10/13/2023

"Risk Mitigation through EDA: Unveiling Patterns in Client Payment Behavior for Informed Lending Decisions in Banking Sector"

Project 3: Exploratory Data Analysis using Python on Jupyter Notebook

POOJIT BHASKAR KASINA

DATA ANALYST TRAINEE & INTERN AT OESON
BATCH NO: OGTIP-BA/DA-AUS-06

OGTIP: OGTIPDAPA230

Mentor: Ms. Smruthi Mehta

Contents

Introduction: 2
Problem Statement: 2
Data Overview: 3
Analysis of Decisions in the Loan Application Process: 3
Business Objectives 3
Importing Libraries 3
Figure 1 Importing Library 4
Data Importing: 4
Figure 2 Reading the data 5
Data Cleaning: 5
Figure 3 Dropping Unnecessary Columns 6
Figure 4 Checking and Handling Missing Values in Both Datasets 6
Detecting Outliers: 6
Figure 5 Detecting Outliers 7
Figure 6 Handling XNA and XAP Values 9
Merging Datasets 9
Figure 7 Merged Data 9
Figure 8 Handling Remaining Null Values after Merging the Data 10
Figure 9 Converting Negative Values into Absolute Values 10
Figure 10 Merged Data After Cleaning 11
Data Analysis 11
Figure 11 Statistical Analysis 11
Univariate Analysis 12
Figure 12 Income Type by Gender 12
Figure 13 Gender Comparison 13
Figure 14 Car Ownership Comparison 14
Figure 15 Comparison of Defaulters and Non-defaulters 15
Figure 16 Loan by Education Type 15
Figure 17 Loan By Client Type 16
Figure 18 Loan By Contract Status 17
Figure 19 Loan applicant by Property 18
Figure 20 Loan by the distribution of applicants by marital status 18
Bivariate Analysis 19

Figure 21 Goods Price vs Contract Types Based on Target 19
Figure 22 Loan approval by Gender........ 20
Figure 23 Goods prices Vs Contract Types 20
Figure 24 Scatter Plot Comparison 21
Figure 25 Loan Approval/Rejection By occupation Type 22
Figure 26 Loan Acceptance and Rejection 23
Figure 27 Avg.Credit Amount vs Client Type........ 24
Figure 28 Analysing Category and Numerical columns Based on Target........ 25
Multivariate Analysis........ 25
Figure 29 Correlation Heatmap of All Variables 26
Figure 30 Loan Approval Rate - Hosuing Type vs Family Status 27
Figure 31 Correlation Heatmap: Contract Type Vs Income_type 28
Figure 32 Correlation Heat map- Amount Category Columns Based on Target 29
Figure 33 Co-relation Heatmap_ Housing Type vs Own Reality 30
Conclusion........ 30

Introduction:

The purpose of this case study is to identify patterns that indicate if a client may have difficulty paying their instalments. These patterns can be used to take actions such as denying the loan, reducing the amount of the loan, or lending to risky applicants at a higher interest rate. The goal is to ensure that only those capable of repaying the loan are approved. This case study aims to identify such applicants using EDA. In simpler terms, the company wants to understand the variables that drive loan defaults. These are the strong indicators of loan default.

With this knowledge, the company can better assess risk and manage its portfolio. To better understand the domain, it is recommended to research risk analytics independently. This will help in understanding the types of variables and their significance.

Problem Statement:

"In this comprehensive case study, the exploration delves into the intricate landscape of financial data using Python on Jupyter Notebook. Rigorous techniques such as Univariate Analysis, Bivariate Analysis, and Multivariate Analysis, aim to identify crucial patterns that indicate a client's likelihood of facing difficulties in paying their instalments.

Furthermore, the correlation analysis applied to the datasets enhances understanding of the intricate relationships between various factors. This meticulous approach not only unravels the complexities of client payment behaviour but also provides a foundation for informed decision-making in banking.

The ultimate goal is to employ these insights for effective risk mitigation, ensuring that loan applicants with the capability to repay are not overlooked, while simultaneously addressing potential risks through tailored lending strategies."

Data Overview:

- Previous_application.csv: This Excel file provides data regarding the client's past loan transactions.
- Application_data.csv: This Excel document provides detailed information about the client at the point of their loan application.

Analysis of Decisions in the Loan Application Process:

- Approved: Company approval of the loan application.
- Cancelled: Client withdrawal, possibly due to changing circumstances or unfavourable terms.
- Refused: Loan denial based on factors like non-compliance with requirements or risk assessment.
- Unused Offer: Client-initiated cancellation at different stages.

Analysing these decisions defines the understanding of client-company interactions, shedding light on the nuances of the loan approval journey."

Business Objectives

- The objective is to identify patterns indicating whether a client will have difficulty paying their loan instalments.
- This information can be used to take actions such as reducing the loan amount, denying the loan or lending to risky applicants at a higher interest rate.
- Additionally, it is important to identify applicants who are capable of repaying the loan, so they are not unfairly rejected.

Importing Libraries

```
## Importing Libraries
```

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

Figure 1 Importing Library

In this analysis, I employed several Python libraries to manipulate and visualize the data. The key libraries utilized include:

1. NumPy: A powerful library for numerical operations, facilitating efficient array handling.
2. Pandas: A versatile data manipulation library, essential for cleaning, organizing, and structuring the dataset.
3. Matplotlib: A comprehensive plotting library for creating static, interactive, and animated visualizations in Python.
4. Seaborn: Built on top of Matplotlib, Seaborn provides an aesthetically pleasing and high-level interface for drawing attractive and informative statistical graphics.

Data Importing:

Load the datasets as shown below, I have printed the first 5 rows of both datasets to understand the data view and later I can use them for data cleaning

```
## Reading the Data
```

```python
# Load datasets
previous_application = pd.read_csv('previous_application.csv')
application_data = pd.read_csv('application_data.csv')

# Display the first few rows of each dataset
print("Previous Application Data:")
print(previous_application.head())

print("\nApplication Data:")
print(application_data.head())
```

Figure 2 Reading the data

Data Cleaning:

After importing the library, I loaded both datasets for the data cleaning the following steps were taken as shown below:

```
## Dropping Unnecessary Columns

# Drop unnecessary columns in previous applications dataset
unnecessary_cols_previous = [
    'SK_ID_PREV', 'FLAG_LAST_APPL_PER_CONTRACT', 'NFLAG_LAST_APPL_IN_DAY', 'SELLERPLACE_AREA'
]
previous_application = previous_application.drop(unnecessary_cols_previous, axis=1)

# Drop unnecessary columns in application data dataset
unnecessary_cols_app = [
    'FLAG_MOBIL', 'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE', 'FLAG_EMAIL',
    'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY', 'FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3',
    'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6', 'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8',
    'FLAG_DOCUMENT_9', 'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12', 'FLAG_DOCUMENT_13',
    'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17', 'FLAG_DOCUMENT_18',
    'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21'
]
application_data = application_data.drop(unnecessary_cols_app, axis=1)

# Display the remaining columns in each dataset
print("Remaining columns in Previous Applications Data:")
print(previous_application.columns)

print("\nRemaining columns in Application Data:")
print(application_data.columns)
```

Figure 3 Dropping Unnecessary Columns

```
## Handling Missing Values in both datasets

# Check for missing values in previous_application dataset
print("Missing values in Previous Applications Data:")
print(previous_application.isnull().sum())

# Check for missing values in application_data dataset
print("\nMissing values in Application Data:")
print(application_data.isnull().sum())
```

Figure 4 Checking and Handling Missing Values in Both Datasets

Detecting Outliers:

```
## Detecting Outliers

# Columns to check for outliers in previous_application data
columns_to_check_previous = ['AMT_ANNUITY', 'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_DOWN_PAYMENT', 'AMT_GOODS_PRICE', 'RATE_DOWN_PA

# Create individual box plots for each column in previous_application data
plt.figure(figsize=(15, 10))

for i, column in enumerate(columns_to_check_previous, 1):
    plt.subplot(2, 4, i)
    sns.boxplot(y=previous_application[column], color='skyblue', width=0.5)
    plt.title(f'Boxplot of {column}')
    plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

plt.tight_layout()
plt.show()
```

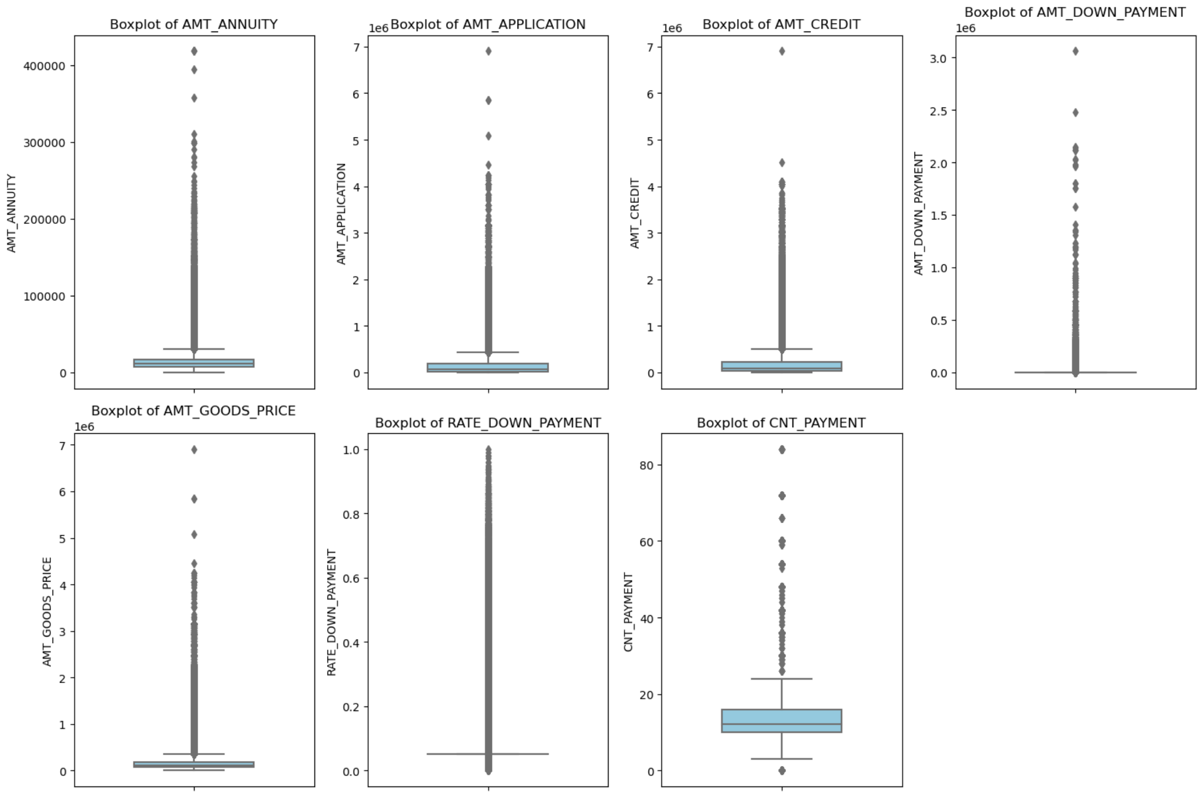

Figure 5 Detecting Outliers

- Wide variation in all four variables (AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT, and AMT_DOWN_PAYMENT).
- Some outliers in the data.
- Median AMT_ANNUITY is around 100,000.
- Median AMT_APPLICATION is around 10,000.
- Median AMT_CREDIT is around 300,000.
- Median AMT_DOWN_PAYMENT is around 100,000.
- Strong correlation between AMT_ANNUITY and AMT_CREDIT.

