University of Niagara Falls Canada

Master of Data Analytics

Predictive Analytics (DAMO-510-6)

Winter 2025

Assignment #2

Credit Card Fraud Prediction Using Multiple Machine Learning Models

Submitted by

Arya Kaippully (NF1010532)

Catherine Calantoc (NF1001422)

Jayasri Katragadda (NF1000622)

Sridevi Pemmasani (NF1002949)

Professor: Foad Aghamiri

February 2025

Table of Contents

1 INTRODUCTION 4

1.1 Statement of Purpose 4

1.2 Problem Statement 4

1.3 Objective 4

1.4 Key Goal Analyses 5

1.5 Research Questions 6

2 SCOPE OF THE PROJECT 6

2.1 Key Components of the Project: 6

2.2 Limitations and Constraints 8

3 BACKGROUND RESEARCH AND LITERATURE REVIEW 9

3.1 Background Research 9

3.2 Literature Review 9

3.3 Gaps in Literature 10

4 DESIGN AND DATA COLLECTION METHODS 11

4.1 Dataset Overview 11

4.2 Dataset Attributes 11

4.3 Data Cleaning and Preprocessing 12

4.4 Categorical Variable Analysis 14

4.5 Exploratory Data Analysis (EDA) 14

4.6 Data Splitting & Model Training Strategy 17

5 METHODOLOGY/STRATEGIES 18

5.1 Univariate Analysis (Descriptive Statistics) 18

5.2 Bivariate Analysis (Feature Relationships & Correlation Analysis) 18

5.3 Feature Transformation & Selection 19

5.4 Machine Learning Models 19

5.5 Evaluation of Linear, Log-transformed, Square Root and Box-Cox Models: 22

5.6 Decision Tree Model 24

5.7 Evaluation of Decision-Tree Model using Confusion Matrix: 25

5.8 Decision Tree Classifier (Model Optimization) 27
5.9 Build additional Models - LDA, QDA and KNN 28
5.10 Model Comparison of all the performed Models in Terms of Accuracy: 29
6 KEY INSIGHTS 30
6.1 Fraudulent Transaction Patterns 30
6.2 Model Performance Analysis 30
6.3 Business Impact and Fraud Prevention 31
6.4 Recommendations for Future Improvements 31
7 CONCLUSION 33
8 REFERENCES 34

1 INTRODUCTION

1.1 Statement of Purpose

Credit card fraud is a growing concern in financial transactions, leading to significant financial losses for businesses and consumers. Fraudulent transactions not only impact individuals but also burden financial institutions with chargebacks, security costs, and regulatory challenges.

1.2 Problem Statement

Despite advancements in fraud detection techniques, major challenges persist, including:

- High False Positive Rates – Many legitimate transactions are incorrectly flagged as fraudulent, leading to customer dissatisfaction.
- Data Imbalance – Fraudulent transactions constitute only a small fraction of total transactions, making accurate detection difficult.
- Real-Time Detection Challenges – Traditional fraud detection systems often struggle with identifying evolving fraud patterns in real-time.
- Lack of Model Transparency – Many fraud detection models function as black boxes, making regulatory compliance and interpretation difficult.

1.3 Objective

This project aims to develop a predictive model for fraud detection using machine learning techniques, particularly logistic regression and ensemble models. The primary goals are:

- Enhance fraud detection accuracy while reducing false positives.
- Leverage statistical analysis and data preprocessing techniques to improve model performance.

- Evaluate multiple models and transformations to find the best-performing fraud detection approach.

1.4 Key Goal Analyses

To achieve these objectives, the project focuses on the following key analyses:

1. Model Comparison for Best Performance
    - Fraud detection requires a model that balances precision and recall effectively.
    - We compare different models based on key performance metrics such as accuracy, precision, recall, and AUC-ROC.
2. Data Transformation for Improved Predictive Power
    - Using techniques such as Box-Cox and Square Root transformations to improve data distribution and reduce skewness.
    - Ensuring feature scaling and normalization to enhance model stability.
3. Addressing Data Challenges
    - Fraud datasets are highly imbalanced (fewer fraudulent transactions than non-fraudulent ones).
    - We apply oversampling/undersampling techniques and transformations to handle class imbalance.
4. Ensuring Robustness of the Fraud Detection Model
    - Testing different feature selection techniques to identify the most significant fraud indicators.
    - Evaluating the model using real-world scenarios to minimize false positives and improve decision-making.

1.5 Research Questions

To further refine our approach, we address the following research questions:

1. Which machine learning model performs best in detecting fraudulent transactions?
2. How do data transformation techniques (Box-Cox, Log, and Square Root) improve model performance?
3. What impact does feature selection have on fraud detection accuracy?
4. How can financial institutions implement real-time fraud detection using predictive models?

This study will contribute to enhancing security measures in financial transactions by developing a robust, interpretable, and scalable fraud detection model.

2 SCOPE OF THE PROJECT

This project is structured to effectively analyze various machine learning techniques for fraud detection using Python. The key focus areas of the study include data preprocessing, exploratory data analysis, feature engineering, model development, and evaluation to ensure the accuracy and efficiency of fraud detection.

2.1 Key Components of the Project:

1. Data Preprocessing
    - Handling missing values, outliers, and duplicate entries to maintain dataset integrity.
    - Standardizing and scaling numerical features to ensure uniformity across variables.

- Data balancing techniques (such as oversampling or undersampling) to address class imbalance in fraudulent transactions.

2. Exploratory Data Analysis (EDA)
    - Using summary statistics and visualization techniques (heatmaps, histograms) to identify trends in fraudulent and non-fraudulent transactions.
    - Detecting potential correlations between transaction attributes.
3. Feature Engineering
    - Implementing transformations such as log transformation, Box-Cox, and Square Root transformation to stabilize data distribution.
    - Encoding categorical variables and performing feature selection to optimize input variables for machine learning models.
4. Machine Learning Models
    - Logistic Regression is considered as the baseline model for fraud detection.
    - Decision Tree Classifier to capture non-linear relationships in the dataset.
    - Random Forest Classifier to improve predictive accuracy through ensemble learning.
    - Hyperparameter Optimization using GridSearchCV to fine-tune model parameters for better performance.
5. Model Evaluation
    - Assessing predictions using key performance metrics:
        - Accuracy – Measures overall correctness.
        - Precision & Recall – Evaluates fraud detection efficiency.

        - AUC-ROC Curve – Determines model effectiveness in distinguishing fraudulent transactions.

6. Automation & Reproducibility
    - The entire pipeline, from data pre-processing to fraud detection model execution, is fully automated to function without human intervention.
    - Ensuring reproducibility so that results can be consistently replicated across different computing environments.

2.2 Limitations and Constraints

While this project aims to enhance fraud detection accuracy, certain constraints exist:

- Data Privacy & Security: The dataset is publicly available, but real-world fraud detection involves sensitive financial data that must comply with privacy regulations.
- Computational Complexity: Fraud detection on large-scale datasets requires high-performance computing capabilities, which may limit real-time processing.
- Model Deployment Challenges: Implementing this model in a real-world financial institution requires integration with existing fraud detection systems, which can be complex.
- Evolving Fraud Techniques: Fraudsters continuously adapt to detection mechanisms, requiring continuous model updates to maintain accuracy.

This project aims to provide a robust, scalable, and interpretable fraud detection solution, ensuring high accuracy with minimal false positives.

3 BACKGROUND RESEARCH AND LITERATURE REVIEW

3.1 Background Research

Research on identifying fraudulent credit card transactions has increased significantly due to the rise in digital payments. One of the key challenges in fraud detection is the imbalance of datasets, where fraudulent transactions represent only a small fraction of the total data. Traditional fraud detection methods often struggle with high false positive rates, lack of transparency, and adaptability to evolving fraud patterns.

Early fraud detection systems primarily relied on rule-based models, but modern approaches incorporate machine learning algorithms to enhance fraud detection accuracy. Various techniques, including oversampling methods, feature selection, and transformation techniques, have been explored to address data imbalance and model performance issues.

3.2 Literature Review

I. Credit Card Fraud Detection: A Realistic Modeling Approach

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784-3797.

- This study formalizes the fraud detection problem by simulating real-world financial transactions.
- It highlights the impact of class imbalance on fraud detection and proposes data balancing techniques to improve model performance.
- The research emphasizes the importance of feature selection and sampling strategies, influencing our decision to apply Box-Cox transformation, log

transformation, and Square Root transformation to improve fraud detection accuracy.

