FRAUD DETECTION IN FINANCIAL TRANSACTION

Table of Contents

1. Introduction

Fraudulent activities in the financial transactions are a big worry for businesses and people as well. This scheme employs a machine learning system based on Logistic Regression. Transactions are then identified as fake or real by means of the project. The research part takes under data preprocessing, exploratory data analysis (EDA), feature engineering, model training, evaluation, and the visualization of the results. The information that is taken out from the analysis points to the possibilities of the enhancement of fraud detection mechanisms and the reduction of financial risks.

2. Dataset Overview

Number of rows and columns: **1,000 rows**, **7 columns**

Key features:

- **Transaction Amount**: Continuous variable.
- **Customer Age Group**: Categorical variable.
- **Merchant Category**: Categorical variable.
- **Transaction Time**: Categorical variable.
- **Payment Method**: Categorical variable.
- **Risk Score**: Continuous variable.
- **Fraudulent**: Target variable (1 for fraud, 0 for genuine)

3. EDA

3.1 Checking Duplicate rows & Missing values

- Check for the duplicate rows & remove them if needed. However, there is no duplicated data.
- Identify if there is any missing values. However, there is no missing values

```
Dimension of the DataFrame (no of rows, no of columns): (1000, 7)
Number of duplicate rows: 0
Missing data in each column: Transaction Amount    0
Customer Age Group    0
Merchant Category     0
Transaction Time      0
Payment Method        0
Risk Score            0
Fraudulent            0
dtype: int64
```

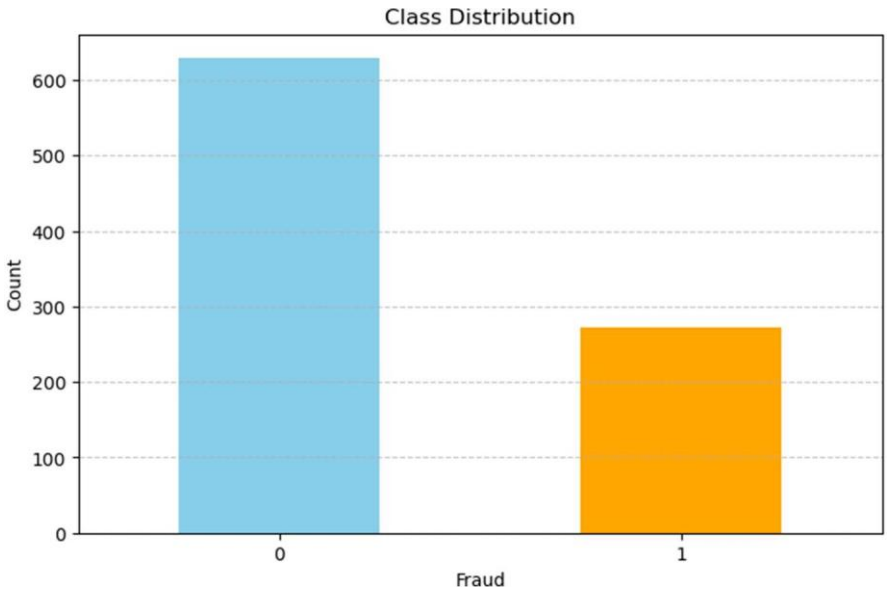
3.2 Outlier Detection and Removal

- Outliers in Transaction Amount (99 outliers) were detected using the Interquartile Range (IQR) method.
- 99 outliers were removed.

```
Number of outliers for each numerical feature:
Transaction Amount: 99
Risk Score: 0
Fraudulent: 0
Dimension of the DataFrame after removing outliers: (901, 7)
```

3.3 Class Distribution of Target Variable

- Class 1 (Fraudulent) is noticeably lower, indicating that it is the minority class while Class 0 (Non- Fraudulent) is significantly higher.
- This distribution suggests an imbalanced dataset, so it is required to handling the balancing during the modeling phase to ensure fair performance in the contexts of fraud detection.



