Fraud Transaction Analysis Using Logistic Regression

Table of Contents

INTRODUCTION

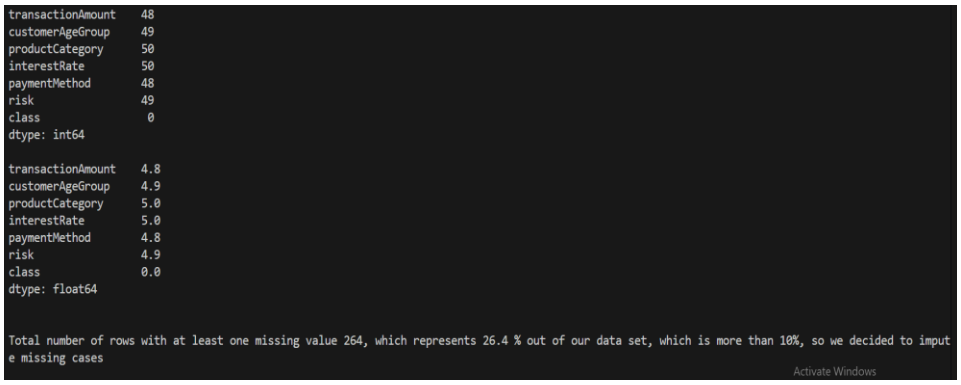
Fraud detection is a critical application of machine learning, particularly in the financial sector, where identifying fraudulent transactions can save significant resources and enhance security. This project aims to develop a machine learning model to classify transactions as fraudulent or legitimate using the provided dataset. The process begins with **Exploratory Data Analysis (EDA)** to understand the dataset's structure, examine key features, and identify patterns or anomalies that distinguish fraudulent transactions from legitimate ones. Next, the **Data Pre-processing** phase involves cleaning and transforming the data to ensure suitability for modelling, including handling missing values, scaling features, and encoding categorical variables. Following this, a **classification algorithm** will be selected and applied to build a predictive model, with a focus on addressing the class imbalance often present in fraud detection scenarios. Finally, the **model will be evaluated** using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to assess its effectiveness in identifying fraudulent transactions while minimizing false positives and negatives. This project demonstrates the power of machine learning in solving real-world challenges, emphasizing data-driven insights and optimized model performance for fraud detection.

Exploratory Data Analysis

Understanding the Dataset and Initial Inspection

To gain a comprehensive understanding of the dataset and its features, we begin by examining its structure using the **.info()** method. This step provides essential insights, including column names, data types, and the presence of null values. Additionally, the inspection reveals the dataset consists of seven columns, with six features **(**transactionAmount, customerAgeGroup, productCategory, interestRate, paymentMethod, risk) and one response variable (class). transactionAmount has 952 entries with a mean of 13,465.26, a high standard deviation of 99,185.77, and a maximum value of 993,548.68, indicating possible outliers. customerAgeGroup (951 entries) has 5 categories, with "Young Adult" as the most frequent, appearing 445 times. productCategory (950 entries) contains 5 categories, with "Electronics" as the most common (382 times). interestRate has 950 entries, a mean of 11.91, and values ranging from 0.0057 to 23.99. paymentMethod has no missing values (952 entries) with 4 unique categories, "Credit Card" being the most frequent (474 times). risk (951 entries) has a mean of 0.477, ranging from 0.0115 to 4.81. The response variable class has 1,000 entries, with approximately 30% of the transactions being fraudulent (class = 1). Missing values exist in customerAgeGroup, productCategory, interestRate, and risk, while features like transactionAmount and risk show significant variability.