For the previous_application dataset:

Impute Numerical Columns:

AMT_ANNUITY, AMT_CREDIT, AMT_DOWN_PAYMENT, AMT_GOODS_PRICE, RATE_DOWN_PAYMENT, CNT_PAYMENT: I can consider imputing these numerical columns with their mean or median.

Impute Categorical Columns:

NAME_TYPE_SUITE, PRODUCT_COMBINATION: These are categorical columns. I impute missing values with the mode (most frequent category).

Dropping Columns:

RATE_INTEREST_PRIMARY, RATE_INTEREST_PRIVILEGED: These columns have a significant number of missing values. Depending on their importance to analysis, I might consider dropping them.

Handling Data Columns:

For columns related to days (e.g., DAYS_FIRST_DRAWING, DAYS_FIRST_DUE, DAYS_LAST_DUE_1ST_VERSION, DAYS_LAST_DUE, DAYS_TERMINATION), consider if there's a meaningful way to impute or handle them.

For the application_data dataset:

Impute Numerical Columns:

I consider imputing numerical columns with their mean or median, depending on the distribution.

Impute Categorical Columns:

For categorical columns, impute missing values with the mode.

Dropping Columns or Rows:

If a column has a small percentage of missing values and is not crucial for analysis, I considered dropping the rows with missing values.

Handling 'XNA' and 'XAP' Values:

Replaced 'XNA' and 'XAP' values with NaN where appropriate.

1. Merging Data frames: Merged data from two different data frames (`application_data` and `previous_application_data`) using a common key (`SK_ID_CURR`).
2. Handling Null Values:
   - Checked for missing values in the merged data.
   - Imputed missing values in numerical columns with the median.
   - Imputed missing values in categorical columns with the mode.
3. Handling Specific Values: Replaced specific values ('XNA' and 'XAP') in the 'CODE_GENDER' column with the mode

4. Subset Data for Analysis: Selected specific columns for analysis, including variables like income, credit, annuity, and time-related features.

5. Correlation Analysis: Calculated and visualized the correlation matrix using a heatmap for selected variables.

Handling XNA or XAP values in previous_application

```
# Replace 'XNA' or 'XAP' with NaN ()
previous_application.replace(['XNA', 'XAP'], np.nan, inplace=True)

# Handle missing values dropping columns with more than 30% missing values)
threshold = 0.3
previous_application = previous_application.dropna(thresh=len(previous_application) * (1 - threshold), axis=1)
```

```
# Check 'XNA' or 'XAP' values along with NaN in Previous Application Data
xna_xap_counts = {}
for column in previous_application.columns:
    xna_count = previous_application[column].eq('XNA').sum()
    xap_count = previous_application[column].eq('XAP').sum()
    nan_count = previous_application[column].isna().sum()
    xna_xap_counts[column] = {'XNA': xna_count, 'XAP': xap_count, 'NaN': nan_count}

# Display the counts
print("Counts of 'XNA' or 'XAP' values in Previous Application Data after handling NaN:")
print(xna_xap_counts)
```

Figure 6 Handling XNA and XAP Values

Merging Datasets

- Merged Previous Applications Data and Application Data on 'SK_ID_CURR'.

Merge the data after Data Cleaning

```
# Merge datasets
merged_data = pd.merge(application_data, previous_application, on='SK_ID_CURR', how='inner')

# Display the merged dataset
print("Merged Data:")
# Display the first few rows of the merged dataset in a table
merged_data.head()
```

```
Merged Data:
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CF
0	100002	1	Cash loans	M	N	Y	0	202500.0	4
1	100003	0	Cash loans	F	N	N	0	270000.0	12
2	100003	0	Cash loans	F	N	N	0	270000.0	12
3	100003	0	Cash loans	F	N	N	0	270000.0	12
4	100004	0	Revolving loans	M	Y	Y	0	67500.0	1

5 rows × 117 columns

Figure 7 Merged Data

```
# Step 1: Identify null values
null_values = merged_data.isnull().sum()

# Display the null values
print("Null values in the merged dataset:")
print(null_values)

# Step 2: Handling null value

# Example: Filling null values with 0
merged_data = merged_data.fillna(0)

# Display the updated DataFrame
print("\nMerged dataset after handling null values:")
print(merged_data.head())
```

```
Null values in the merged dataset:
SK_ID_CURR                 0
TARGET                     0
NAME_CONTRACT_TYPE_x       0
CODE_GENDER                0
FLAG_OWN_CAR               0
                          ..
DAYS_FIRST_DUE             0
DAYS_LAST_DUE_1ST_VERSION  0
DAYS_LAST_DUE              0
DAYS_TERMINATION           0
NFLAG_INSURED_ON_APPROVAL  0
Length: 117, dtype: int64
```

Figure 8 Handling Remaining Null Values after Merging the Data

Conversion of Negative Values in date type columns:

```
# Identify numeric columns
numeric_columns = merged_data.select_dtypes(include=['number']).columns

# Convert negative values to absolute values in numeric columns
merged_data[numeric_columns] = merged_data[numeric_columns].abs()

# Display the first few rows of the updated merged dataset
print("Updated Merged Data with Absolute Values:")
print(merged_data.head())
```

Figure 9 Converting Negative Values into Absolute Values

After I performed Data Type Handling where I Checked and handled non-numeric columns.

Modified merged_data

In [22]:
```
print("Modified Merged Data with Absolute Values:")
merged_data.head()
```

```
Modified Merged Data with Absolute Values:
```

Out[22]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CF
0	100002	1	Cash loans	M	N	Y	0	202500.0	4
1	100003	0	Cash loans	F	N	N	0	270000.0	12
2	100003	0	Cash loans	F	N	N	0	270000.0	12
3	100003	0	Cash loans	F	N	N	0	270000.0	12
4	100004	0	Revolving loans	M	Y	Y	0	67500.0	1

5 rows × 117 columns

Figure 10 Merged Data After Cleaning

Data Analysis

In [23]:
```
# Display summary statistics for numeric columns
summary_stats = merged_data.describe()

# Display the summary statistics
print("Summary Statistics for Merged Data:")
print(summary_stats)
```

```
Summary Statistics for Merged Data:
         SK_ID_CURR        TARGET  CNT_CHILDREN  AMT_INCOME_TOTAL  \
count  1.413701e+06  1.413701e+06  1.413701e+06      1.413701e+06
mean   2.784813e+05  8.655296e-02  4.048933e-01      1.733160e+05
std    1.028118e+05  2.811789e-01  7.173454e-01      1.985734e+05
min    1.000020e+05  0.000000e+00  0.000000e+00      2.565000e+04
25%    1.893640e+05  0.000000e+00  0.000000e+00      1.125000e+05
50%    2.789920e+05  0.000000e+00  0.000000e+00      1.575000e+05
75%    3.675560e+05  0.000000e+00  1.000000e+00      2.070000e+05
max    4.562550e+05  1.000000e+00  1.900000e+01      1.170000e+08

