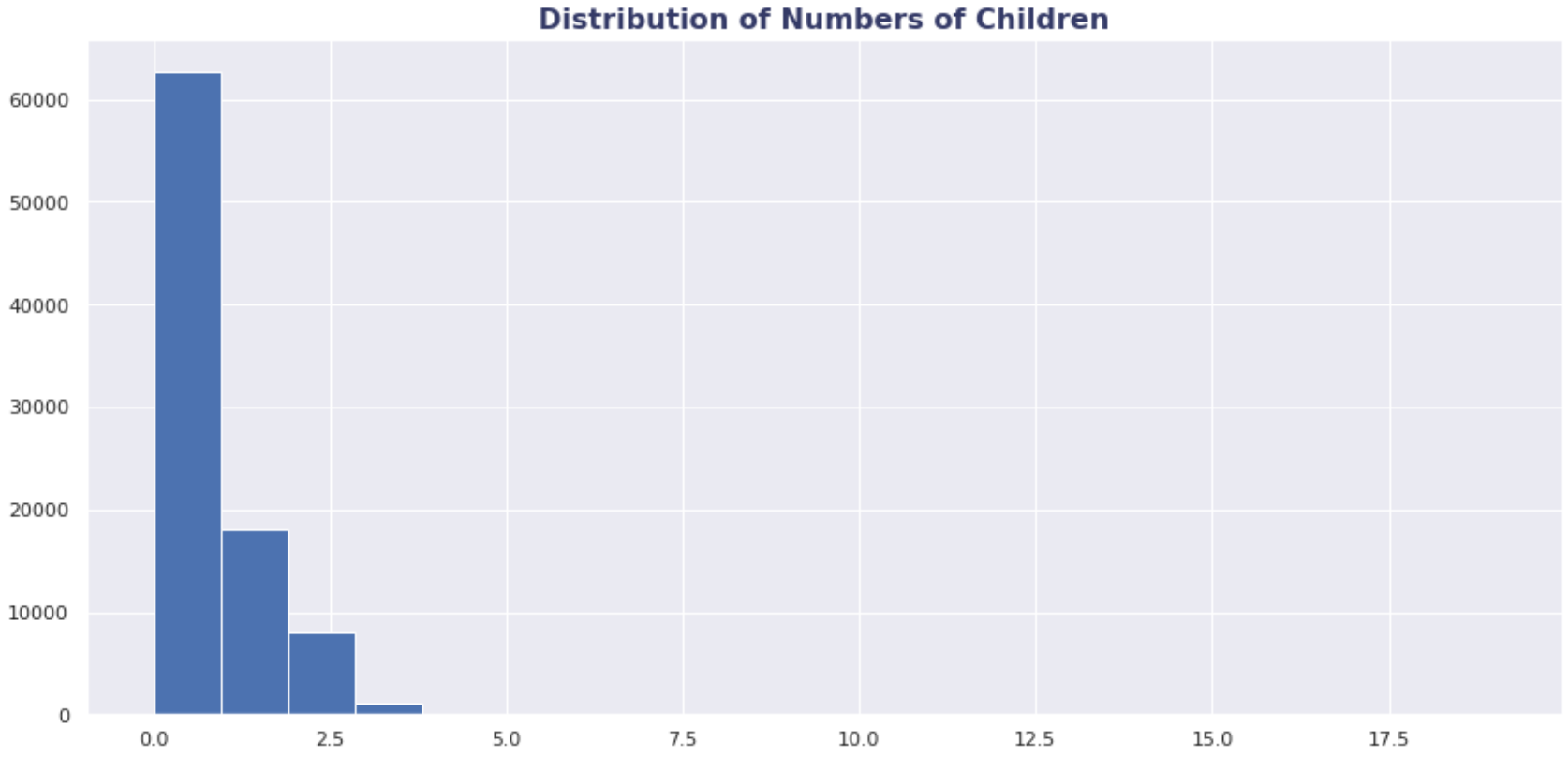


- About 80% have a house, and less than 5% are living with parents.



- About 65% have secondary education, about 23% have higher education and less than 5% have incomplete higher education.

EDA [Data Visualization]



- About 65% of customers doesn't have children, 18% have one child and 8% have 2 children.

Note: the distribution is right skewed (there're outliers)