Checking the null values

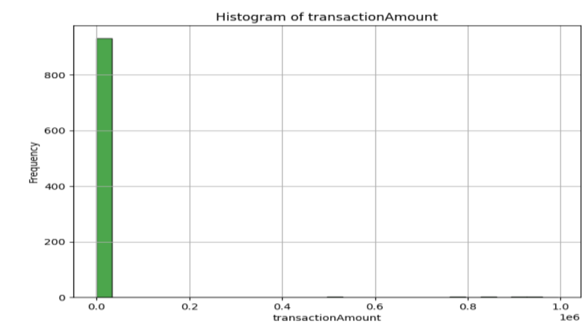


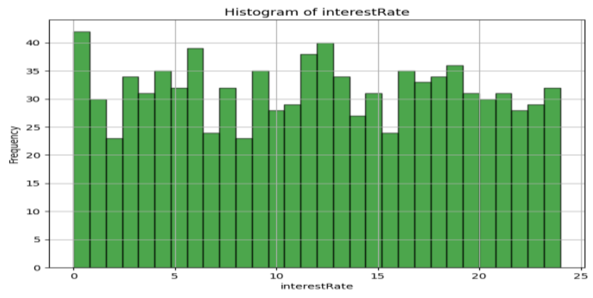
```
transactionAmount    48
customerAgeGroup     49
productCategory      50
interestRate         50
paymentMethod        48
risk                 49
class                 0
dtype: int64

transactionAmount    4.8
customerAgeGroup     4.9
productCategory      5.0
interestRate         5.0
paymentMethod        4.8
risk                 4.9
class                0.0
dtype: float64


Total number of rows with at least one missing value 264, which represents 26.4 % out of our data set, which is more than 10%, so we decided to imput
e missing cases
```

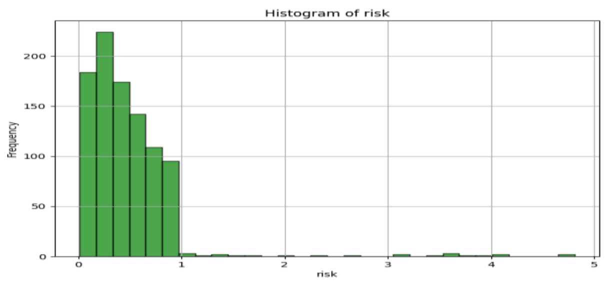
The dataset has missing values in multiple features, affecting 26.4% of rows. Since this exceeds the 5% threshold, imputing missing values is preferred to avoid significant data loss