II. **A Comprehensive Review of Financial Fraud Detection**

West, J., & Bhattacharya, M. (2016). Intelligent Financial Fraud Detection: A Comprehensive Review. Computers & Security, 57, 47-66.

- This review explores various machine learning techniques for fraud detection, including neural networks, decision trees, and SVMs.
- It examines real-time fraud detection challenges and highlights the need for effective feature selection to improve fraud classification accuracy.
- The findings support our choice of logistic regression as a baseline model while also comparing it with ensemble methods like Random Forest for improved fraud detection.

3.3 Gaps in Literature

Despite existing research, the following gaps remain:

1. Scalability Issues – Many studies focus on small datasets, but fraud detection systems must handle large-scale, real-time transactions.
2. Explainability & Transparency – Many machine learning models act as black boxes, making it difficult for financial institutions to understand and trust their decisions.
3. Evolving Fraud Patterns – Fraudsters adapt their techniques over time, requiring continuous model retraining for improved fraud detection accuracy.

Our Project's aims to contribute the following:

- To address class imbalance, we incorporate data transformation techniques (Box-Cox, Log, and Square Root) to improve feature stability and model interpretability.

- We compare logistic regression, decision trees, and Random Forest models to find the best-performing fraud detection algorithm.
- Our model prioritizes precision, recall, and AUC-ROC metrics, ensuring a more effective fraud detection system with minimal false positives.

4 DESIGN AND DATA COLLECTION METHODS

4.1 Dataset Overview

The dataset used in this study, titled "Credit Card Fraud Detection," was obtained from Kaggle, a widely recognized platform for machine learning datasets. The dataset contains 14,383 transactions with 29 features, including both numerical and categorical variables. Each transaction is labeled as either fraudulent (is_fraud = 1) or legitimate (is_fraud = 0), making it a binary classification problem.

4.2 Dataset Attributes

To better understand the dataset, we examined its structure using the df.info() function, which provides details about the number of records, data types, and missing values.
Below is a breakdown of key attributes:

Column Name	Description	Data Type
trans_date_trans_time	Date and time of transaction	datetime64[ns]
merchant	Name of the merchant	object
category	Type of transaction (e.g., shopping, groceries)	object
amt	Transaction amount	float64
city	City where the transaction occurred	object
state	State where the transaction occurred	object

lat, long	Latitude and longitude of the transaction	float64
city_pop	Population of the transaction's city	int64
job	Profession of the cardholder	object
dob	Date of birth of the cardholder	datetime64[ns]
trans_num	Unique transaction ID	object
merch_lat, merch_long	Merchant's latitude and longitude	float64
grocery_net_flag, shopping_net_flag, misc_pos_flag, grocery_pos_flag, health_fitness_flag, gas_transport_flag, travel_flag	Binary flags indicating the type of transaction	int64
is_fraud	Fraud label (1 = Fraud, 0 = Legitimate)	int64

4.3 Data Cleaning and Preprocessing

To ensure data quality and reliability, the following preprocessing steps were applied:

1. Handling Missing Values
    - Checked for null values in the dataset.
    - Since the dataset contained minimal missing data, rows with missing values were removed instead of applying imputation on them.
2. Duplicate and Outlier Removal
    - Identified and removed duplicate transactions to avoid data redundancy.
    - Used interquartile range (IQR) to detect and remove extreme outliers in transaction amounts.
3. Feature Selection
    - Selected one numerical variable and one categorical variable for analysis.

- The chosen features were:
    - Transaction Amount – A key numerical factor influencing fraud probability.
    - Payment Category – Certain transaction types have a higher likelihood of fraud. The payment category is one-hot encoded for better analysis.

After changing the categorical data into binary, the dataset is all ready to undergo analysis. The access to the dataset is dynamically provided from the system as:

```
Please provide the dataset file path:  C:/Users/pemma/Documents/UNF/2. Term-2/Predictive Analytics/Project/Credit Card Fraud Detection.xlsx
Thank you !
File found at specified path: C:/Users/pemma/Documents/UNF/2. Term-2/Predictive Analytics/Project/Credit Card Fraud Detection.xlsx
```

```
# Check the number of rows and columns in the dataset
print ("(Rows and Columns) : ", df_CCD_Original.shape)
```

```
(Rows and Columns) :  (14384, 29)
```

The following code describes independent and dependent variables considered for the analysis.

```
# Selecting independent variables and dependent variable
independent_vars = [
    'grocery_net_flag', 'shopping_net_flag', 'misc_pos_flag', 'grocery_pos_flag',
    'health_fitness_flag', 'gas_transport_flag', 'misc_net_flag', 'kids_pets_flag',
    'shopping_pos_flag', 'entertainment_flag', 'food_dining_flag', 'home_flag',
    'personal_care_flag', 'travel_flag', 'amt'
]

X = df_CCD_Original[independent_vars]
y = df_CCD_Original['is_fraud']
```

- Highly correlated features (above 0.85 correlation) were removed to avoid redundancy and multicollinearity.
- One-Hot Encoding was applied to categorical variables (category) to convert them into numerical format.

- Feature importance analysis was conducted to retain only the most significant predictors for fraud classification.

4. Data Transformation Techniques
    - Log Transformation: Applied considering residuals are normally distributed and independent variables are non-skewed. (Same goes with linear regression but Fraud detection is usually done in Log models as the complexity of the data is high in real world applications)
    - Box-Cox Transformation: Applied to reduce skewness in transaction amounts.
    - Square Root Transformation: Used to stabilize variance in highly imbalanced numerical features.
    - StandardScaler & PowerTransformer: Applied to normalize data for better model performance.

4.4 Categorical Variable Analysis

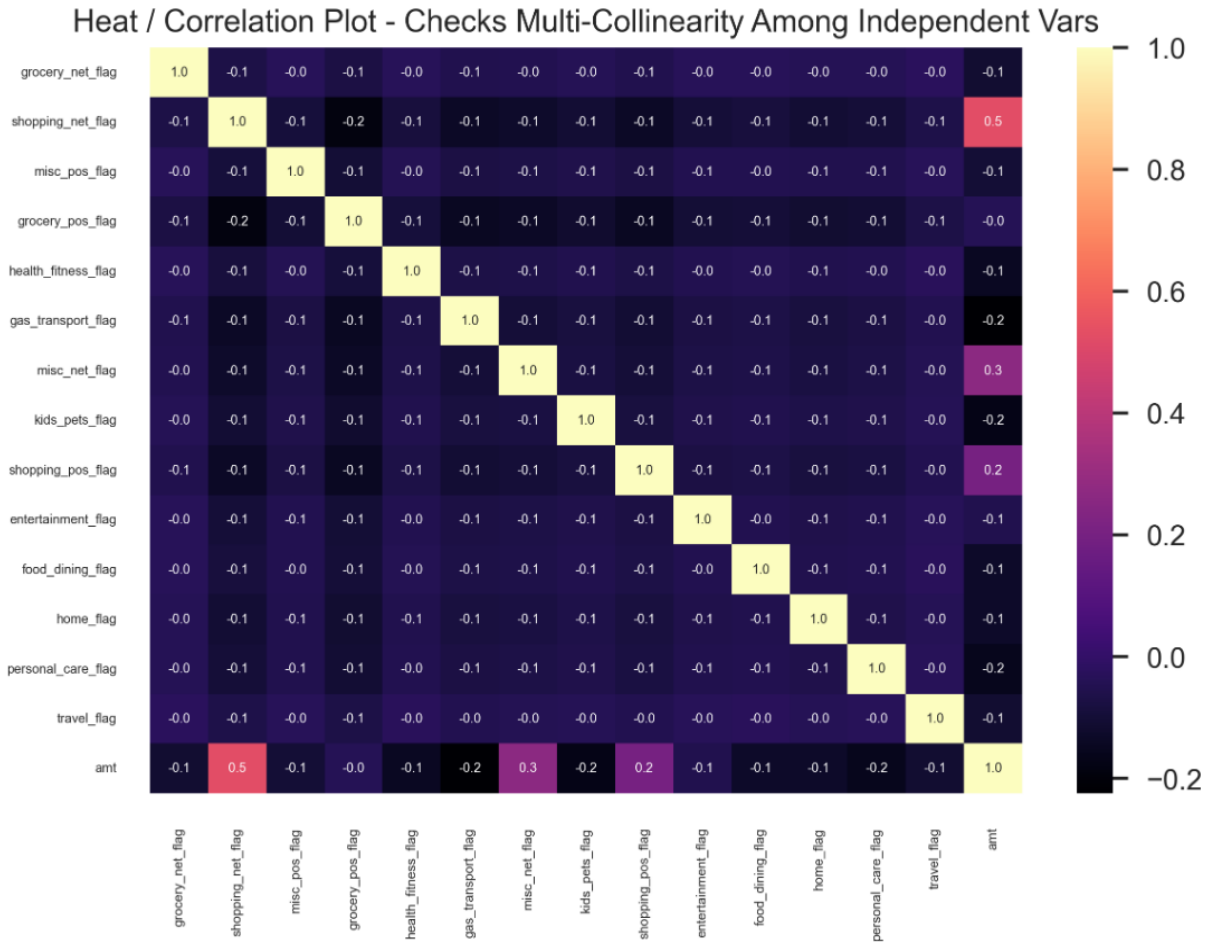
- Examined the impact of Payment Category on Fraud Occurrence.
- Checked for correlation between categorical variables to eliminate redundancy.
- Applied one-hot encoding to convert categorical variables into a machine-readable format.