         AMT_CREDIT_x  AMT_ANNUITY_x  AMT_GOODS_PRICE_x  \
count  1.413701e+06   1.413701e+06       1.413701e+06
mean   5.875537e+05   2.701688e+04       5.276522e+05
std    3.849173e+05   1.395072e+04       3.531028e+05
min    4.500000e+04   1.615500e+03       4.050000e+04
25%    2.700000e+05   1.682100e+04       2.385000e+05
50%    5.084955e+05   2.492550e+04       4.500000e+05
75%    8.079840e+05   3.454200e+04       6.795000e+05
max    4.050000e+06   2.250000e+05       4.050000e+06
```

Figure 11 Statistical Analysis

Univariate Analysis

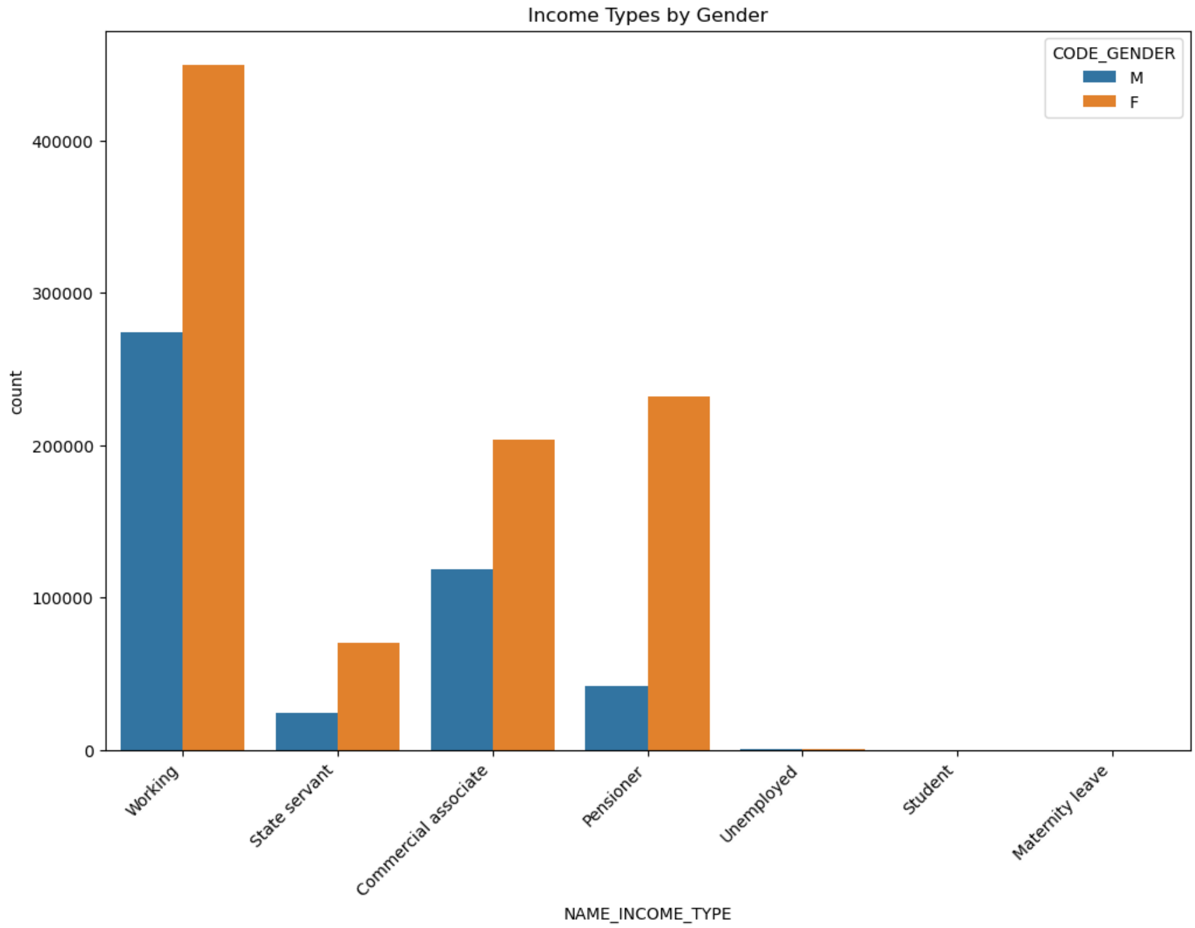

Figure 12 Income Type by Gender

There is a significant difference in income between men and women, with men having a higher income than women in all income types.

The bar chart shows that the average income for men is highest in the "Working" category, followed by the "Commercial associate" and "State servant" categories. The average income for women is highest in the "Working" category, followed by the "State servant" and "Commercial associate" categories.

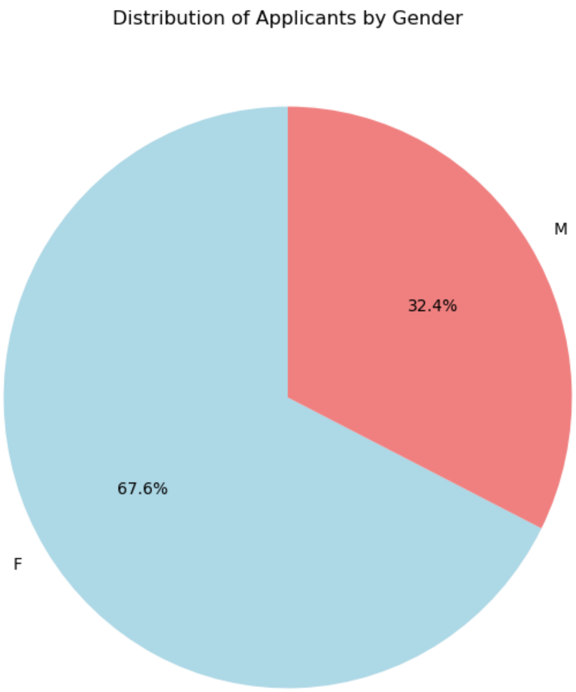

Figure 13 Gender Comparison

Applicant Distribution: Majority (over 60%) have approved contract status, showcasing successful conversion.

Areas for Improvement: Unused offers (around 20%) and canceled contracts (around 10%) highlight opportunities for enhancing conversion rates.

Actionable Insights: Analyzing reasons for declines/cancellations can guide improvements in the banking process and contract offerings.

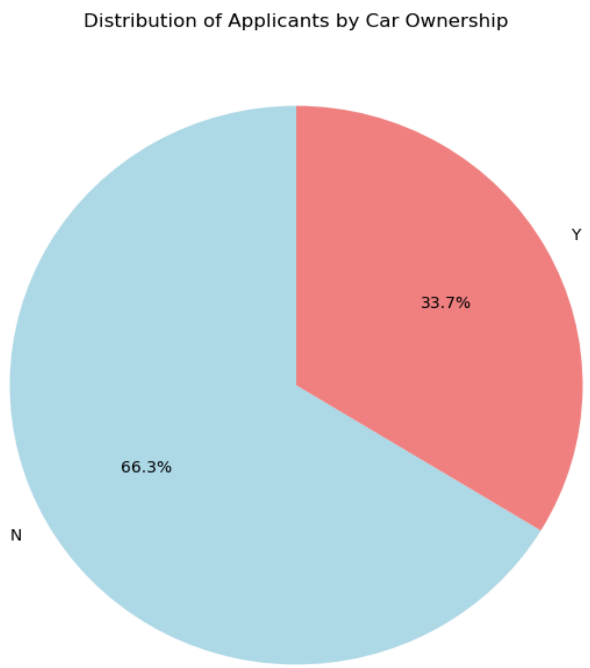

Figure 14 Car Ownership Comparison

Most Clients Don't Have a Car: 66.3% of clients don't own a car, which is a big majority.

Credit Applications: When looking at clients asking for credit, it's important because not having a car might mean fewer financial responsibilities.

Financial Picture: Knowing if someone has a car helps understand their money situation and how it might affect their ability to handle credit.

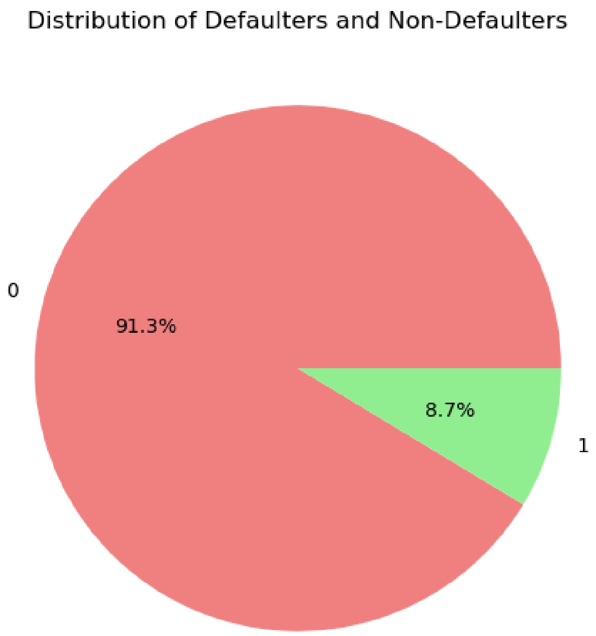