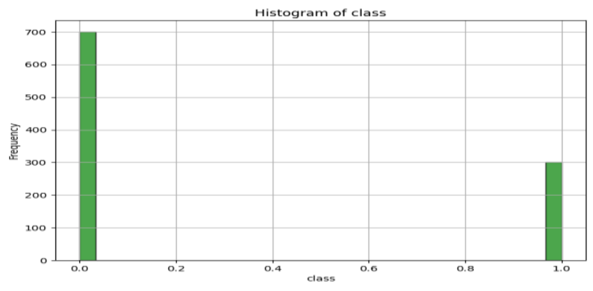
Distribution of Data Set

Numerical variables

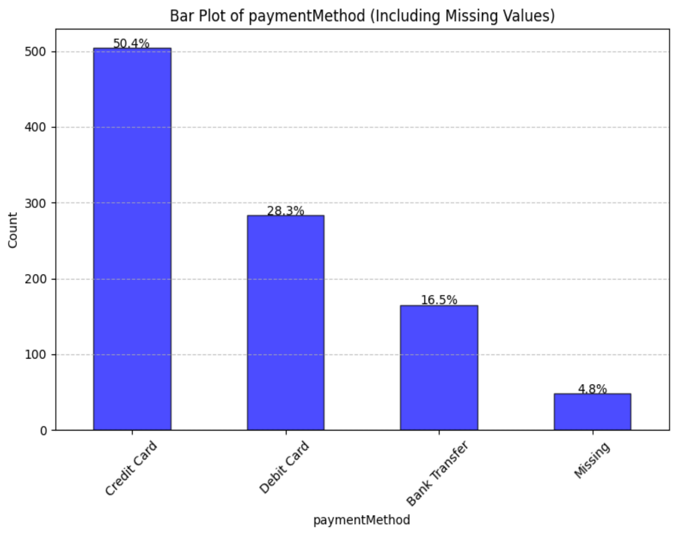


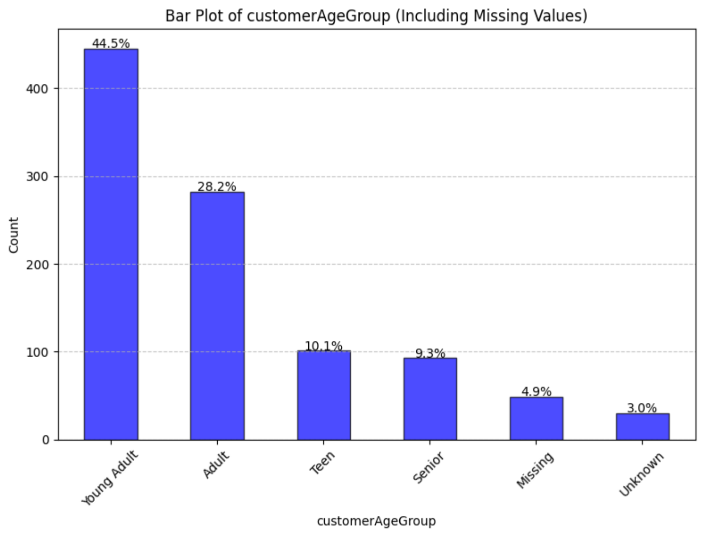


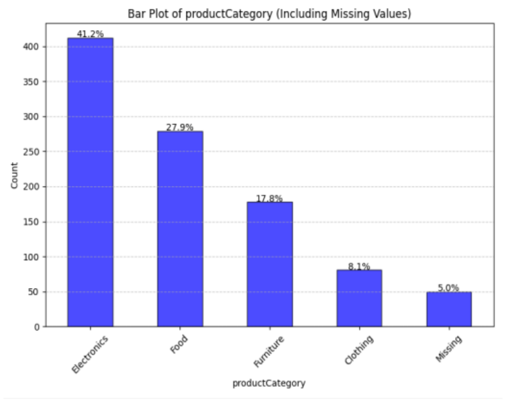




Categorical Variables

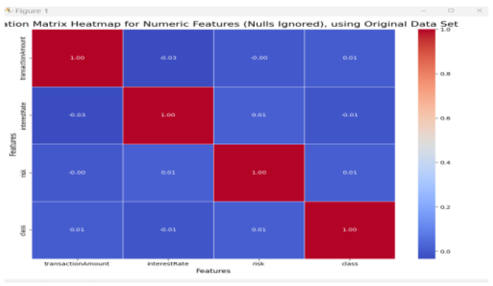






Overall, all the numerical variables are not normally distributed, except for interest rate that is symmetrical, and the targeted variables class shows that the data set is imbalance in favour of 0s. Also, the categorical variables show that "Credit Card" (50.7%), "Young Adult" (44.1%), and "Electronics" (38.2%) are the most common payment method, age group, and product category, respectively, with minor missing data across features. We have to deal with the not normally distributed continuous variables and the imbalanced target variables.

Correlation Matrix of the numerical variable

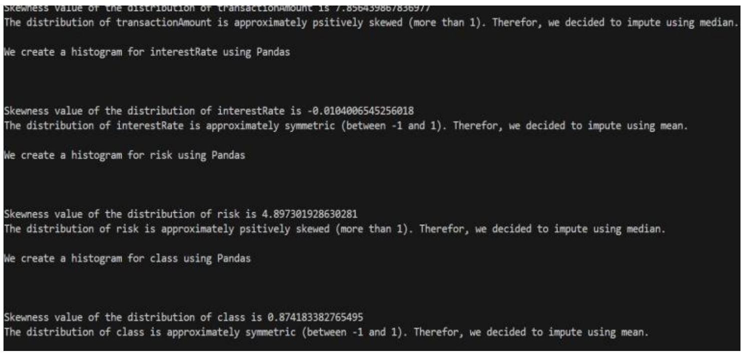


There is no strong correlation between the numerical variable i.e no linear relationship between them

Data Cleaning and Preposing

Handling null value of continuous variable

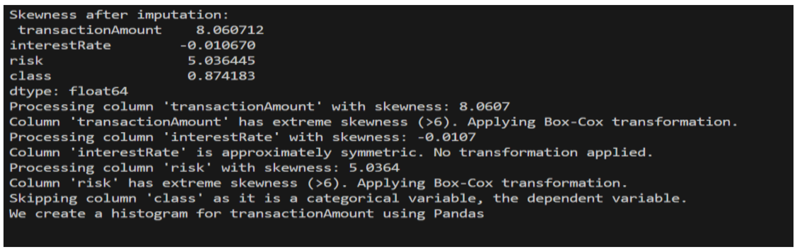
We check skewness of each continuous variable and impute the null value with mean if it is normally distributed otherwise, we will use the median. Although we know this already from the distribution but we are doing for easy imputation using IF function. The output has been shown below



```
Skewness value of the distribution of transactionAmount is 7.856439867836977
The distribution of transactionAmount is approximately psitively skewed (more than 1). Therefor, we decided to impute using median.

We create a histogram for interestRate using Pandas



Skewness value of the distribution of interestRate is -0.0104006545256018
The distribution of interestRate is approximately symmetric (between -1 and 1). Therefor, we decided to impute using mean.

We create a histogram for risk using Pandas



Skewness value of the distribution of risk is 4.897301928630281
The distribution of risk is approximately psitively skewed (more than 1). Therefor, we decided to impute using median.

We create a histogram for class using Pandas



Skewness value of the distribution of class is 0.874183382765495
The distribution of class is approximately symmetric (between -1 and 1). Therefor, we decided to impute using mean.
```