4.5 Exploratory Data Analysis (EDA)

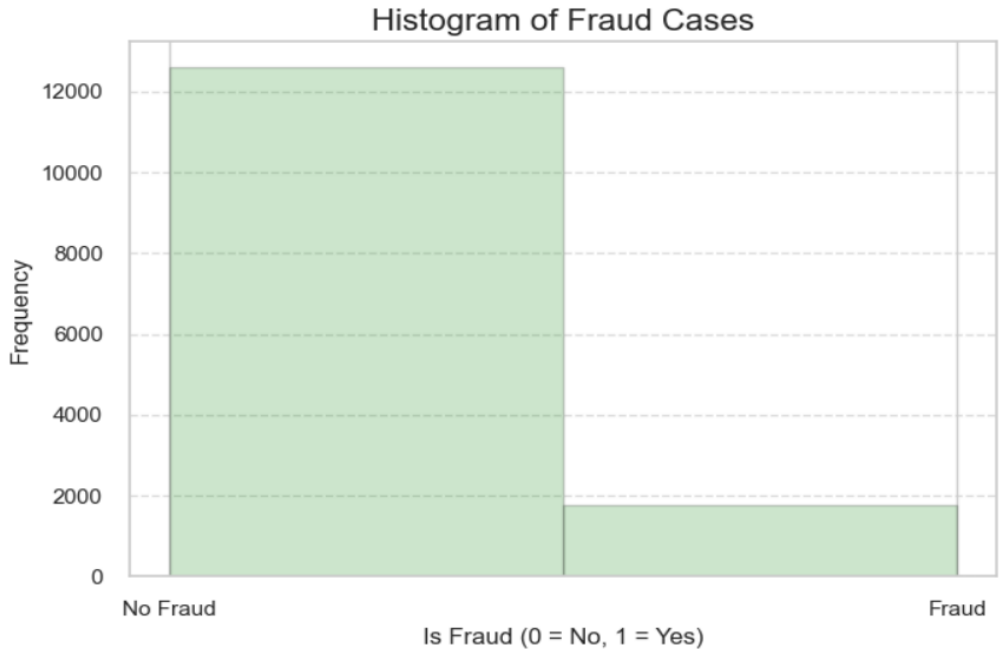
To gain insight into transaction behaviors, the following EDA techniques were performed:

- Statistical Summary: Used df.describe() to analyze the distribution of numerical features.
- Correlation Analysis: Computed correlation coefficients between features to identify dependencies.
- Visualization Techniques:

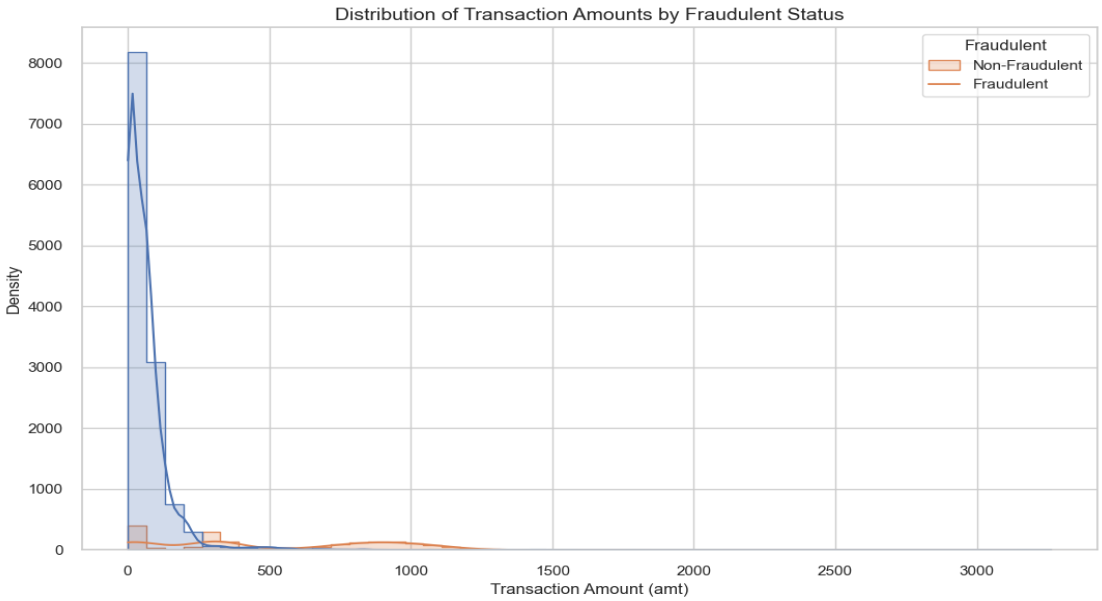
- Heatmaps – Displayed feature correlations to identify relationships:



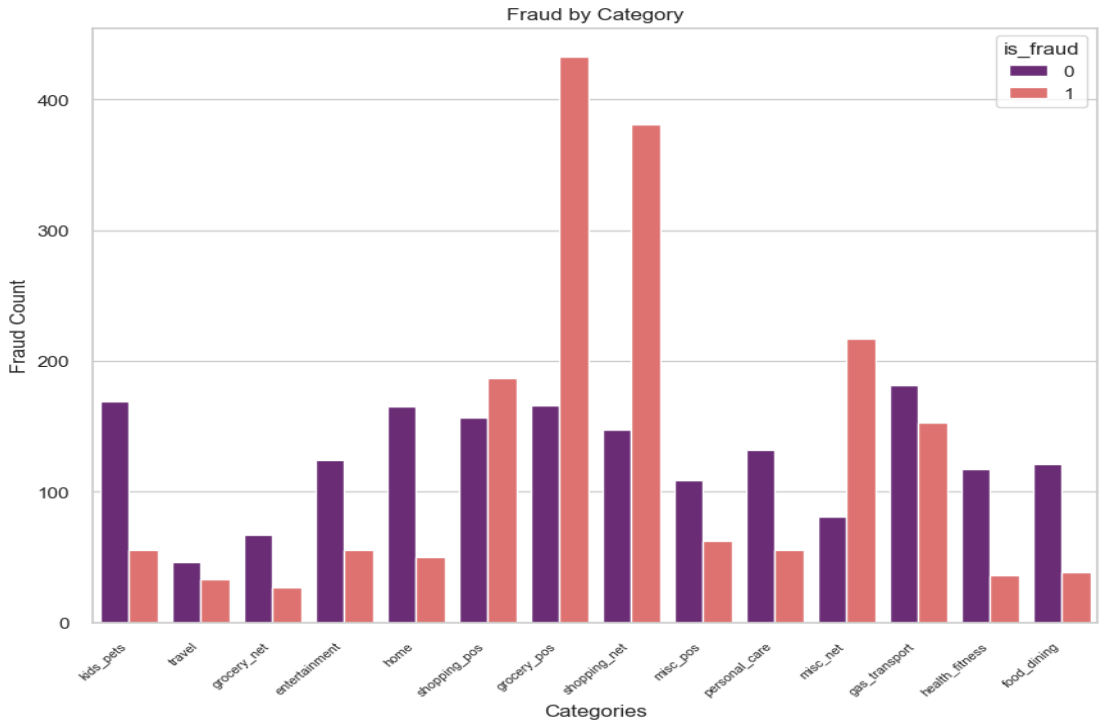
- Histograms & Box Plots – Analyzed distributions and potential anomalies.