Figure 15 Comparison of Defaulters and Non-defaulters

- Variable TARGET indicates if an applicant has payment difficulties (1) or not (0).
- Applicants with payment difficulties may default on loan repayments.
- Pie charts show that 91.9% are non-defaulters while 8.1% are still Defaulters

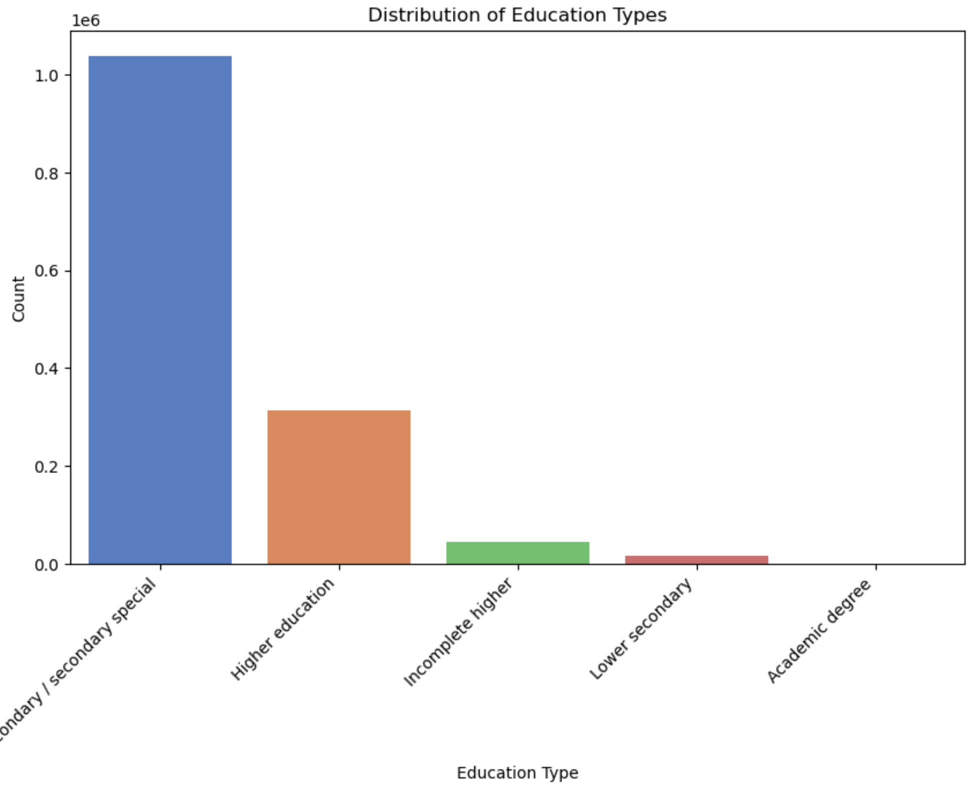

Figure 16 Loan by Education Type

- Data illustrates the distribution of loans based on education.

- Majority of applicants hold at least a secondary college degree.
- Applicants without completed higher education are more likely to face loan rejection.

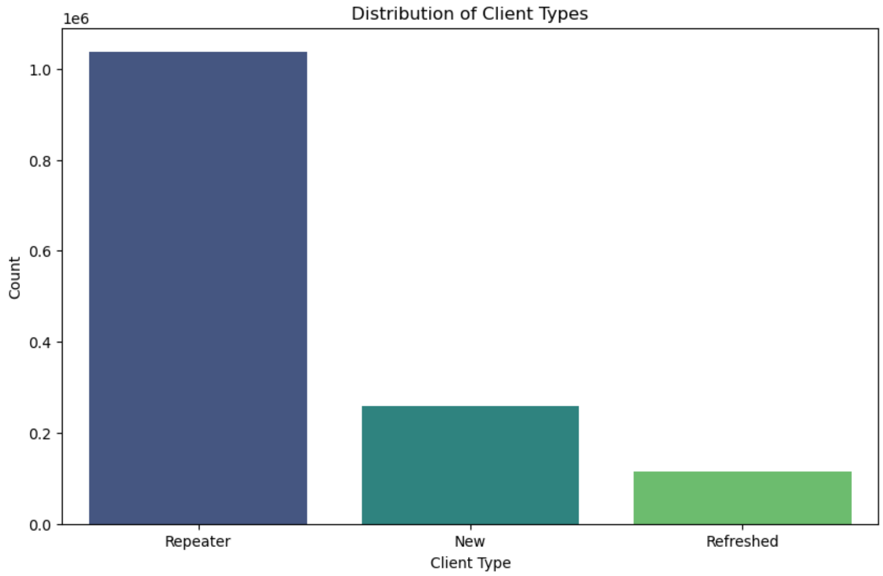

Figure 17 Loan By Client Type

- Graph indicates that a majority of loans are repeat transactions.
- New clients are expressing interest in loans, potentially influenced by word of mouth or digital marketing efforts.

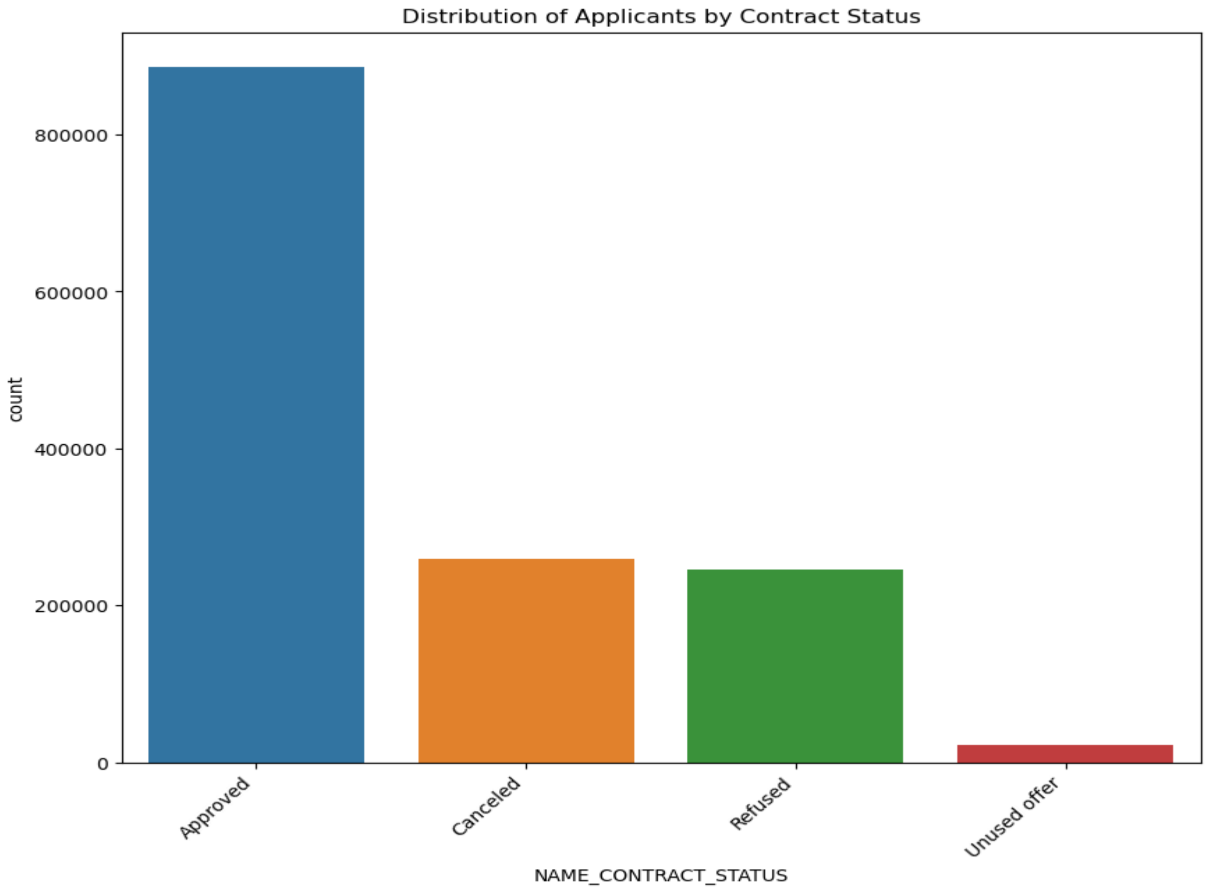

Figure 18 Loan By Contract Status

- Majority (over 60%) have approved contract status.
- Unused offers: Around 20%.
- Canceled contracts: Around 10%.
- Refused contracts: Around 10%.

Implications:

- Success in converting applicants to customers, but room for improvement.
- Opportunities to enhance conversion rates by analyzing declines and cancellations.
- Identify reasons for contract refusals and cancellations.

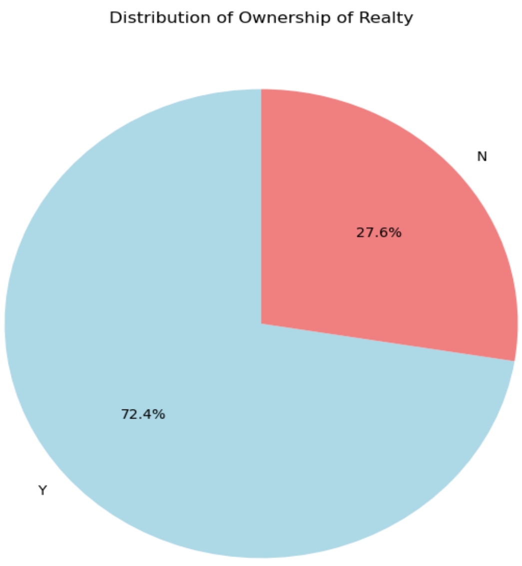

Figure 19 Loan applicant by Property

This shows how many applicants have property. It is estimate that 72.4% applicants have property

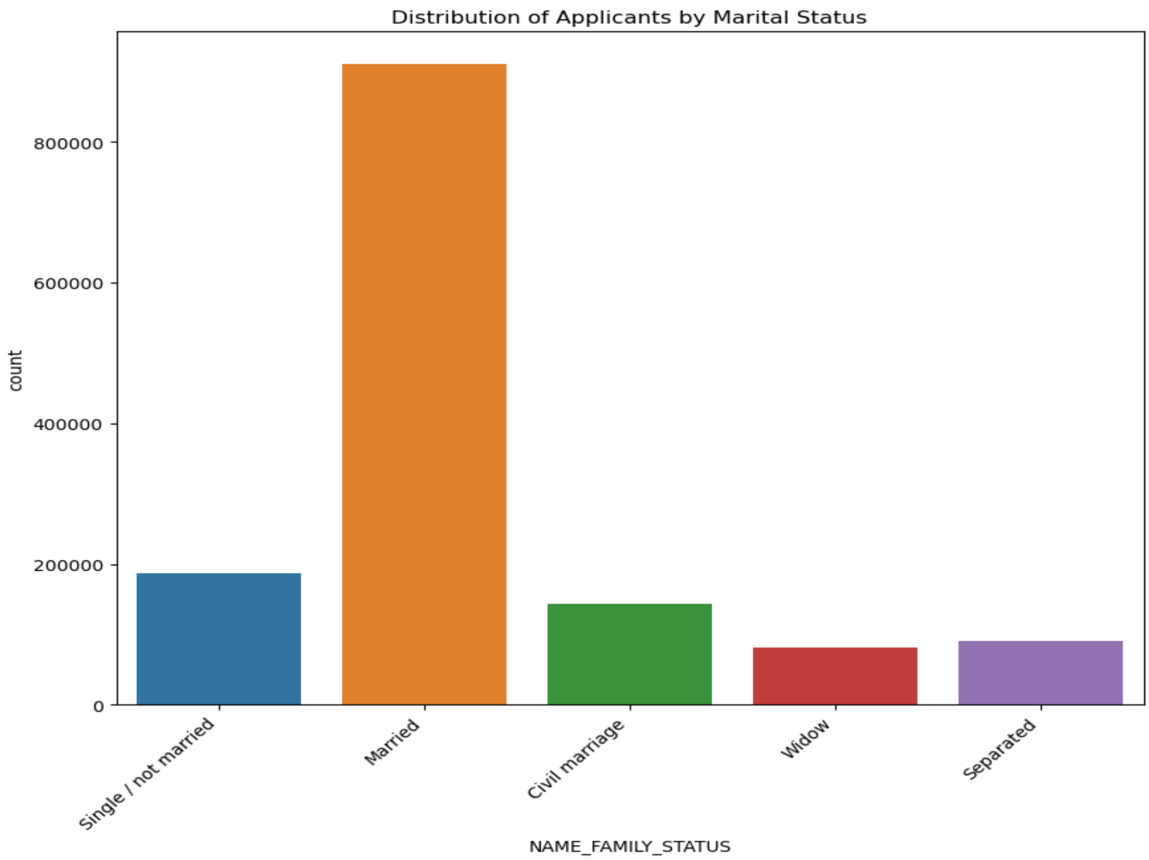

Figure 20 Loan by the distribution of applicants by marital status

- Predominantly, loan applicants are married.

- Noteworthy, there are applicants who are single, in civil marriages, or widowed applying for loans.

Bivariate Analysis

- During the analysis, it was observed that clients with a working status had the highest number of approved loans, while students, unemployed individuals, and businessmen had the lowest. Additionally, clients who owned vehicles were given the highest cash loans. Furthermore, clients who owned real estate had no problem with their loan payments.