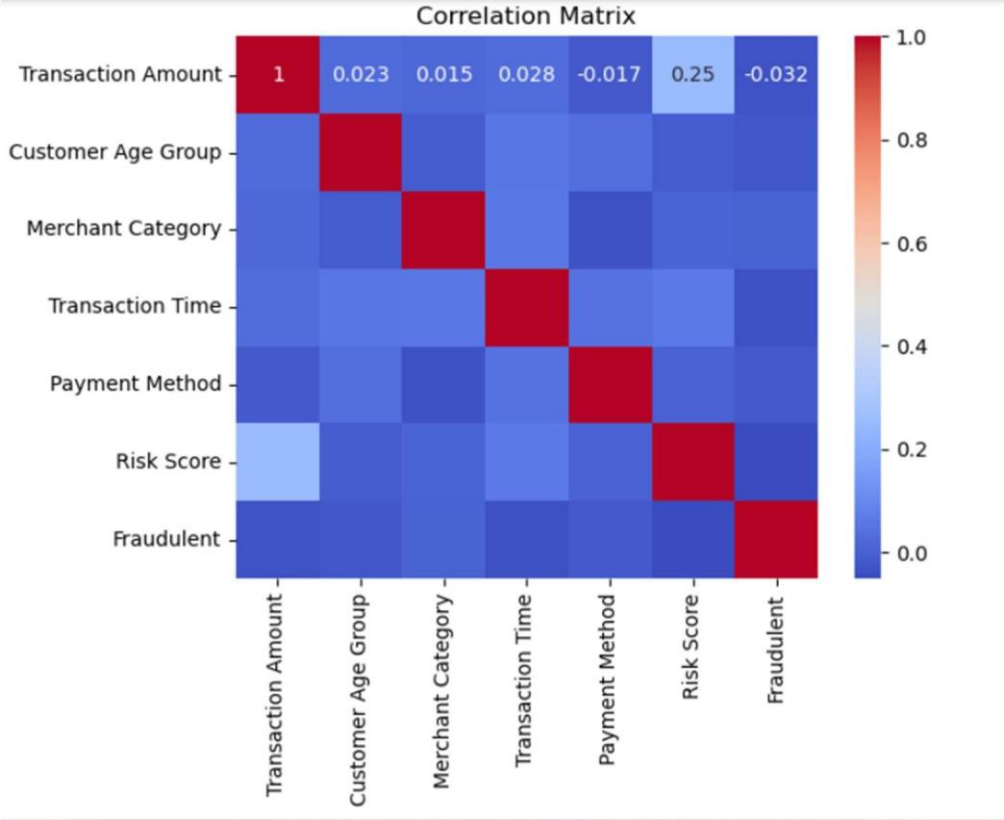
4. Data Preprocessing

4.1 Convert Categorical Variable to Numeric

- There are some categorical variables (such as Customer Age Group, Merchant Category, Transaction Time, Payment Method).
- Because the model (i.e. Logistic Regression) required numerical input, categorical variables need to be properly transformed before they can be fed into a model.

4.2 Correlation

- Correlation matrix is a crucial step in understanding the relationships between variables in your dataset, especially in the contexts like fraud detection.
- Correlation is
    - To understand relationship between variables
    - To select features for Modeling
    - To detect multicollinearity
- Key observations:
    - The correlation between **Transaction Amount** and **Risk Score** is moderate (~0.25), which could indicate that higher transaction amounts are somewhat linked to higher risk.
    - Other variables, such as **Customer Age Group**, **Merchant Category**, and **Payment Method**, have very weak correlations with **Transaction Amount** and **Risk Score.**
    - The **Fraudulent** column shows weak or negligible correlations with most other variables, including **Transaction Amount** (-0.032).