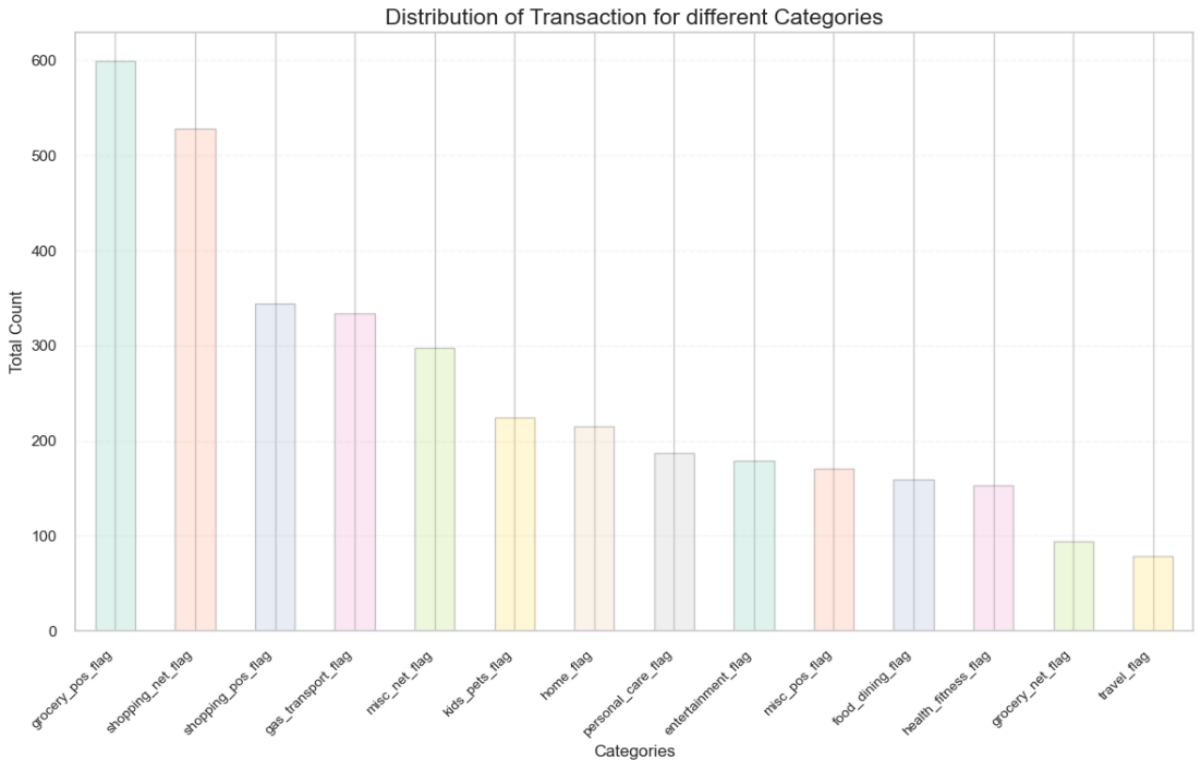
Explored fraud occurrence patterns based on transaction amounts:



- Explored Categorical distribution of Transactions using bar chart:



Visualized Fraud Distribution by Category:



4.6 Data Splitting & Model Training Strategy

To ensure robust model performance, the dataset was split into training and testing sets:

- Train-Test Split:
    - 80% Training Data → Used for model training.
    - 20% Test Data → Used for evaluation.
    - Stratified Sampling was used to preserve class balance, ensuring that both fraudulent and non-fraudulent transactions were proportionally represented in training and testing datasets.
- Random Seed: random_state=42 was used to ensure reproducibility across different executions.

5 METHODOLOGY/STRATEGIES

The methodology used for fraud detection consists of data preprocessing, feature transformation, data splitting, machine learning model selection, and evaluation. This structured approach ensures an optimized, reproducible, and automated fraud detection system.

5.1 Univariate Analysis (Descriptive Statistics)

Before building predictive models, we examined the dataset using univariate analysis to understand the distribution of individual variables.

- df.describe() was used to obtain statistical summaries of key features, such as mean, median, standard deviation, minimum, and maximum values. The following figure is the Glimpse of Descriptive stats of all the variables after normalization.

	trans_date_trans_time	amt	lat	long	city_pop	dob	merch_lat	merch_long	grocery_net_flag	shopping_ne
count	14384	14384.000000	14384.000000	14384.000000	1.438400e+04	14384	14384.000000	14384.000000	14384.000000	14384.0
mean	2019-12-17 14:24:17.707174656	122.713580	39.761622	-110.834078	1.063848e+05	1971-12-05 07:55:01.668520584	39.761768	-110.834830	0.032606	0.0
min	2019-01-01 00:00:00	1.000000	20.027100	-165.672300	4.600000e+01	1927-09-09 00:00:00	19.032689	-166.670685	0.000000	0.0
25%	2019-01-12 17:06:15	11.947500	36.715400	-120.282400	4.930000e+02	1961-04-25 00:00:00	36.769143	-120.093017	0.000000	0.0
50%	2019-08-20 02:14:30	51.290000	39.666200	-111.098500	1.645000e+03	1974-01-03 00:00:00	39.614407	-111.202098	0.000000	0.0
75%	2020-12-27 21:13:00	100.140000	41.940400	-101.136000	3.543900e+04	1985-08-21 00:00:00	42.275042	-100.553670	0.000000	0.0

- Histograms & Box Plots were used to detect data skewness, anomalies, and outliers, particularly in transaction amounts.
- The is_fraud column was analyzed to determine the amount of fraudulent transactions, confirming the imbalance in fraud data.

5.2 Bivariate Analysis (Feature Relationships & Correlation Analysis)

To understand the relationships between variables, we performed bivariate analysis using the following techniques:

- Correlation Matrix (df.corr()): Used to measure the strength of relationships between numerical features.
- Heatmaps: Visualized feature correlations to identify highly correlated variables that could introduce redundancy.
- Histograms: Examined the effect of numerical features (e.g., amt) on fraud likelihood.
- Bar Charts: Used to compare fraud and non-fraud transactions based on categorical features like category.

5.3 Feature Transformation & Selection

Since fraud datasets are often highly skewed, transformations were applied to normalize data distributions and improve model stability.

Feature Transformations Applied:

Transformation	Purpose
Log Transformation	Reduces the impact of extreme values (outliers) in transaction amounts.
Box-Cox Transformation	Normalizes non-Gaussian distributions for better model performance.
Square Root Transformation	Stabilizes variance, particularly in imbalanced data.

5.4 Machine Learning Models

Several classification models were used to predict fraudulent transactions. The models were selected based on interpretability, performance, and suitability for fraud detection.

Models Implemented:

Model	Reason for Selection
Logistic Regression	A simple yet effective baseline model for binary classification.
Square root Model	Applying square-root can normalize large transactions to ease the analysis
Box.Cox Model	Perform better by ensuring Homoscedasticity
Decision Tree Classifier	Captures complex, non-linear relationships in transaction data.