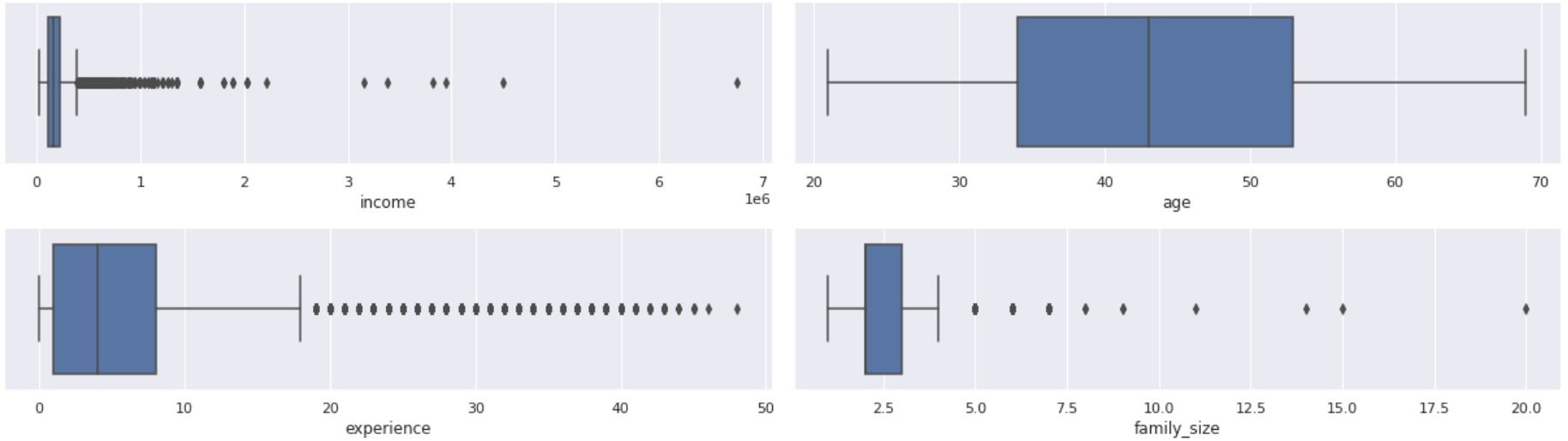
Feature Engineering

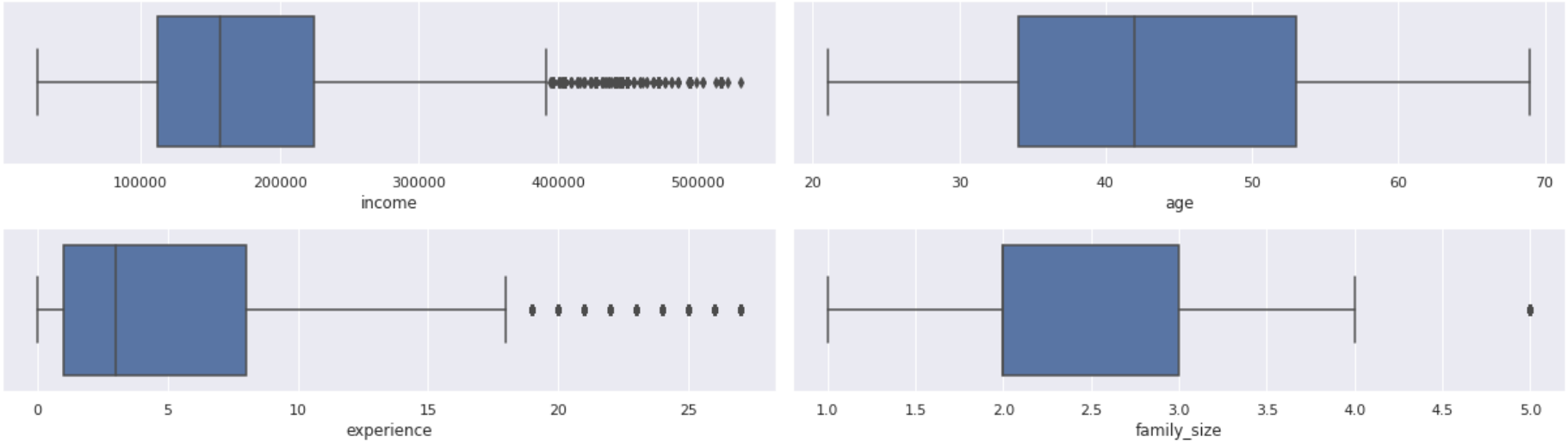
Feature Engineering has been done through the following steps:

- Creating new features
    - Get the month when users open their accounts.
    - Get household size.
    - Convert birth days into positive numbers and years.
    - Convert employment days into positive numbers and years
- Apply one hot encoding
- Removing Outliers by using z-scores.
- Create target column

Feature Engineering [Outliers Analysis]

Before removing the outliers





After removing the outliers

Feature Engineering [Vintage Analysis]

- The vintage analysis measures the performance of a portifolio in different periods of time after the loan (or credit card was granted).
- We will use it to create the target variable.
- If a client had no loans throughout the initial approval of the credit card account, by default, this would be considered a good client as well. To identify a bad client, the number of past dues would exceed the number of loans paid off or if the client only has past dues.

Data Preparations

Feature scalling

- We scaled the data before splitting them into test and train because we want the model to receive values in the same range for both training and testing.
- Since it's a classification task we used standarization.

Handling Imbalanced Data

- We used over-sampling cause we had small data.
- We over-sampled the training data only. By oversampling only on the training data, none of the information in the testing data is being used to create synthetic observations. So these results should be generalizable.

Note: We recognize from the target data that the good customers are more than bad customers

Modeling & results

Modeling part have been performed using two high-performance models:

Random Forest

- Random Forest produced a significantly high accuracy scores on the train sets (100%) and test set accuracy (around 97.8% for the Random Forest).

XGboost

Accuracy	0.98
Presicion	0.97
Recall	0.98
F1_score	0.97

Thank You

LinkidIn: randahesham