Handling the distributed of the Continuous variable

We check the skewness of the continuous variable after the imputation. Since the variable that is not normally distributed and positively skewed, we tried the transformation using logarithm transformation and square root, but it did not improve the distribution, so we decided to go with boxcox. We only applied it to “transaction Amount” and “Risk” was transformed since intrestedRate is symmetrical

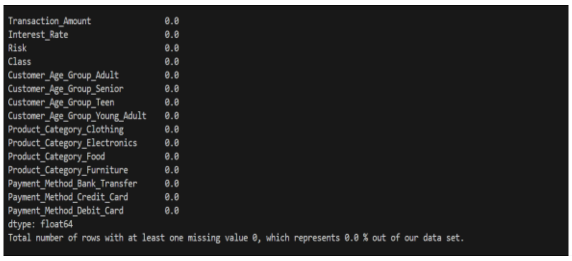


```
Skewness after imputation:
 transactionAmount    8.060712
interestRate        -0.010670
risk                 5.036445
class                0.874183
dtype: float64
Processing column 'transactionAmount' with skewness: 8.0607
Column 'transactionAmount' has extreme skewness (>6). Applying Box-Cox transformation.
Processing column 'interestRate' with skewness: -0.0107
Column 'interestRate' is approximately symmetric. No transformation applied.
Processing column 'risk' with skewness: 5.0364
Column 'risk' has extreme skewness (>6). Applying Box-Cox transformation.
Skipping column 'class' as it is a categorical variable, the dependent variable.
We create a histogram for transactionAmount using Pandas
```

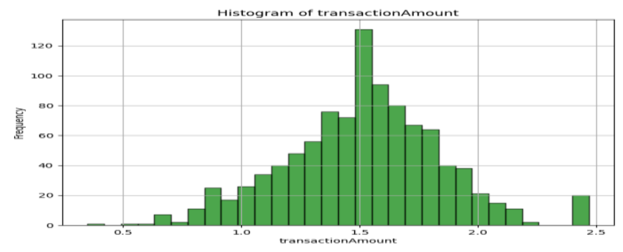
Dealing with null value of Categorical variable

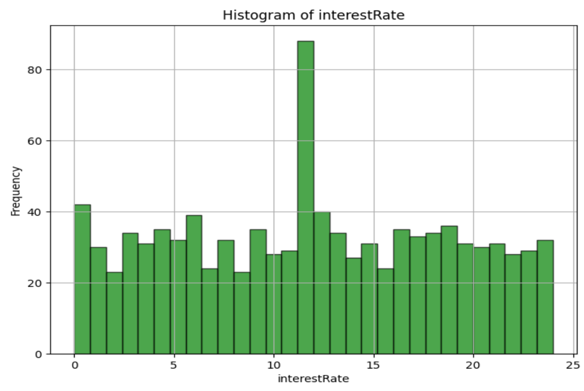
The null value for the categorical variable was filled with mode for both nominal variable (“Payment Method” and “Product Category”) and Ordinal variable “(Customer Age Group”)

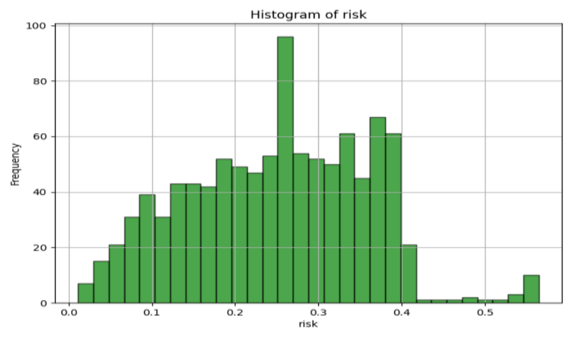
Distribution of data and Null Value after transformation and distribution and cleaning