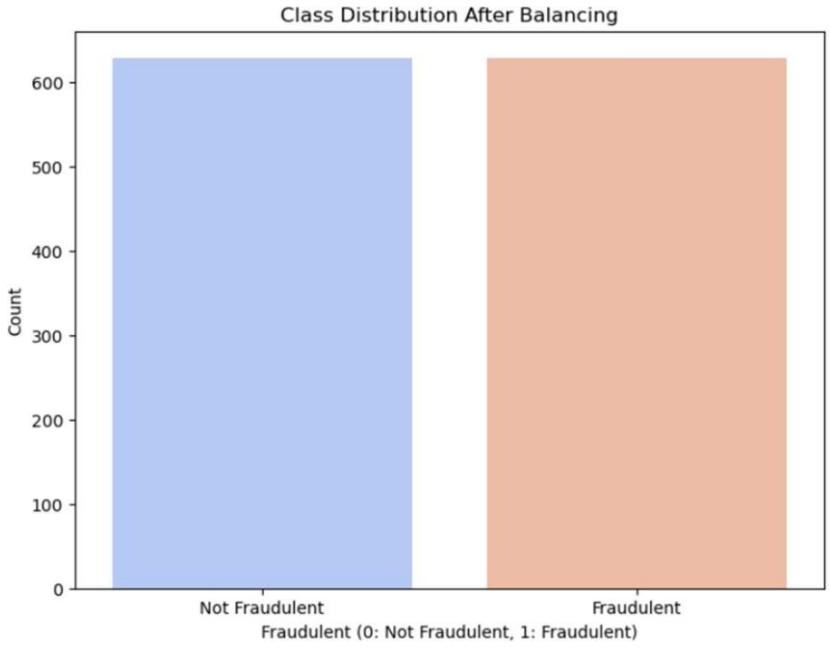


4.3 Featuring Engineering

- Generating Interaction Terms
  - We use Polynomial Features to generate interaction terms. They capture relationships between two or more features by creating new features that represent the combination.
  - The output of interaction terms are added back to the original dataset and the target variable (Fraudulent) is also included.
- After that, the data was separated into feature (x) and the target variable (y).

4.4 Balancing the Data

- To balance the dataset, SMOTE (Synthetic Minority Over-sampling Technique) is used. SMOTE helps to oversample the minority class by generating synthetic samples. It helps balance the dataset by ensuring that the number of fraudulent and non-fraudulent transactions are more equal, which improves model performance when dealing with imbalanced classes.
- Visualization of the distribution of the target variable (Fraudulent):



4.5 Normalizing the data

- Normalizing the features is an important step in the preprocessing step, especially when we are using machine learning models that are sensitive to the scale of input features, such as logistic regression, or k-nearest neighbors.
- StandardScaler is used to standardize the data by removing the mean and scaling to unit variance.
- After applying the transformation, the features will all have the same scale.

5. Methodology

5.1 Model Selection

Although the correlations are weak, the models (like Logistic Regression) may still uncover meaningful relationships when combining features.

- Logistic Regression
  - Simple and Interpretable
  - Handles Binary Classification well
  - Efficient and Scalable
- K-Nearest Neighbors (KNN)
  - Non-Linear decision boundaries, meaning that the model does not assume any distribution of the data
  - Simplicity and Effectiveness
  - Flexibility