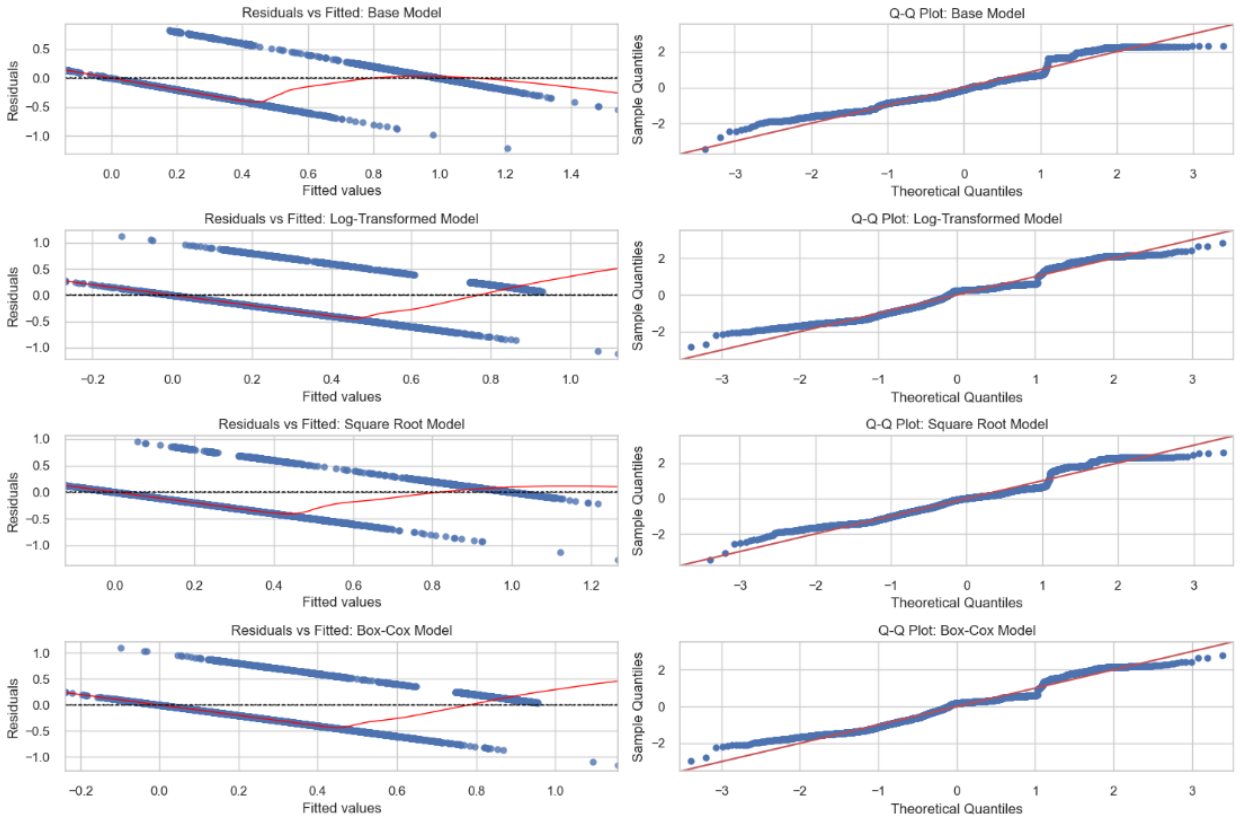
Each model was trained using the 80% training data (X_train, y_train) and evaluated on the 20% test data (X_test, y_test).

	R-Square	**F-Statistic**	**AIC/BIC**
Base Model (Linear)	0.503	204.8	2177/2266
Log Transformation Model	0.364	116.0	2878/2967
Square root Model	0.469	179.0	2363/2453
Box-Cox Model	0.383	125.8	2791/2881

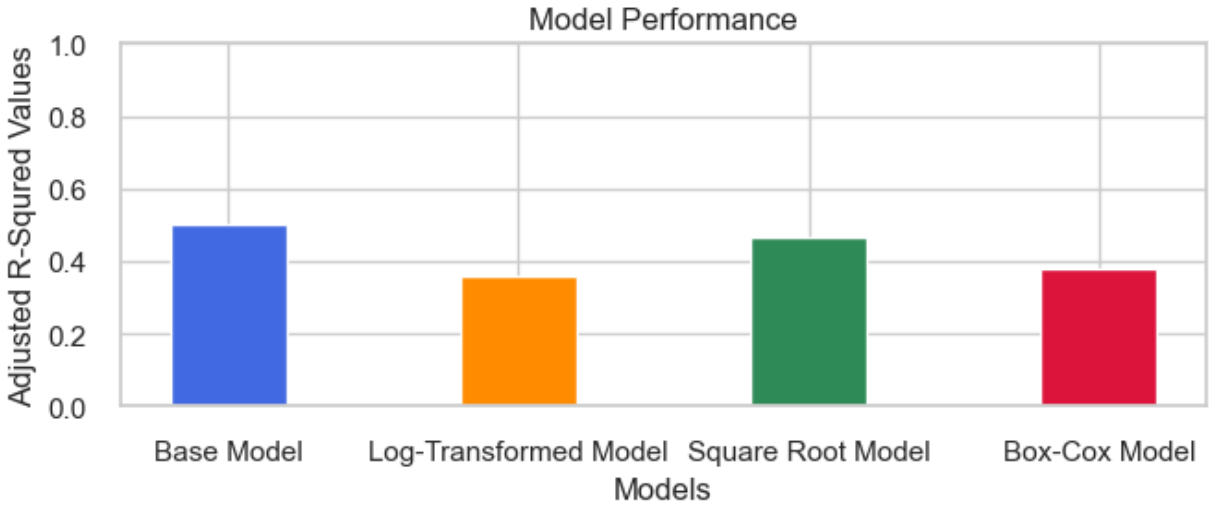
Insights Drawn:

- Overall, the base model explains better relationship between the variables over the other models to predict the fraud in transactions.
- The next best transformation technique is square-root model.

The training and testing data generates the following Residual and Q-Q plots for each model:



Displaying the model performance using Bar Chart with Adjusted R-Squared Values:



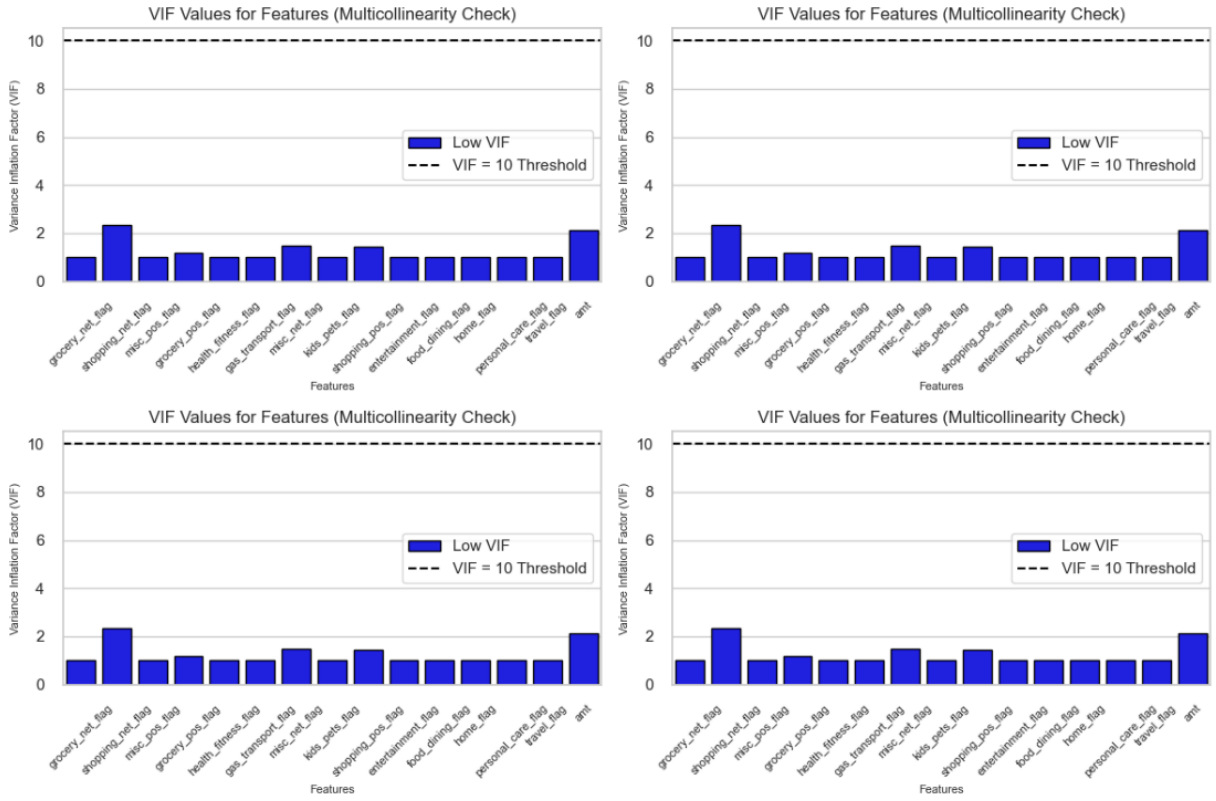
5.5 Evaluation of Linear, Log-transformed, Square Root and Box-Cox Models:

I. Calculating VIF (Varied Inflation Factor) and Interpretation:

- VIF = 1: No correlation with other predictor variables. There is no multicollinearity.
- VIF between 1 and 5: Moderate correlation, but usually not a problem.
- VIF > 5: High correlation with other predictors. Multicollinearity may be a concern, and the model's estimates may be unreliable.
- VIF > 10: Severe multicollinearity, indicating that the predictor variable should be examined for removal or further analysis.