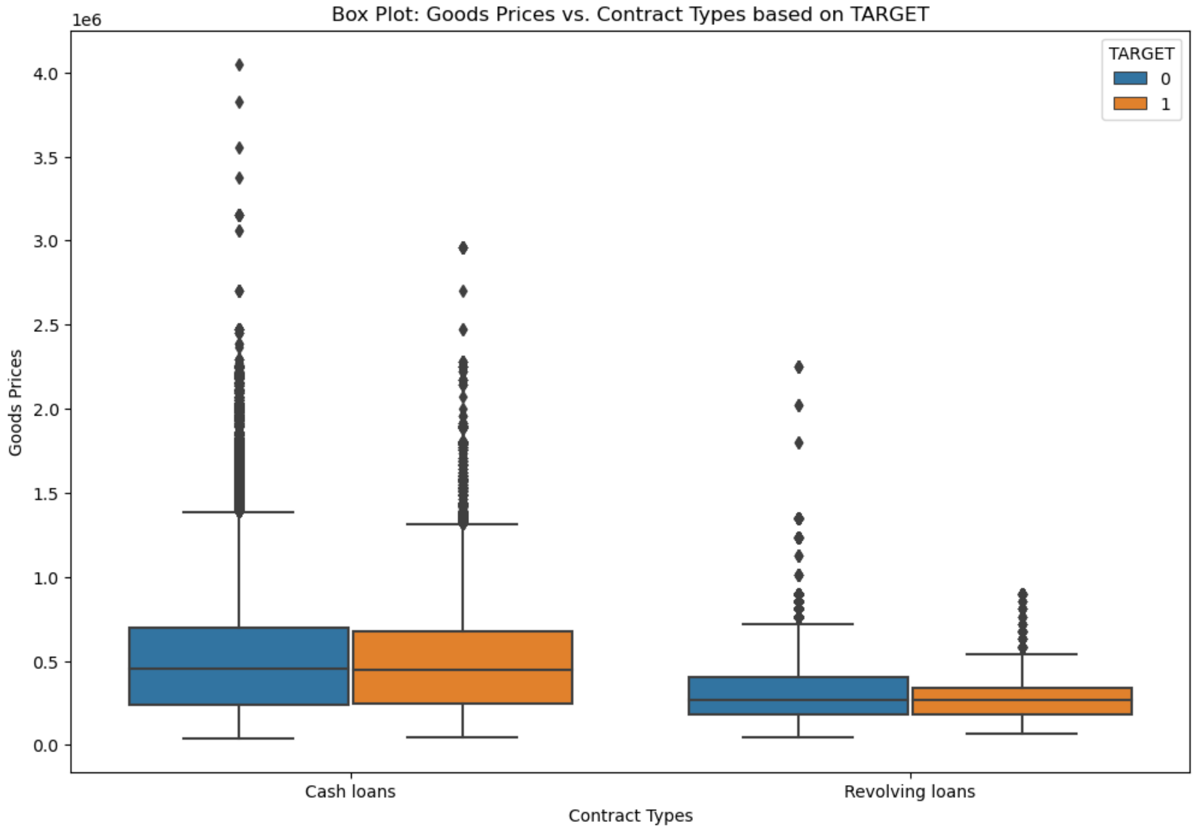

Figure 21 Goods Price vs Contract Types Based on Target

- The box plot provides a clear visualization of the distribution of goods prices for different contract types, with the added insight of loan approval status (TARGET).

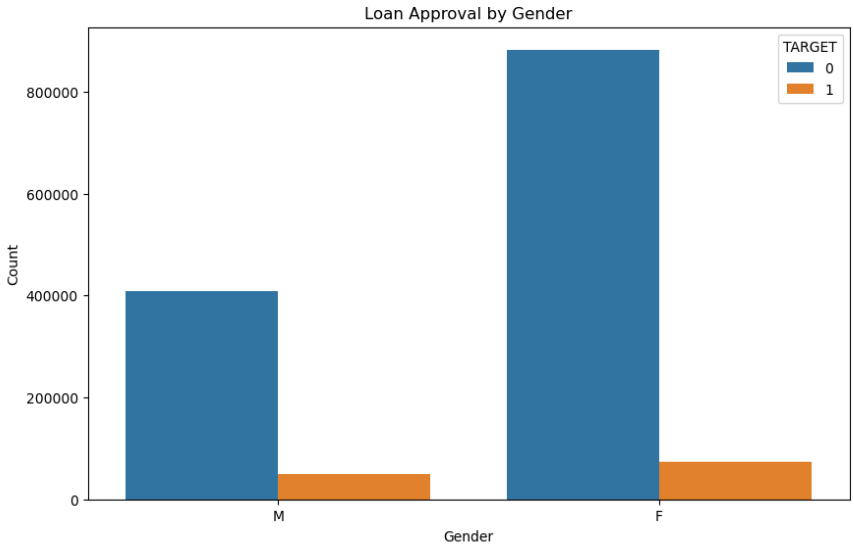

Figure 22 Loan approval by Gender

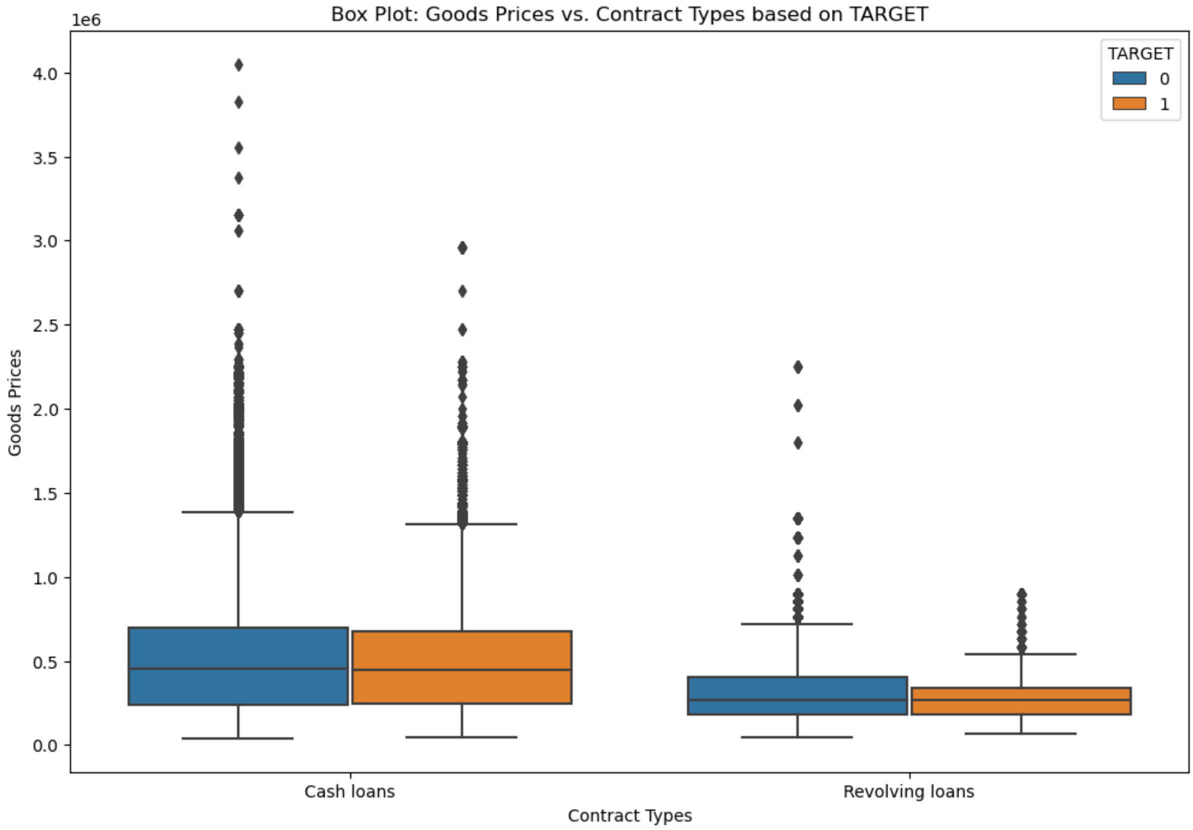

Figure 23 Goods prices Vs Contract Types

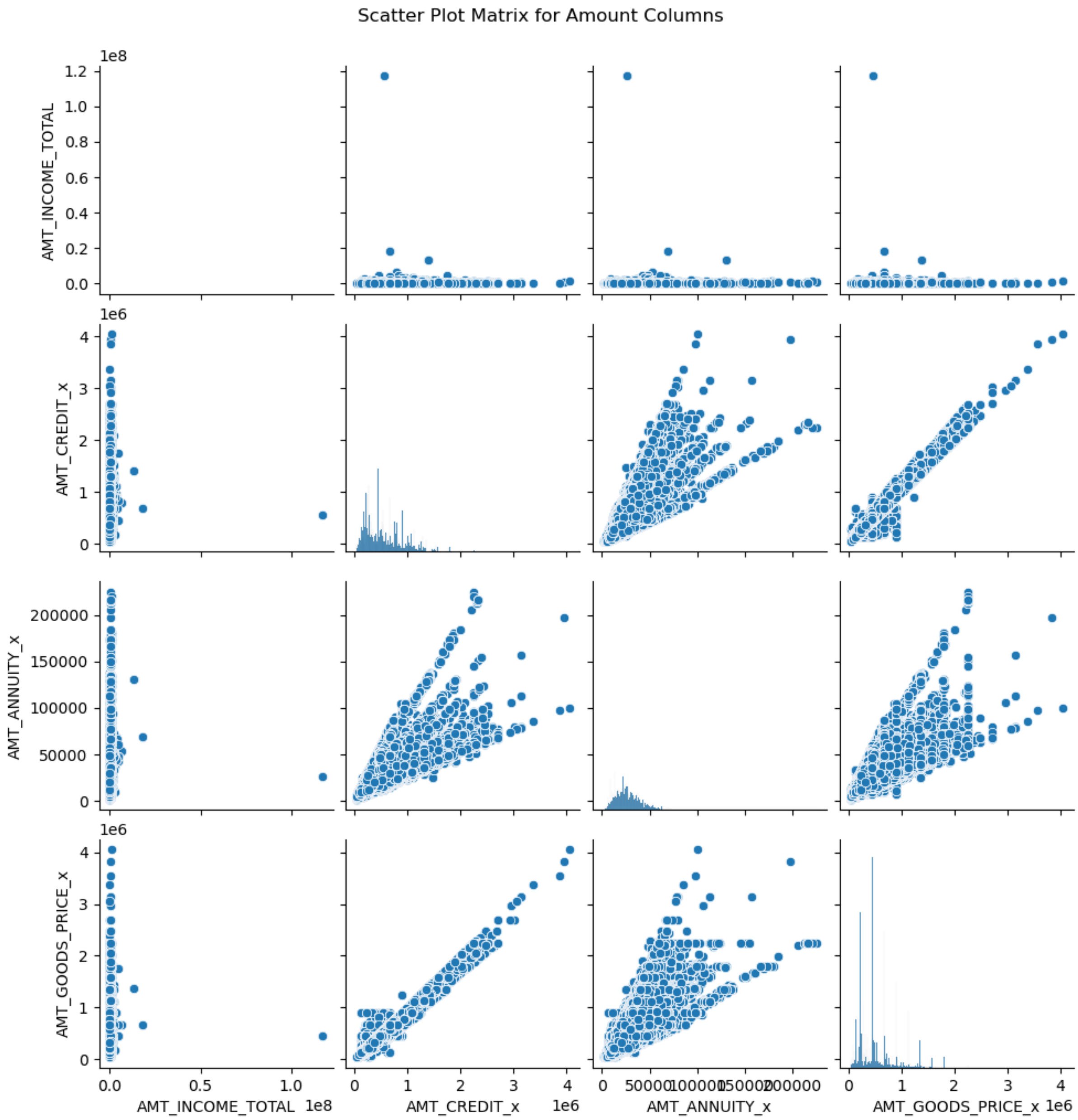

Figure 24 Scatter Plot Comparison

The scatter plot matrix provides a visual representation of the relationships between the specified numeric variables.

- The scatter plot matrix shows the relationship between key numerical variables of Amount Category columns
- Higher total income correlates with larger credit limits.
- Higher income may lead to higher annuity payments and more expensive goods.
- These financial decisions are interconnected, and understanding them helps make better decisions in areas like loan approval and pricing strategies.

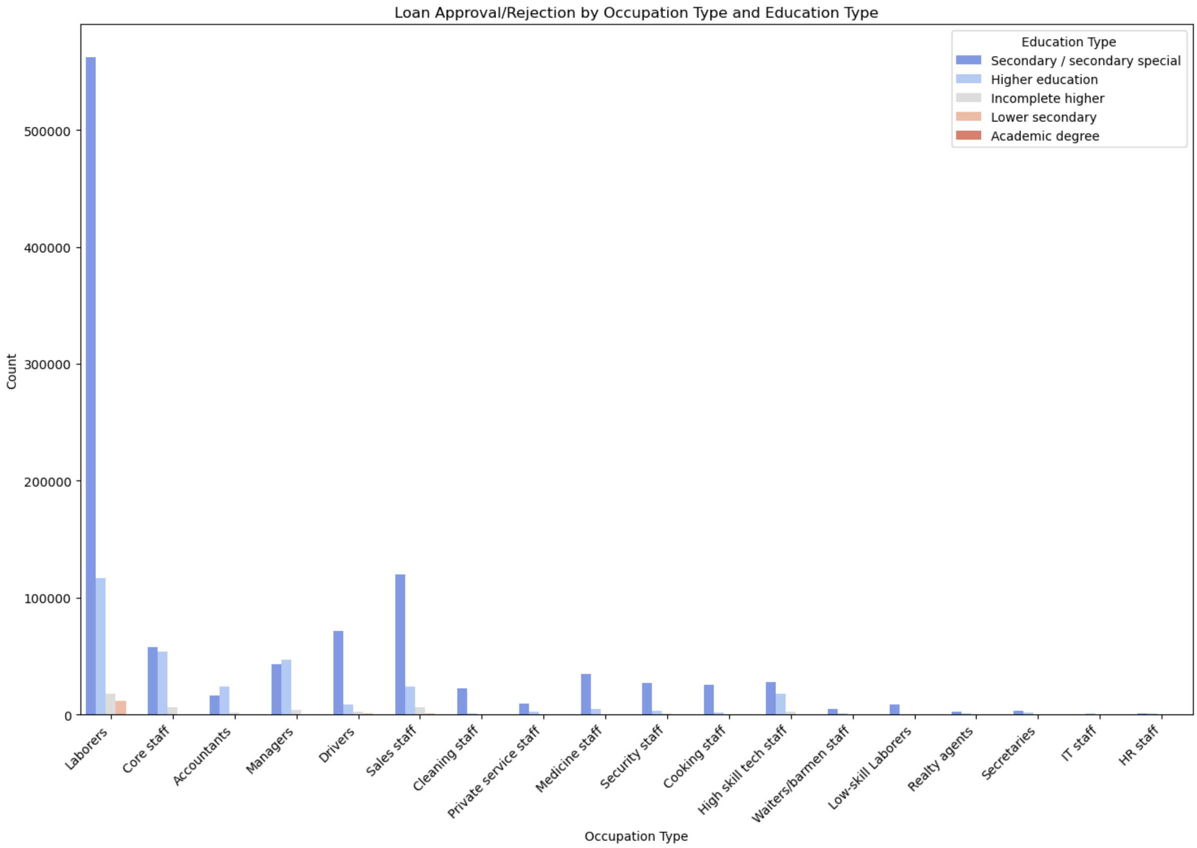

Figure 25 Loan Approval/Rejection By occupation Type

- People with higher education levels and higher-skilled occupations are more likely to have their loans approved.
- This is likely because lenders perceive these borrowers as being more likely to be able to repay their loans.

However, it is important to note that there are other factors, such as credit score and debt-to-income ratio, that also influence loan approval rates

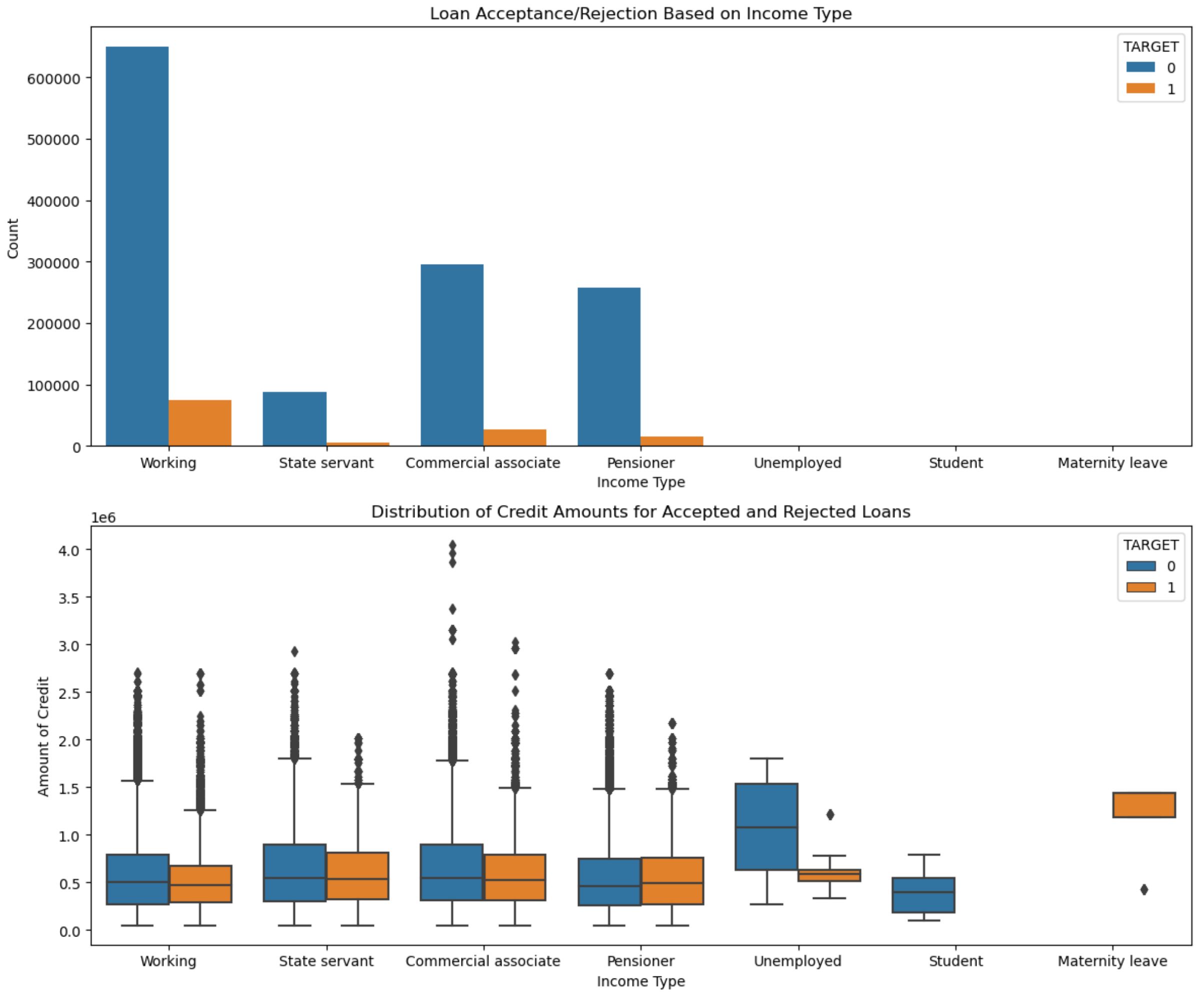

Figure 26 Loan Acceptance and Rejection

1. correlation between income and credit: Higher incomes lead to higher chances of getting approved for loans and receiving larger credit amounts. This suggests a positive association between income and creditworthiness.
2. On the other hand, people with lower incomes typically face higher loan rejection rates, which may indicate that lenders perceive them as a financial risk.
3. Moreover, the distribution of accepted loans among higher-income earners has a right-skewed pattern, indicating that larger loans tend to be concentrated among a smaller group of people.
4. Lenders exercise caution when dealing with lower-income applicants due to concerns about their ability to repay the loan. In contrast, pensioners tend to borrow more, possibly for retirement or unforeseen expenses.
5. Lastly, unemployed and student applicants have the lowest acceptance rates, while there are opportunities for lenders to refine their practices and offer tailored approaches to meet the diverse needs of borrowers.

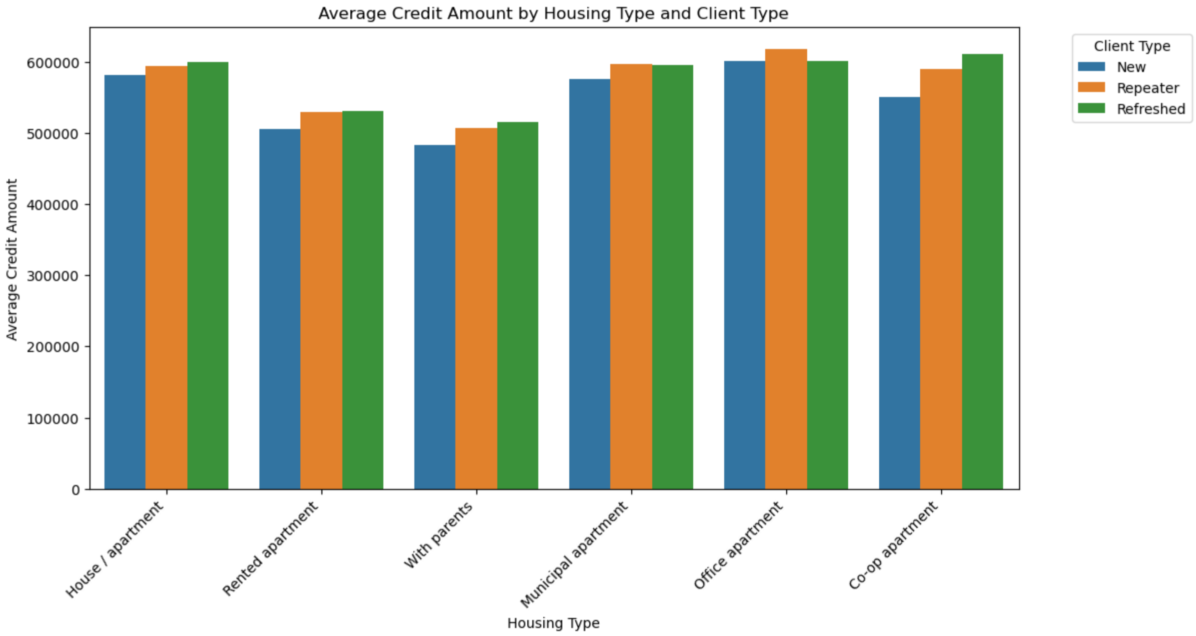

Figure 27 Avg.Credit Amount vs Client Type

- People who own single-family homes have the highest average credit amount.
- There is a large variation in credit amount within each housing type.
- New and refreshed clients have higher average credit amounts than repeater clients.
- There is a positive correlation between homeownership and credit amount.
- Credit amount is also influenced by other factors, such as income, age, and credit history.

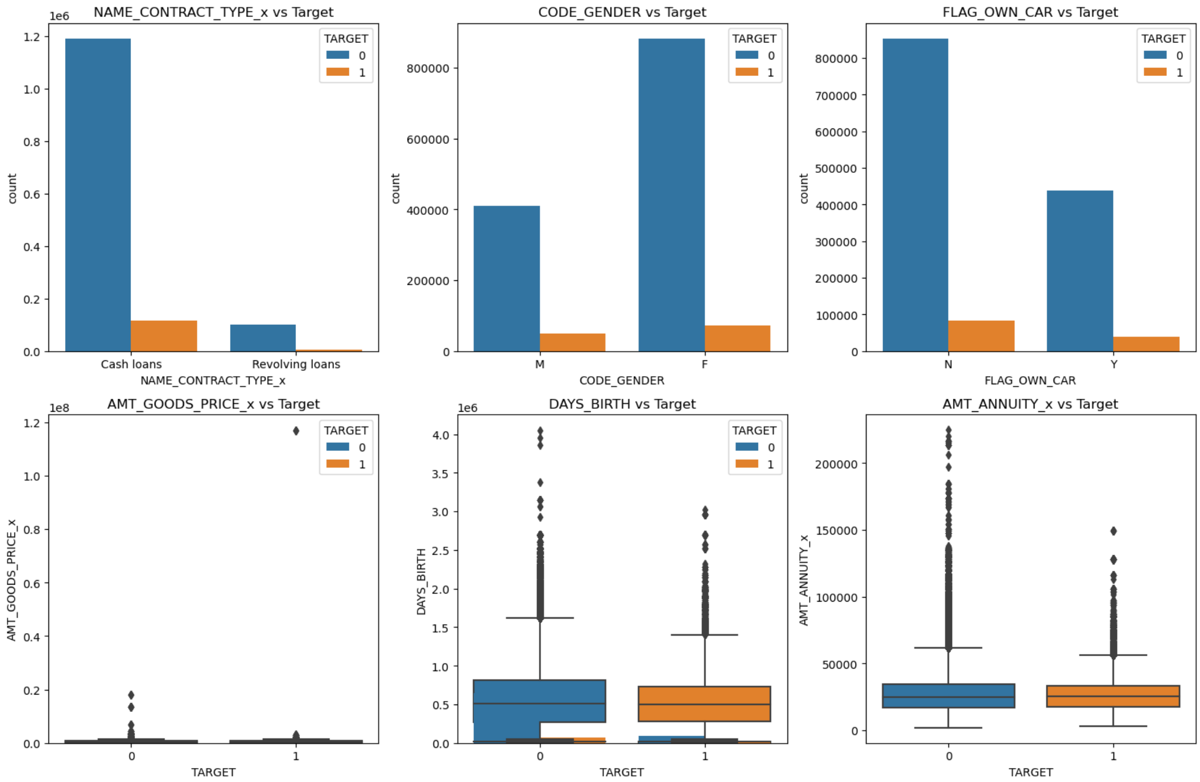

Figure 28 Analysing Category and Numerical columns Based on Target

- the Categorical columns and numerical columns are analyzed Based on Target
- NAME_CONTRACT_TYPE: The target variable is higher for cash loans than for revolving loans.
- CODE_GENDER: The target variable is higher for males than for females.
- FLAG_OWN_CAR: The target variable is higher for people who own a car than for people who do not.

Multivariate Analysis

This helps to compare multiple variables at a time

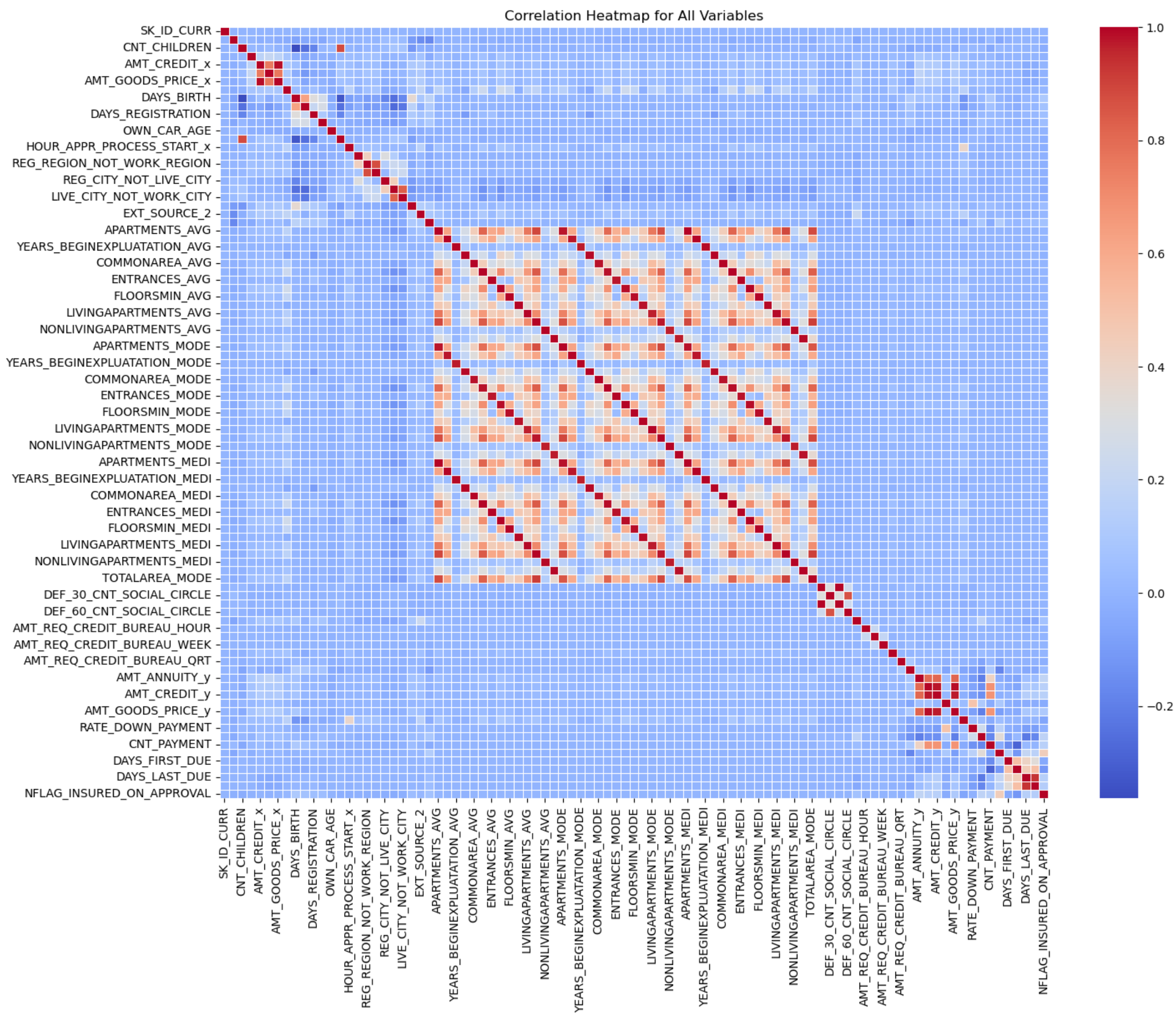

Figure 29 Correlation Heatmap of All Variables

- The correlation heatmap shows the correlation between all variables in the dataset.
- The correlation coefficient is a measure of the strength and direction of the linear relationship between two variables. It can range from -1 to 1, with a value of 1 indicating a perfect positive correlation, a value of -1 indicating a perfect negative correlation, and a value of 0 indicating no correlation.
- The heatmap shows that there are a number of strong correlations between the variables, both positive and negative. For example, there is a strong positive correlation between the variables AMT_CREDIT_Y and AMT_GOODS_PRICE_Y, indicating that borrowers who borrow more money tend to buy more expensive goods. There is also a strong negative correlation between the variables CNT_PAYMENT and

NFLAG_INSURED_ON_APPROVAL, indicating that borrowers who make more payments are less likely to have their loans insured.

- The heatmap also shows that some of the variables are highly correlated with each other. For example, the variables NONLIVINGAPARTMENTS_MODE and APARTMENTS_MODE are almost perfectly correlated, indicating that they are essentially the same variable.

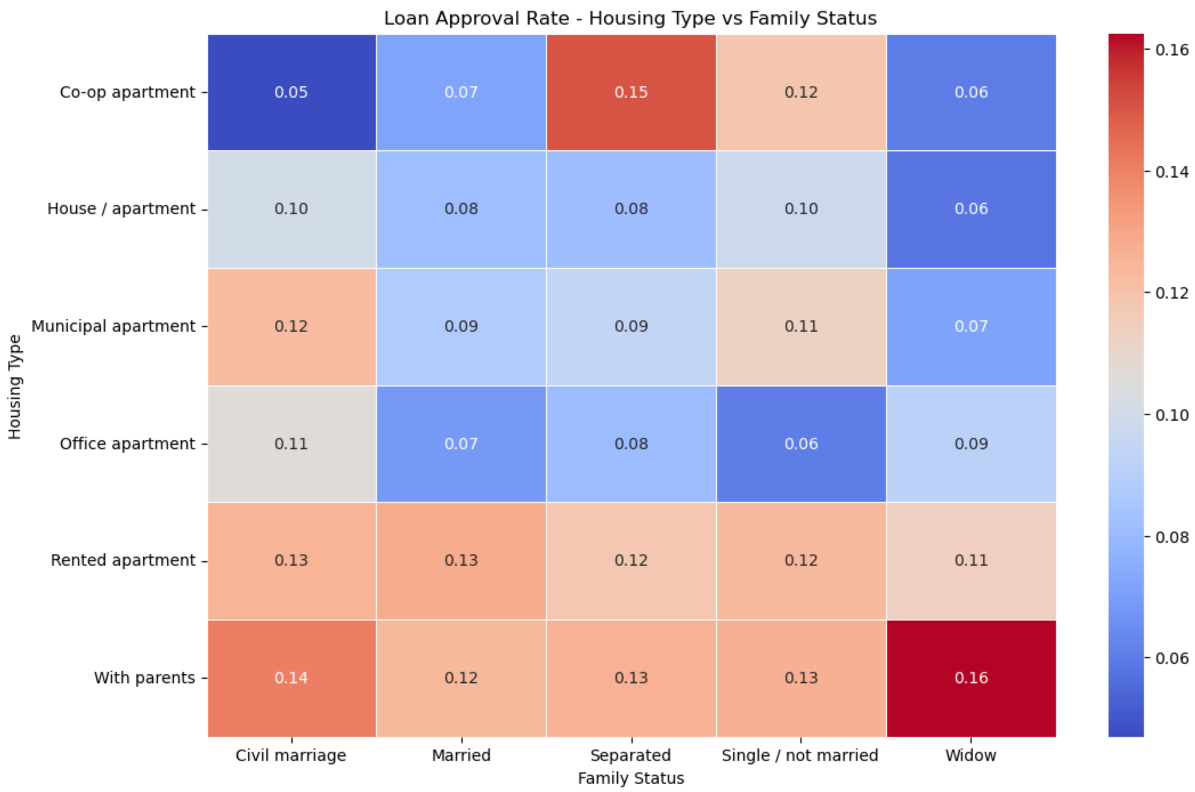

Figure 30 Loan Approval Rate - Hosuing Type vs Family Status

Observations:

- Single/not married borrowers have the lowest loan approval rates for all housing types, except for co-op apartments.
- Married borrowers have the highest loan approval rates for all housing types, except for rented apartments.

- Separated borrowers have the second highest loan approval rates for all housing types, except for co-op apartments and rented apartments.

- Borrowers who live with their parents have the highest loan approval rates for rented apartments.

To conclude, the heatmap shows that there is a relationship between loan approval rate, housing type, and family status. Married borrowers and borrowers who live with their parents have the highest loan approval rates, while single/not married borrowers have the lowest loan approval rates.

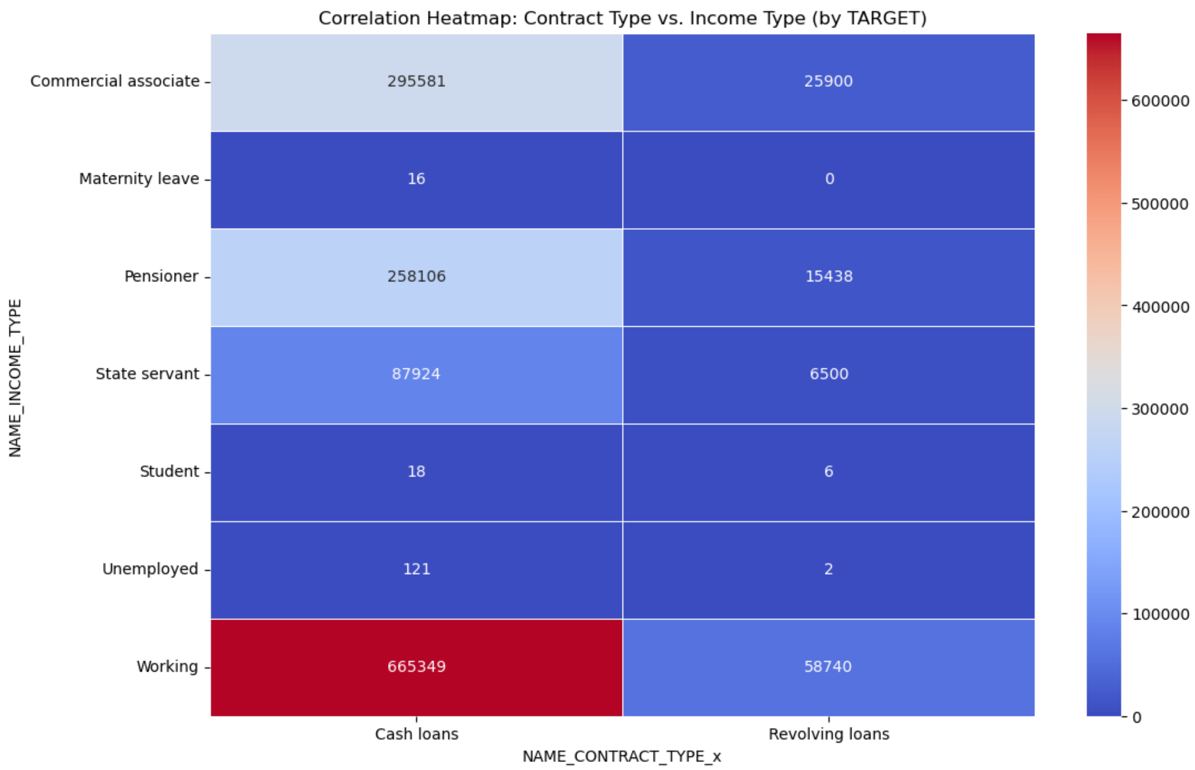

Figure 31 Correlation Heatmap: Contract Type Vs Income_type

- People who are commercial associates are more likely to have revolving loans.
- Pensioners are less likely to have cash loans.
- Commercial associates and state servants are essentially the same contract type.
- Unemployed people are more likely to have pensioner contracts.
- Students are more likely to have cash loans.

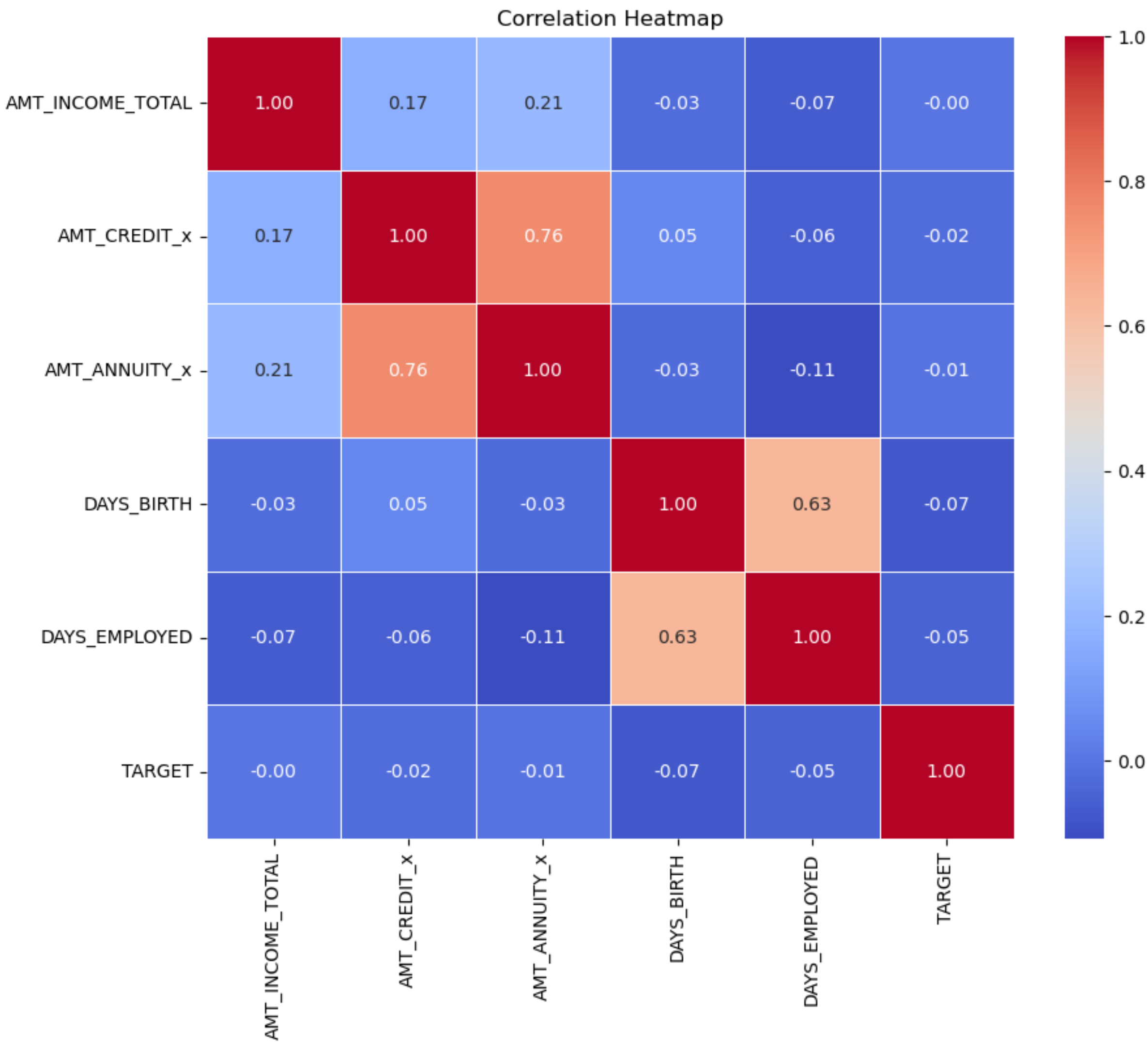

Figure 32 Correlation Heat map- Amount Category Columns Based on Target

- The correlation heatmap shows that there is a negative correlation between income and anxiety. This means that as income increases, anxiety tends to decrease. The correlation coefficient is -0.8, which is considered strong.
- There is also a positive correlation between credit and anxiety. This means that as credit increases, anxiety tends to increase. The correlation coefficient is 0.76, which is also considered strong.
- The other correlations are relatively weak. However, there is a small negative correlation between days employed and anxiety. This means that as the number of days employed increases, anxiety tends to decrease

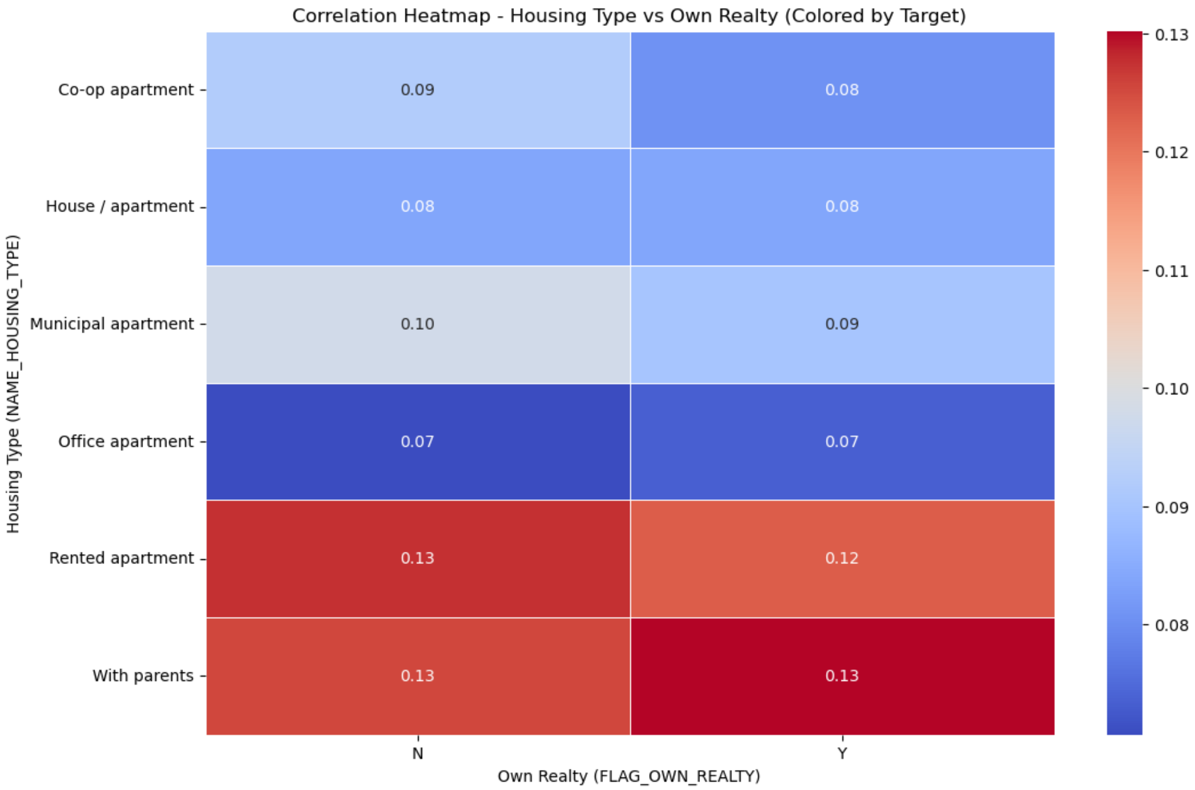

Figure 33 Co-relation Heatmap_ Housing Type vs Own Reality

It shows that client who owns properties more likely to have house/apartments in similar range and havedifficulties in paying the loan

Conclusion

Target Demographics:

- Businessmen and Students: Identified as more profitable segments for targeted banking services.

- Females: Indicated as potentially better clients, suggesting a focus on catering to female demographics.

Loan Portfolio Strategy:

- Mobiles and Electronics: Emphasizes a strategic focus on credit offerings for these products.

- Cash Loans: Recognized as the most popular loan type, suggesting potential for tailored products.

Asset Ownership and Marital Status:

- Car Ownership: Highlighted that most clients do not own cars, informing potential financing options.

- Marital Status: The majority of clients being married indicates potential for family-centric financial products or services.