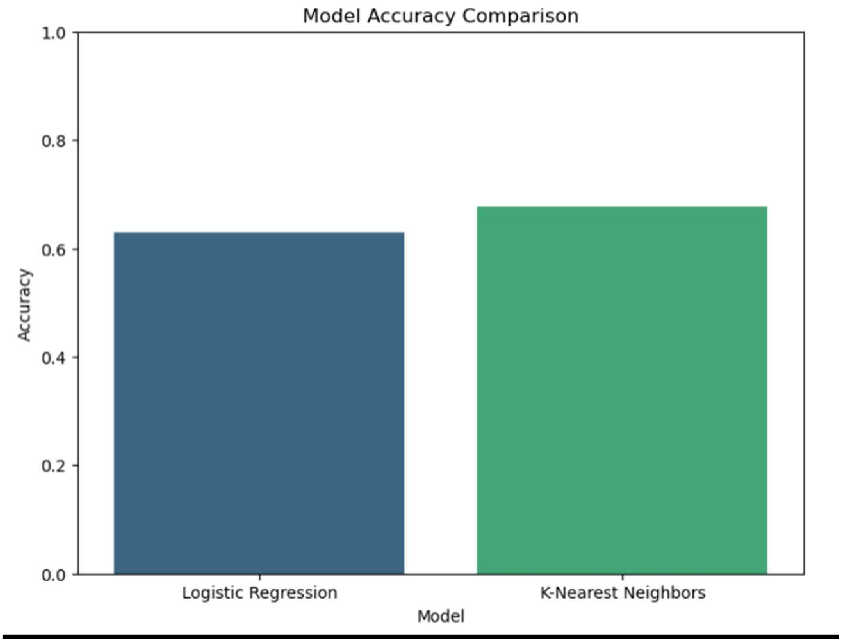
5.2 Data Splitting

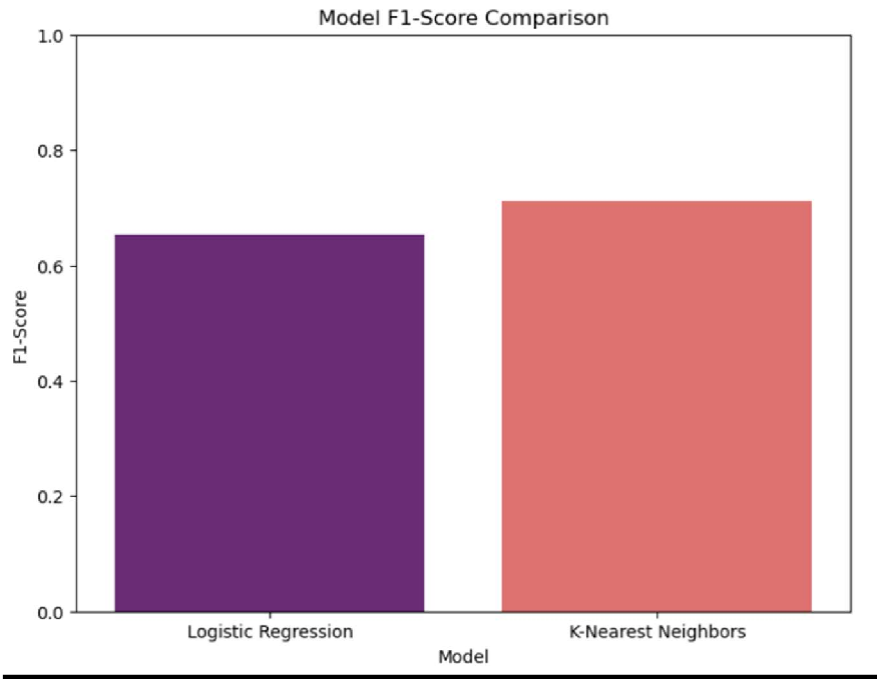
- Training Data: 70% of the dataset.
- Testing Data: 30% of the dataset.
- **Stratification** ensures that the class distribution of the target variable (y_balanced) is preserved in both training and testing sets.

5.3 Model Evaluation

- Accuracy & F1 Score

```
Logistic Regression Accuracy: 0.6296
Logistic Regression F1 Score: 0.6535
KNN Accuracy: 0.6772
KNN F1 Score: 0.7123
```





- Logistic Regression:
  - Accuracy is 0.6296, indicating that the Logistic Regression model correctly predicted 62.96% of the test samples.
  - The F1 score (0.6535) is relatively good, suggesting a reasonable balance between precision and recall, especially considering the likely class imbalance in fraud detection.
- KNN:
  - The KNN model has a higher accuracy than Logistic Regression, correctly predicting 67.72% of the test samples.
  - The F1 score for KNN is better than Logistic Regression, indicating that KNN is achieving a better balance between precision and recall, which is important for identifying fraudulent transactions.

5.4 The Best Model

- **Logistic Regression:**
  - Performance Metrics:
    - **Accuracy**: 0.63 (63%)
    - **F1-Score**: 0.65 (weighted balance between precision and recall)
    - **Precision & Recall**:
      - Class 0 (non-fraud): Precision = 0.65, Recall = 0.56
      - Class 1 (fraud): Precision = 0.61, Recall = 0.70
    - I**nterpretation**:

        - Logistic regression performs slightly better at detecting fraudulent transactions (class 1) than non-fraudulent transactions (class 0), as shown by the higher recall for class 1.
        - However, it struggles with false positives (precision for class 1 is 0.61) and false negatives (recall for class 0 is 0.56).
    - Confusion Matrix:
        - Class 0 (non-fraud): 106 correct predictions, 83 misclassified as fraud.
        - Class 1 (fraud): 132 correctly identified as fraud, 57 misclassified as non-fraud.
        - The confusion matrix indicates a moderate bias toward detecting fraud.

- **KNN**:
    - **Performance Metrics**:
        - **Accuracy**: 0.68 (68%)
        - **F1-Score**: 0.71 (higher than logistic regression)
        - **Precision & Recall**:
            - Class 0 (non-fraud): Precision = 0.73, Recall = 0.56
            - Class 1 (fraud): Precision = 0.64, Recall = 0.80
        - **Interpretation**:
            - KNN performs better overall, especially for identifying fraudulent transactions (class 1), with a higher recall (0.80).
            - However, KNN sacrifices recall for class 0 (non-fraud), which remains low (0.56).
    - **Confusion Matrix**:
        - Class 0 (non-fraud): 105 correct predictions, 84 misclassified as fraud.
        - Class 1 (fraud): 151 correctly identified as fraud, 38 misclassified as non-fraud.
        - KNN is better at minimizing false negatives for class 1 (fraudulent cases), which is important in fraud detection.

- **Comparison**:
    - **Accuracy**:
        - KNN performs better (68%) than logistic regression (63%).
    - **F1-Score**:

        - KNN has a higher F1-Score (0.71) compared to logistic regression (0.65), indicating it strikes a better balance between precision and recall.
    - **Fraud Detection (Class 1)**:
        - KNN outperforms logistic regression in detecting fraud, with higher recall (0.80 vs. 0.70) and a better balance of precision and recall.
    - **Non-Fraud Detection (Class 0)**:
        - Logistic regression performs slightly better in classifying non-fraudulent cases, with slightly higher precision (0.65 vs. 0.73 for KNN) and a similar recall (0.56).
- **Conclusion**: based on the results, **KNN appears to be the better choice for detecting fraudulent transactions.**

```
--- Logistic Regression Results ---
Accuracy: 0.63, F1-Score: 0.65
Confusion Matrix:
 [[106  83]
 [ 57 132]]
Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.56      0.60       189
           1       0.61      0.70      0.65       189

    accuracy                           0.63       378
   macro avg       0.63      0.63      0.63       378
weighted avg       0.63      0.63      0.63       378


--- K-Nearest Neighbors Results ---
Accuracy: 0.68, F1-Score: 0.71
Confusion Matrix:
 [[105  84]
 [ 38 151]]
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.56      0.63       189
           1       0.64      0.80      0.71       189

    accuracy                           0.68       378
   macro avg       0.69      0.68      0.67       378
weighted avg       0.69      0.68      0.67       378


--- Best Model ---
The best model is: K-Nearest Neighbors with an accuracy of 0.68
```

6. Conclusion

- K-Nearest Neighbors (KNN) outperforms logistic regression in both accuracy (68% vs. 63%) and F1-Score (0.71 vs. 0.65).
- Fraud Detection (Class 1): KNN demonstrates better recall for fraudulent transactions (0.80 vs. 0.70), which is crucial in fraud detection where missing a fraudulent transaction (false negatives) has a higher cost than a false positive.
- Fraud detection systems prioritize recall for the fraudulent class (class 1) to minimize the risk of undetected fraudulent transactions. KNN, with its higher

recall for fraud (0.80), is better suited for this application. Therefore, **KNN is the recommended model** based on the current analysis

- However, further optimizations can be explored:
    - Fine-tune the number of neighbors (n_neighbors) and weights in KNN.
    - Test additional algorithms like Random Forest to further improve performance.
    - Consider evaluating models with metrics like ROC-AUC to confirm their discriminative ability.
- Regarding Logistic Regression, as the correlations among the features are not strong, there is a consideration to drop these features and replace by other features which are highly correlation.