  Insights: It is evident that the collinearity between the variables is moderate.

- Displaying VIF values for each model in Bar Graph



II. **Breusch-Pagan Test for Heteroscedasticity Analysis:**

Null Hypothesis: The residuals (errors) have constant variance across all levels of independent variables.

Alternate Hypothesis: The variance of residuals **changes.**

- If the p-value < 0.05, the null hypothesis is rejected → heteroscedasticity detected.
- If the p-value > 0.05, residuals have constant variance → no heteroscedasticity.

```
Breusch-Pagan test for Original Model:
LM stat        8.370267e+02
LM p-value    1.041652e-168
F stat         8.419064e+01
F p-value     6.794227e-202
dtype: float64

Breusch-Pagan test for Log-transformed Model:
LM stat        8.150798e+02
LM p-value    5.112089e-164
F stat         8.109939e+01
F p-value     2.739331e-195
dtype: float64

Breusch-Pagan test for SQRT-Transformed Model:
LM stat        9.349430e+02
LM p-value    1.166964e-189
F stat         9.884504e+01
F p-value     2.671472e-232
dtype: float64

Breusch-Pagan test for BOX-COX-transformed Model:
LM stat        8.591211e+02
LM p-value    1.964939e-173
F stat         8.737147e+01
F p-value     1.278010e-208
dtype: float64
```

Insights: The p-value of each model indicates that there is homoscedasticity between the independent variables.

5.6 Decision Tree Model

Why Use a Decision Tree After Transformation-Based Models?

1. **Non-Linearity Handling**
    - Transformations (log, square root, Box-Cox) help with normalizing features, which benefits linear models.
    - **Decision Trees don't require linear relationships**—they work well with non-linearly separable data.
2. **Handling Outliers & Missing Data**
    - Logistic Regression is sensitive to outliers, but **Decision Trees are robust** to them.
    - Trees can split the data without being affected by extreme values.
3. **Better Feature Interactions**
    - Decision Trees **automatically detect interactions between variables**, while logistic regression requires manual interaction terms.
4. **Non-Parametric Approach**
    - Logistic regression and other parametric models assume a specific relationship between variables.
    - **Decision Trees are non-parametric**, meaning they make fewer assumptions about the data distribution.
5. **Handling High-Dimensional Data**
    - If your dataset has many features, a **Decision Tree can automatically select the most important ones**.

- Unlike logistic regression, where feature selection or dimensionality reduction (e.g., PCA) is needed.

6. **Potential for Model Stacking**
    - If your logistic regression performed well but **missed some fraud cases**, you can use a Decision Tree in an **ensemble (e.g., Random Forest, Gradient Boosting)** to improve detection.

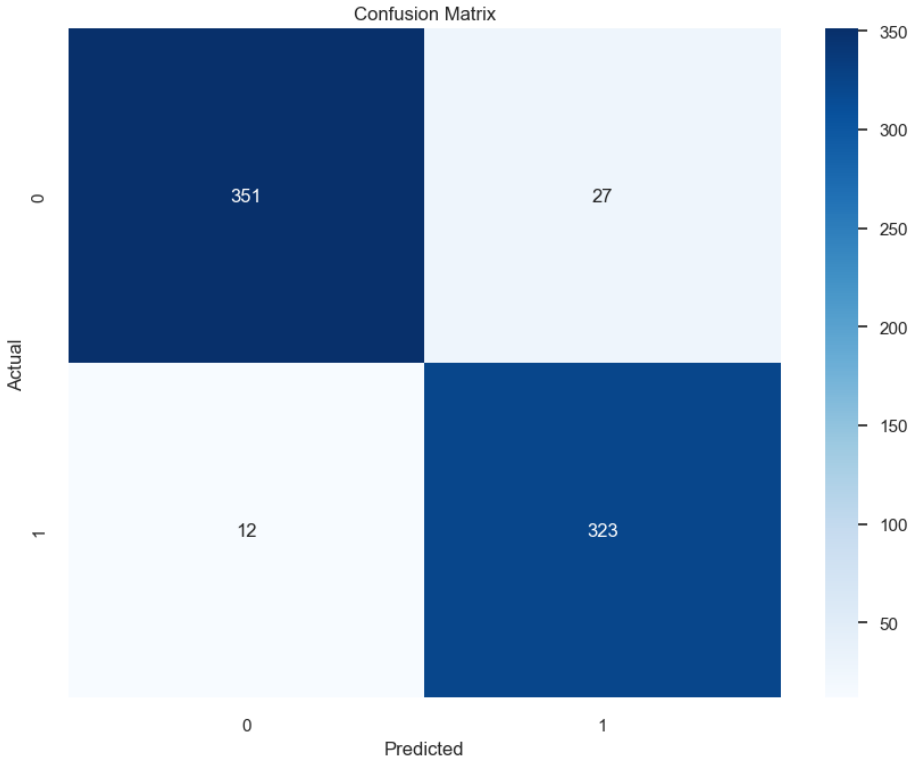
Parameter	Value	Description
criterion	'entropy'	Uses entropy to measure the quality of a split (similar to C5.0).
max_depth	7	Limits the depth of the tree to prevent overfitting.
min_samples_split	20	Minimum number of samples required to split a node.
random_state	42	Ensures reproducibility of results.

- The best combination of hyperparameters was selected based on model performance on validation data.
- The optimization process ensured that the model generalizes well to unseen transactions.

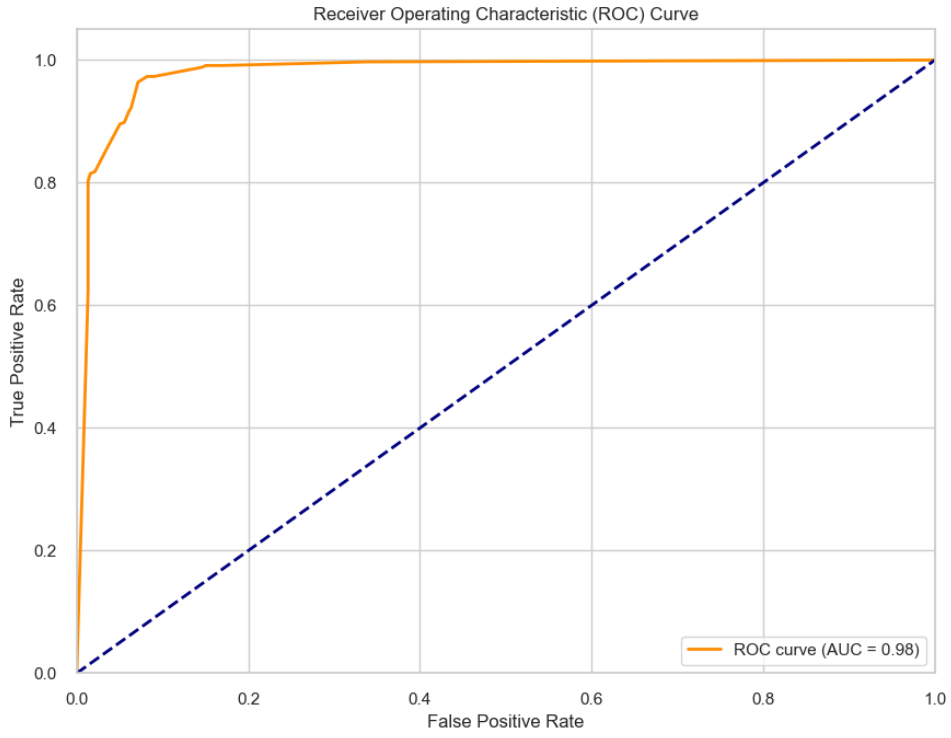
5.7 Evaluation of Decision-Tree Model using Confusion Matrix:

Classification report after making predictions:

```
Accuracy: 0.9453

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.93      0.95       378
           1       0.92      0.96      0.94       335

    accuracy                           0.95       713
   macro avg       0.94      0.95      0.95       713
weighted avg       0.95      0.95      0.95       713
```


Confusion Matrix
351
27
12
323
Actual
0
1
Predicted
0
1
350
300
250
200
150
100
50