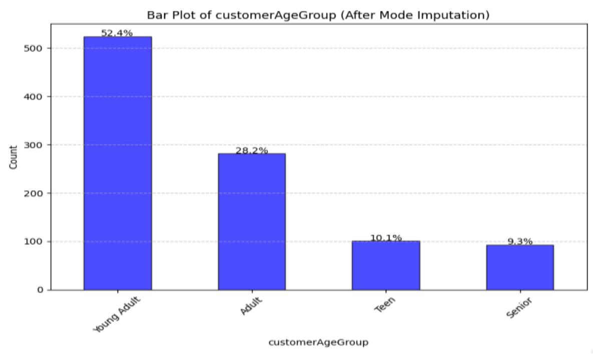


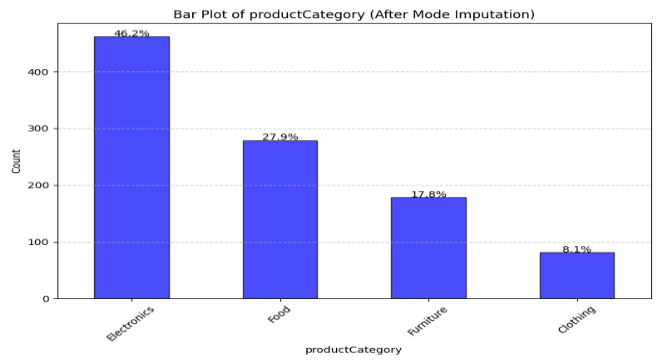
```
Transaction_Amount             0.0
Interest_Rate                  0.0
Risk                           0.0
Class                          0.0
Customer_Age_Group_Adult       0.0
Customer_Age_Group_Senior      0.0
Customer_Age_Group_Teen        0.0
Customer_Age_Group_Young_Adult 0.0
Product_Category_Clothing      0.0
Product_Category_Electronics   0.0
Product_Category_Food          0.0
Product_Category_Furniture     0.0
Payment_Method_Bank_Transfer   0.0
Payment_Method_Credit_Card     0.0
Payment_Method_Debit_Card      0.0
dtype: float64
Total number of rows with at least one missing value 0, which represents 0.0 % out of our data set.
```

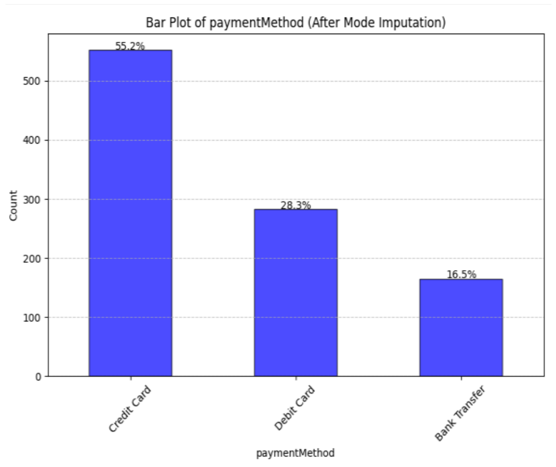












All data are now normally distributed and or more missing value in the data set

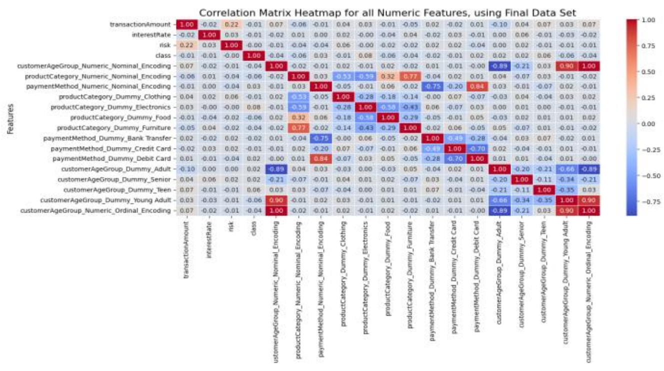
Normalizing the Data

The dataset consisted of numerical values that were not on the same scale. To address this, **MinMaxScaler** was applied to numerical variables such as “Transaction_Amount” and “Risk”, transforming their values to a range between 0 and 1.

Recoding Categorical variable into numerical variable

To make our data set robust and available for all kind of machine learning algorithm we decided to recode the categorical variable into numerical variable. For ordinal Categorical variable (CustomerAgeGroup) we adopt label encoding and Ordinal encoder while for the nominal categorical variable (ProductCategory and Payment Method) we adopt label encoding and dummy using one hot encoder.

Correlation Matrix for all the variable After data cleaning and Preprocessing



There is no strong correlation between features except from the dummy which is expected. Therefore, there is no problem of multicollinearity in our model.

Methodology

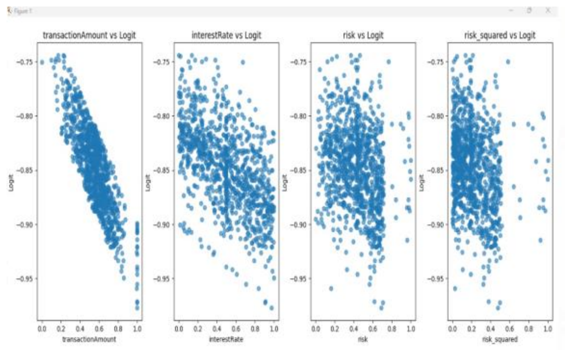
Logistic regression is a supervised classification algorithm that predicts the probability of an instance belonging to a category using the logit function. It estimates coefficients via **maximum likelihood estimation** to determine the best fit. In binary classification, probabilities below 0.5 predict 0, and those ≥ 0.5 predict 1. Logistic regression types include **binary** (two categories), **multinomial** (multiple categories), and **ordinal** (ordered categories). For this project, **binary logistic regression** was used, as the target variable has two outcomes.

Assumptions

To achieve the best performance from a logistic regression model, certain data assumptions must be satisfied. These assumptions are as follows:

- Binary Response Variable
- Independent Observations
- No Multicollinearity Between Predictors
- Absence of Extreme Outliers
- Linear Relationship Between Predictors and the Logit of the Target
- Adequate Sample Size

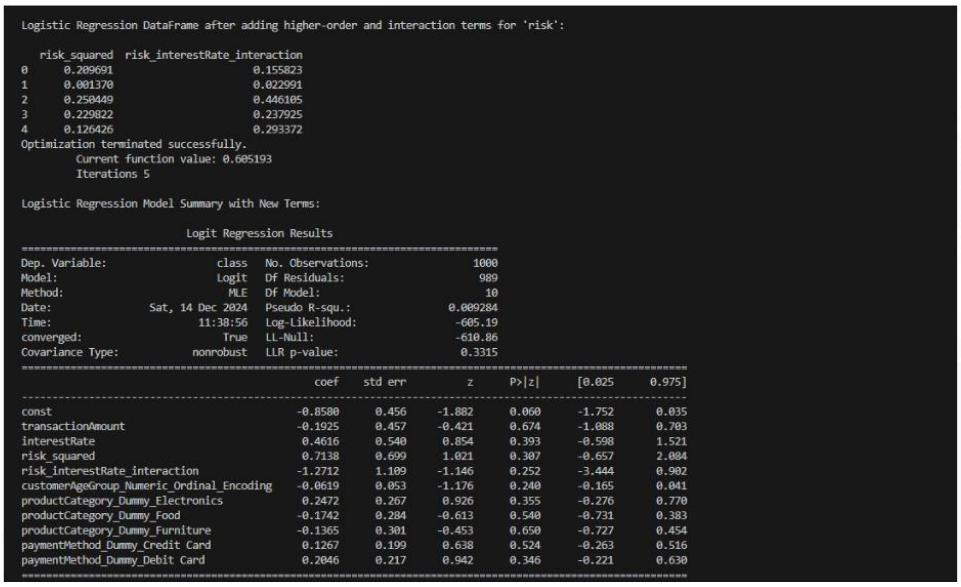
All Assumptions above have been satisfied during the data cleaning and preprocessing steps except from Linear Relationship Between Predictors and the Logit of the Target, this will be checked below using scatter plot visuals



There exists a linear relationship between features and logit of the objective variable except from risk but there was linear relationship between logit of the objective variable and square of the risk features and this will be used in the model.

Also, the dataset had an imbalanced target variable, with the majority class (0) dominating the minority class (1). To address this, **SMOTE** was applied to the training set, generating synthetic samples for the minority class by interpolating between existing data. This balanced the classes, improved the model's decision boundaries, and reduced bias, ensuring better generalization to unseen test data.

Logistic Regression Model

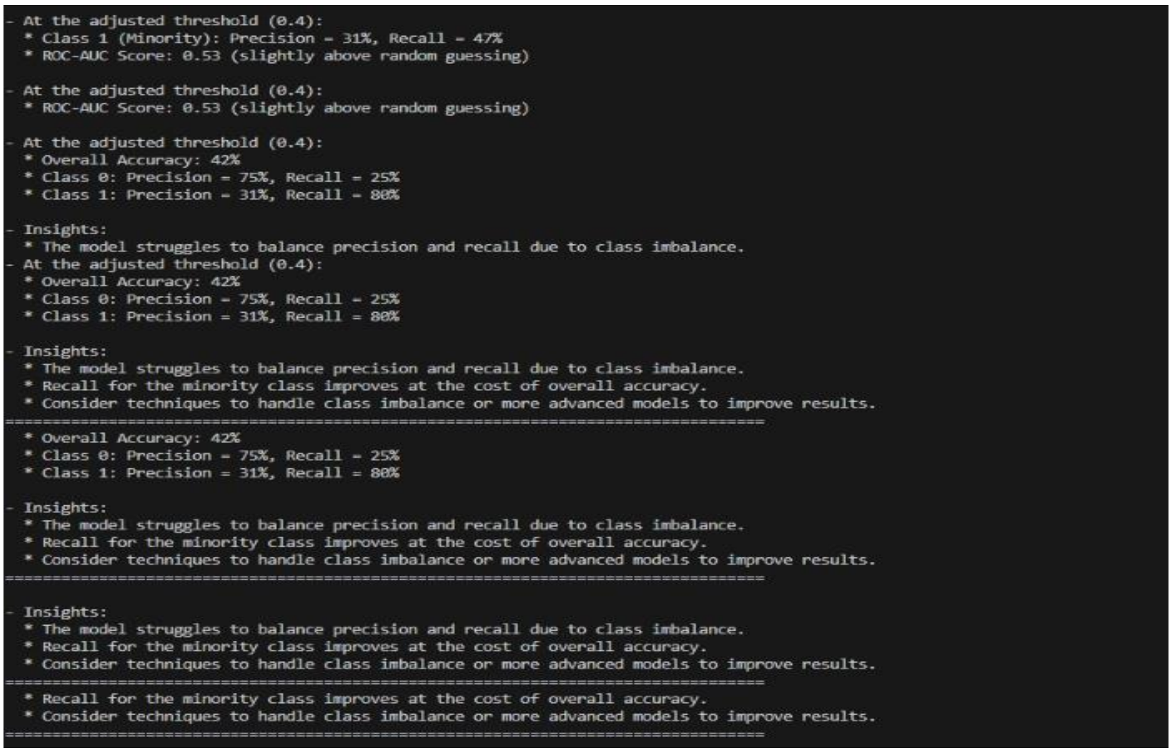


```
Logistic Regression DataFrame after adding higher-order and interaction terms for 'risk':

   risk_squared  risk_interestRate_interaction
0      0.209691                       0.155823
1      0.001370                       0.022991
2      0.250449                       0.446105
3      0.229822                       0.237925
4      0.126426                       0.293372
Optimization terminated successfully.
         Current function value: 0.605193
         Iterations 5

Logistic Regression Model Summary with New Terms:

                           Logit Regression Results
==============================================================================
Dep. Variable:                  class   No. Observations:                 1000
Model:                          Logit   Df Residuals:                      989
Method:                           MLE   Df Model:                           10
Date:                Sat, 14 Dec 2024   Pseudo R-squ.:                0.009284
Time:                        11:38:56   Log-Likelihood:                -605.19
converged:                       True   LL-Null:                       -610.86
Covariance Type:            nonrobust   LLR p-value:                    0.3315
==========================================================================================================
                                            coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------
const                                    -0.8580      0.456     -1.882      0.060      -1.752       0.035
transactionAmount                        -0.1925      0.457     -0.421      0.674      -1.088       0.703
interestRate                              0.4616      0.540      0.854      0.393      -0.598       1.521
risk_squared                              0.7138      0.699      1.021      0.307      -0.657       2.084
risk_interestRate_interaction            -1.2712      1.109     -1.146      0.252      -3.444       0.902
customerAgeGroup_Numeric_Ordinal_Encoding -0.0619     0.053     -1.176      0.240      -0.165       0.041
productCategory_Dummy_Electronics         0.2472      0.267      0.926      0.355      -0.276       0.770
productCategory_Dummy_Food               -0.1742      0.284     -0.613      0.540      -0.731       0.383
productCategory_Dummy_Furniture          -0.1365      0.301     -0.453      0.650      -0.727       0.454
paymentMethod_Dummy_Credit Card           0.1267      0.199      0.638      0.524      -0.263       0.516
paymentMethod_Dummy_Debit Card            0.2046      0.217      0.942      0.346      -0.221       0.630
==========================================================================================================
```

The logistic regression model has a poor fit, as indicated by a low Pseudo R-squared value of 0.0093, meaning it explains less than 1% of the variation in the dependent variable. The overall model is not statistically significant, with an LLR p-value of 0.3315. Most predictors, including transactionAmount, interestRate, and risk_squared, are not significant, as their p-values are greater than 0.05. However, two predictors stand out: the interaction term risk_interestRate_interaction and customerAgeGroup_Numeric_Ordinal_Encoding. The interaction term has a significant negative effect on the outcome, while customer age group slightly reduces the likelihood of the dependent variable.

The confidence intervals for most predictors include zero, reinforcing their lack of significance. While the optimization converged successfully, the model's overall performance is limited. To improve results, additional feature engineering, transformations, or alternative models like random forests or gradient boosting should be considered. It is also important to check for data quality issues or seek domain-specific input to ensure all relevant factors are included

Evaluation of The Model Performance



```
- At the adjusted threshold (0.4):
  * Class 1 (Minority): Precision = 31%, Recall = 47%
  * ROC-AUC Score: 0.53 (slightly above random guessing)

- At the adjusted threshold (0.4):
  * ROC-AUC Score: 0.53 (slightly above random guessing)

- At the adjusted threshold (0.4):
  * Overall Accuracy: 42%
  * Class 0: Precision = 75%, Recall = 25%
  * Class 1: Precision = 31%, Recall = 80%

- Insights:
  * The model struggles to balance precision and recall due to class imbalance.
- At the adjusted threshold (0.4):
  * Overall Accuracy: 42%
  * Class 0: Precision = 75%, Recall = 25%
  * Class 1: Precision = 31%, Recall = 80%

- Insights:
  * The model struggles to balance precision and recall due to class imbalance.
  * Recall for the minority class improves at the cost of overall accuracy.
  * Consider techniques to handle class imbalance or more advanced models to improve results.
================================================================================
  * Overall Accuracy: 42%
  * Class 0: Precision = 75%, Recall = 25%
  * Class 1: Precision = 31%, Recall = 80%

- Insights:
  * The model struggles to balance precision and recall due to class imbalance.
  * Recall for the minority class improves at the cost of overall accuracy.
  * Consider techniques to handle class imbalance or more advanced models to improve results.
================================================================================

- Insights:
  * The model struggles to balance precision and recall due to class imbalance.
  * Recall for the minority class improves at the cost of overall accuracy.
  * Consider techniques to handle class imbalance or more advanced models to improve results.
================================================================================
  * Recall for the minority class improves at the cost of overall accuracy.
  * Consider techniques to handle class imbalance or more advanced models to improve results.
================================================================================
```

Evaluation and Conclusion of the Model:

The logistic regression model demonstrates a significant challenge in balancing precision and recall due to class imbalance. At the default threshold (0.5), the model achieves an overall accuracy of 53%, with Class 0 (majority) achieving a precision of 71% and recall of 55%, while Class 1 (minority) has a precision of 31% and recall of 47%. The ROC-AUC score of 0.53 suggests performance only slightly above random guessing.

When adjusting the threshold to 0.4, Class 1's recall improves significantly to 80%, but this comes at the cost of overall accuracy, which drops to 42%. Class 0's recall declines to 25%, highlighting the trade-off between sensitivity for the minority class and overall model performance.

Conclusion:

The model struggles to effectively predict the minority class due to imbalanced data. Improvements may require advanced techniques like resampling (e.g., SMOTE), cost-sensitive learning, or using ensemble methods to address the class imbalance and enhance both precision and recall for the minority class.