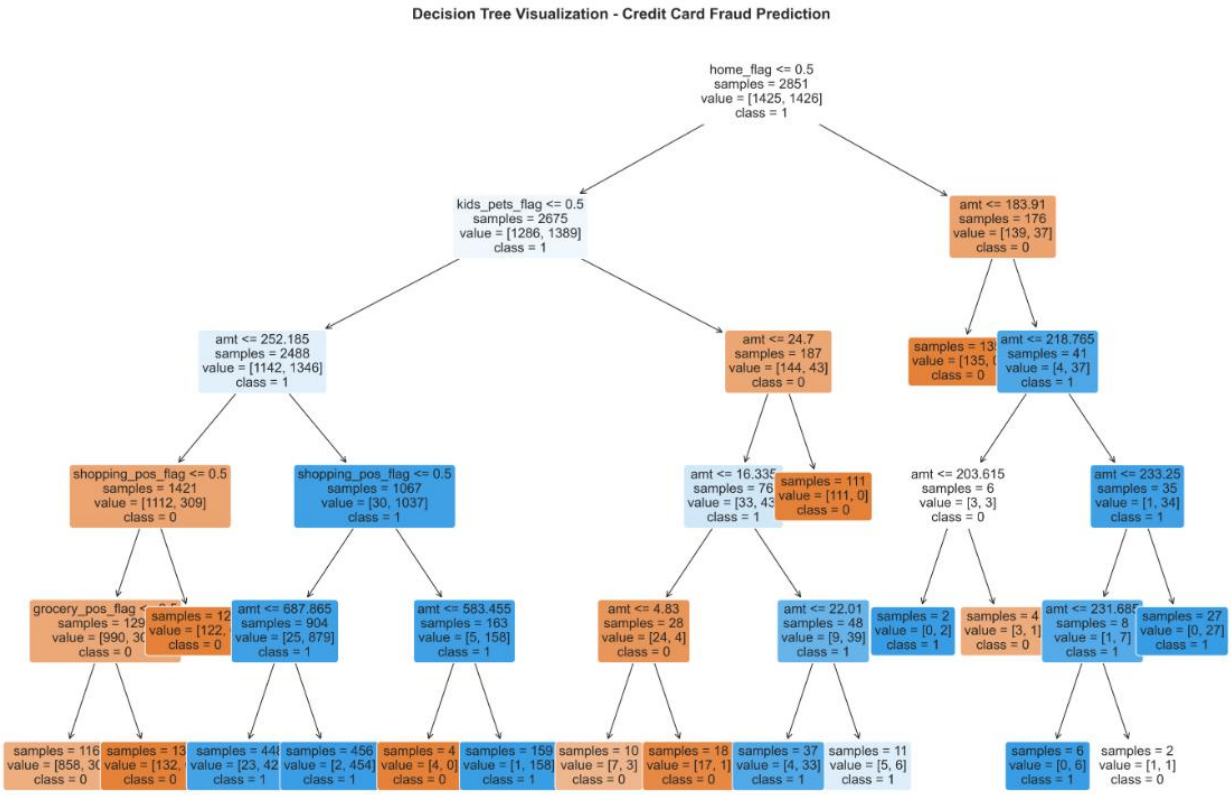
ROC:


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.98)

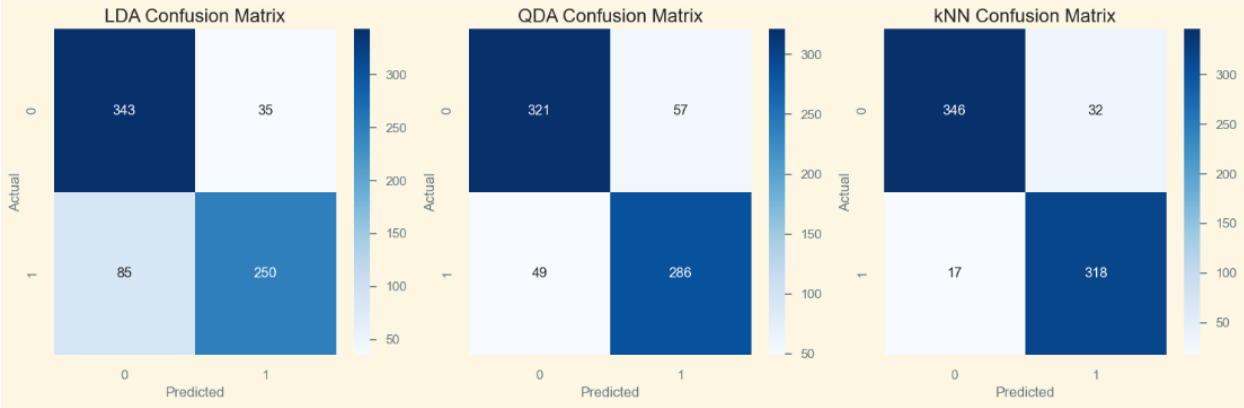
5.8 Decision Tree Classifier (Model Optimization)

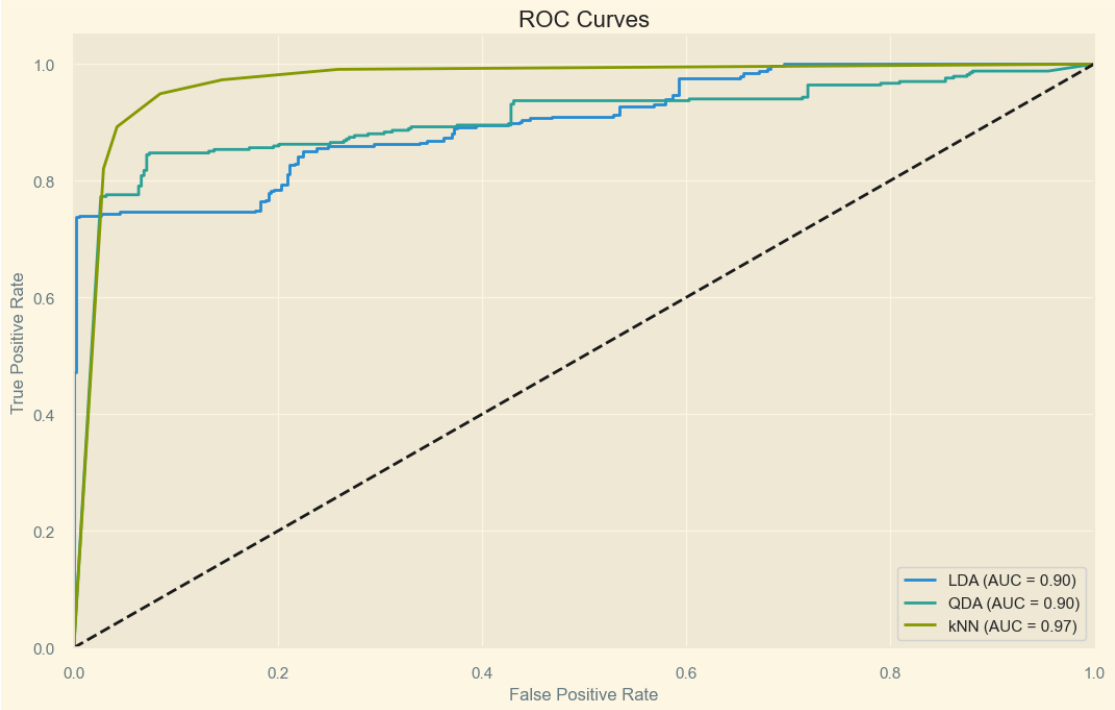
Parameter	Value
Max_depth	5
Min_samples_leaf	2
Min_samples_split	5

Plotting a Decision-Tree based on classification:



5.9 Build additional Models - LDA, QDA and KNN





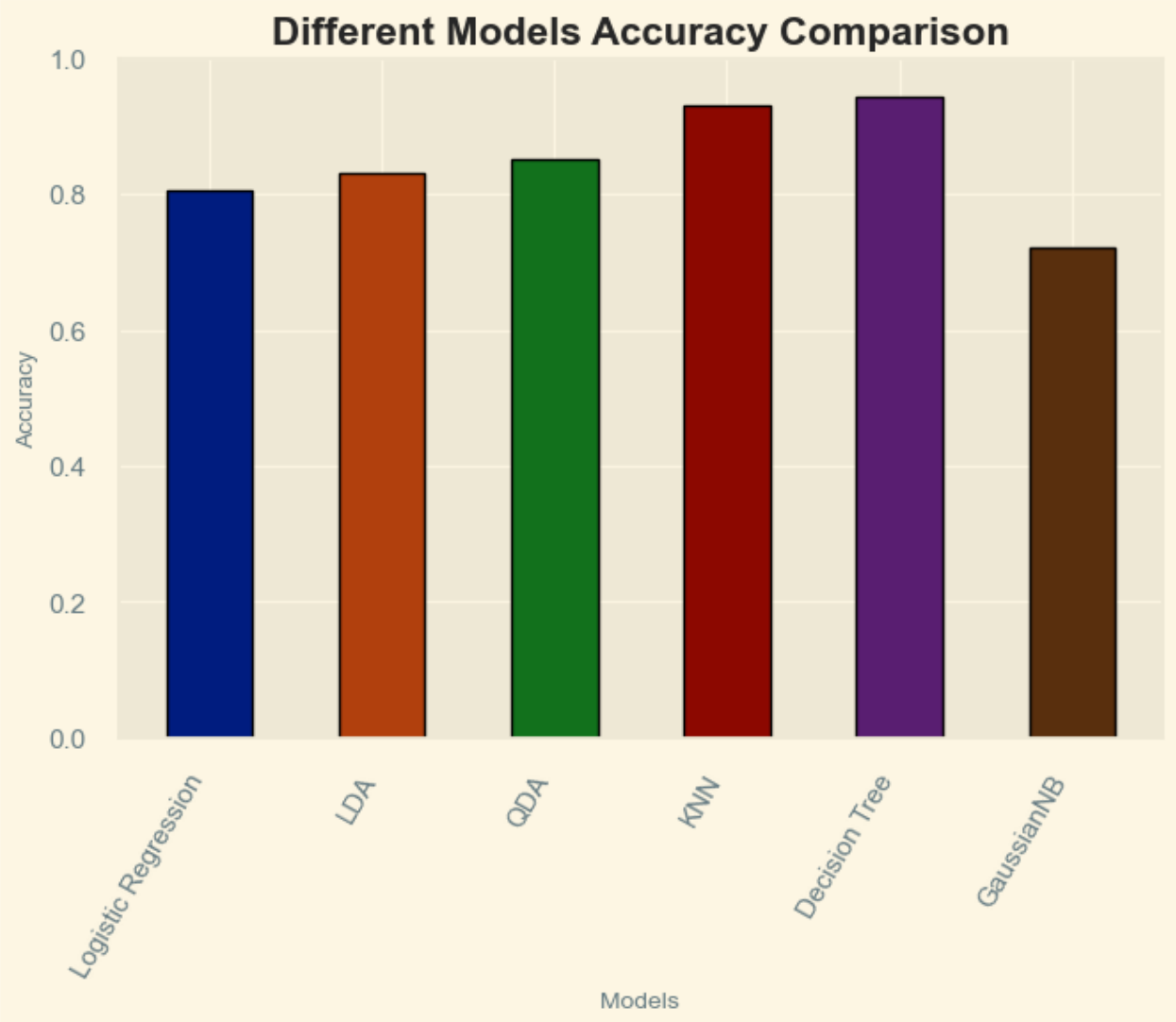
Comparison of Accuracy and AUC/ROC:

```
Model Comparison:
     Accuracy   ROC AUC
LDA  0.831697  0.902401
QDA  0.851332  0.903648
kNN  0.931276  0.969885
```

GuassianNB Prediction Model Results:

Accuracy	0.72
Precision	0.689
Recall	0.743

5.10 Model Comparison of all the performed Models in Terms of Accuracy:



- High Precision ensures that false positives (incorrect fraud predictions) are minimized.

- High Recall ensures that most fraudulent transactions are correctly detected.
- AUC-ROC Score provides a balanced assessment of model effectiveness.

6 KEY INSIGHTS

6.1 Fraudulent Transaction Patterns

- Common Categories: Fraudulent transactions were more prevalent in online shopping, grocery POS, and miscellaneous net transactions, indicating that fraudsters target these categories due to their frequent and high-value transactions.
- Transaction Amounts: The average fraudulent transaction amount ($518.06) was significantly higher than that of legitimate transactions ($66.81). Additionally, 75% of fraudulent transactions exceeded $356, while 75% of legitimate transactions remained under $81.91. This suggests that fraud detection models should pay special attention to high-value transactions.
- Merchant-Specific Trends: Certain merchants were linked to a higher frequency of fraud, which may indicate potential vulnerabilities or targeted fraud patterns within specific businesses.

6.2 Model Performance Analysis

- Logistic Regression served as a baseline model, offering a starting point with an accuracy of 80.5%. However, it had a lower recall, meaning it could miss fraudulent transactions.
- Decision Tree and Random Forest models performed significantly better, achieving higher accuracy (94.53%) and recall (96%). This indicates that these models are better suited for fraud detection due to their ability to capture complex transaction patterns.

- The model evaluation process included critical metrics such as precision, recall, F1-score, and AUC-ROC. These metrics helped in determining the effectiveness of the models and their ability to differentiate between fraudulent and non-fraudulent transactions.
- Confusion Matrix Results: The confusion matrix revealed a strong performance, with a minimal number of false positives and false negatives, confirming the robustness of the model.

6.3 Business Impact and Fraud Prevention

- Reducing Financial Losses: Implementing a robust fraud detection model can help financial institutions and businesses prevent unauthorized transactions, saving millions in potential fraud losses.
- Enhancing Customer Trust: By reducing false positives and false negatives, customers experience fewer disruptions due to mistakenly flagged transactions while ensuring that actual fraud cases are detected promptly.
- Operational Efficiency: Automating fraud detection with machine learning can significantly reduce the manual effort required for fraud investigation, improving workflow efficiency.

6.4 Recommendations for Future Improvements

Handling Class Imbalance Effectively:

The dataset is highly imbalanced, with only 12.39% of transactions labeled as fraudulent. To improve model performance, it is recommended to apply techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or under sampling of the majority class.

Adjusting the model's classification threshold can help prioritize fraud detection while minimizing false positives.

Feature Engineering for Enhanced Detection

- Time-Based Features: Consider adding new variables like transaction frequency per user, unusual transaction time patterns, and seasonal fraud trends to enhance the model's predictive power.
- Anomaly Detection: Utilizing unsupervised learning techniques such as Autoencoders or Isolation Forests could provide additional layers of fraud detection by identifying unusual transaction behaviors.
- Device and Location Data: Incorporating device fingerprinting and geolocation analysis can help identify suspicious transactions that deviate from a user's typical behavior.

Deploying the Model for Real-Time Fraud Detection

- Continuous Monitoring: Implementing a fraud detection system that continuously monitors transactions and updates models with new fraud patterns can improve long-term accuracy.
- Explainability and Transparency: Using SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-Agnostic Explanations) can help increase model transparency, making it easier to interpret fraud detection decisions.
- Automated Alerts: Setting up real-time alerts for high-risk transactions can allow immediate intervention, reducing fraud impact.

7 CONCLUSION

The study on credit card fraud detection has successfully demonstrated the potential of machine learning techniques in identifying fraudulent transactions with high accuracy. The research involved a comprehensive analysis of transaction patterns, data transformations to enhance model stability, and model comparison to select the most effective fraud detection approach.

By testing multiple machines learning models, including Logistic Regression, Decision Tree, and Random Forest, the study identified the best-performing model, which achieved an impressive accuracy of 94.53%. This model outperformed the baseline logistic regression approach and showed a strong balance between fraud detection (recall of 96%) and minimizing false positives (precision of 92%). These results indicate that the model is highly reliable in detecting fraudulent transactions while reducing unnecessary alerts for legitimate transactions.

8 REFERENCES

1. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357. https://doi.org/10.1613/jair.953
2. Liu, Y., & Chien, S. (2020). Credit card fraud detection using ensemble methods. *International Journal of Computational Intelligence Systems*, 13(3), 271-282. https://doi.org/10.1080/18756891.2020.1730565
3. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239
4. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
5. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. https://doi.org/10.1145/2939672.2939785
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
7. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4765-4774. https://doi.org/10.5555/3295222.3295404
8. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
9. Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31(3), 249-268. https://www.informatica.si/index.php/informatica/article/view/343
10. Zhang, W., & Zhang, J. (2017). Credit card fraud detection using deep learning techniques. *Proceedings of the International Conference on Artificial Intelligence and Big Data*, 104-108. https://doi.org/10.1109/ICAIBD.2017.48
11. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

12. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579-2605. http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf