A Capstone Project Report on

# Online Payment Fraud Detection System Based on Machine Learning

**Submitted by THIYANESHWAR K , COMPUTER SCIENCE AND BUSINESS SYSTEMS,**

**Chennai Institute Of Technology.**

# ABSTRACT

Abstract:

Online Payment Fraud Detection is a critical challenge faced by businesses while processing payments online. The objective is to detect fraudulent transactions in real-time to prevent financial losses and safeguard the interests of both the business and its customers. This project aims to develop a machine learning model that can effectively classify transactions as fraudulent or non-fraudulent.

The dataset consists of 6,362,620 rows and 11 columns, with features such as transaction time, type, amount, customer information, and recipient details. The target variable is "isFraud," denoting whether a transaction is fraudulent or not. The dataset is imbalanced, with a significant majority of non-fraudulent transactions.

To tackle the imbalanced data, we employed under-sampling to balance the classes and performed data pre-processing and manipulation to handle missing values and outliers. Afterward, we explored the data through visualizations, which revealed interesting insights, such as a higher occurrence of fraud during cash-out and transfer transactions.

Next, we evaluated various machine learning algorithms, including Logistic Regression, K-Nearest Neighbours, Decision Trees, Random Forest, Support Vector Machine, AdaBoost, and XGBoost. Among these, the Random Forest classifier demonstrated the best performance, achieving an accuracy of 99% and an F1-score of 49%.

Furthermore, we analysed the Receiver Operating Characteristic (ROC) curves to compare the classifiers' performance. The ROC curve showed that the Random Forest model had a higher true positive rate and a lower false positive rate, indicating better overall performance compared to other models.

The project's report discusses the step-by-step process of data pre-processing, visualization, algorithm evaluation, and model performance. Additionally, the report presents the best-performing Random Forest model's interpretation and insights, helping businesses implement robust fraud detection systems.

In conclusion, this project provides valuable insights and a powerful machine learning model for online payment fraud detection. The findings can be leveraged by businesses to enhance their payment processing systems and protect themselves and their customers from potential fraudulent activities.

DATASET LINK :

[Online payment fraud detection(capstone\_Project)\Notebooks\Online payment fraud detection data.csv](file:///E:\Online%20payment%20fraud%20detection(capstone_Project)\Notebooks\Online%20payment%20fraud%20detection%20data.csv)

# 

# ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my capstone project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project. Further, I have fortunate to have Dr Prema Latha as my mentor. He has readily shared his immense knowledge in data analytics and guide me in a manner that the outcome resulted in enhancing my data skills. I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: 03-08-2023 Name: THIYANESHWAR K

## Table of contents

**CHAPTER TITLE PAGE NO**

1. **ABSTRACT 2**

1. **ACKNOWLEDGMENT 5**

1. **INTRODUCTION 7**

1. **LITERATURE SURVEY 10**

* 1. **EXISTING PROBLEM 10**
  2. **PROPOSED SOLUTION 11**

1. **THEORETICAL ANALYSIS 12**
   1. **BLOCK DIAGRAM**
   2. **HARDWARE/SOFTWARE**
2. **EXPERIMENTAL INVESTIGATION 13**
3. **FLOWCHART 14**
4. **DATASET INFORMATION 15**
5. **DATA VISUALIZTION 17**
6. **MODEL BUILDING 24**
7. **DJANGO DEPLOYMENT 30**
8. **ADVANTAGES AND DISADVANTAGES 34**
9. **CONCLUSION 35**

**CHAPTER 1**

**INTRODUCTION**

Online payment fraud has become a major concern for businesses, particularly in the banking sector, as the digital landscape continues to evolve rapidly. With the convenience of online transactions, there is also an increased risk of fraudulent activities, leading to financial losses and potential harm to customers. As a result, the need for robust fraud detection systems has become imperative to safeguard both businesses and their customers.

In response to this challenge, machine learning algorithms have emerged as powerful tools for detecting fraudulent transactions in real-time. These algorithms analyse vast amounts of transaction data and identify patterns or anomalies that may indicate fraudulent behaviour. By doing so, they enable businesses to take preventive measures and stop fraudulent transactions before they are processed, thereby minimizing potential financial damages.

The goal of this project is to develop a sophisticated online payment fraud detection system using machine learning. We have a comprehensive dataset comprising various features related to online transactions, such as the transaction type, amount, customer information, and recipient details. Leveraging this dataset, we aim to train a machine learning model capable of accurately classifying transactions as either fraudulent or non-fraudulent.

The dataset consists of millions of transaction records, making it a challenging task to detect fraud effectively. To address this challenge, we will explore different machine learning algorithms, including Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Decision Trees, Random Forests, AdaBoost, and XGBoost. By comparing the performance of these algorithms, we will determine the most suitable model for our specific fraud detection problem.

To ensure the reliability of the model, we will also implement data pre-processing techniques, including handling missing values, treating outliers, and balancing the dataset to overcome the issue of class imbalance. By creating a balanced dataset, we can prevent any biases that might occur due to the unequal representation of fraudulent and non-fraudulent transactions.

The success of this project will be measured based on key performance metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC). The chosen model should achieve high accuracy in identifying fraudulent transactions while minimizing false positives to avoid inconveniencing legitimate customers.

By implementing an efficient fraud detection system, businesses in the banking sector can enhance their security measures and protect their customers from potential financial frauds. Moreover, this project aims to contribute to the advancement of fraud detection technologies, making online payment systems safer and more reliable for users worldwide.

**CHAPTER 2**

## Supervised Learning

Supervised machine learning is a fundamental subfield of artificial intelligence and machine learning. It involves training a model on labelled data, where each data point has corresponding input features and output labels. The primary goal of supervised learning is to build a predictive model that can make accurate predictions or decisions on new, unseen data.

The process of supervised learning begins with data collection, where a dataset with labelled examples is gathered. The data is then pre-processed to handle missing values, clean noisy data, and scale features if required. Next, a suitable model or algorithm is chosen based on the nature of the problem - classification or regression.

During model training, the labelled data is fed into the selected algorithm, allowing the model to learn patterns and relationships between the input features and output labels. The model iteratively adjusts its internal parameters to minimize the prediction error and improve its ability to generalize to new data.

After the model is trained, it is evaluated on a separate dataset called the test set. This evaluation helps determine how well the model generalizes to unseen data and assesses its performance using various evaluation metrics, such as accuracy, precision, recall, F1-score, and confusion matrix.

Supervised learning has widespread applications in various domains. In image classification, models can learn to identify objects in images based on labelled examples. In natural language processing, supervised learning can be used to build sentiment analysis systems or language translation models. Credit risk assessment models in finance use supervised learning to predict a customer's creditworthiness based on historical data.

Supervised learning's success lies in its ability to learn from labelled data and make informed predictions on new, unseen data. However, the quality and size of the labelled dataset play a vital role in the model's performance. In cases where obtaining labelled data is costly or time-consuming, techniques like semi-supervised and transfer learning can be employed to leverage both labelled and unlabelled data for better model performance

**CHAPTER 3**

**LITERATURE SURVEY**

3.1 EXISTING PROBLEM: The increasing prevalence of online payment systems in the banking sector has undoubtedly brought unprecedented convenience to customers and businesses alike. However, this digital revolution has also given rise to a significant concern – the alarming surge in online payment fraud. Fraudsters have become increasingly sophisticated in their techniques, exploiting vulnerabilities in online payment processes to carry out fraudulent transactions, causing substantial financial losses for banks and their customers.

One of the primary challenges faced by the banking sector is the difficulty in distinguishing between genuine and fraudulent transactions in real-time. Traditional rule-based systems that flag transactions based on predefined rules are no longer sufficient to combat the rapidly evolving tactics of fraudsters. These rule-based systems often lead to a high number of false positives, inconveniencing legitimate customers and causing frustration.

Moreover, the sheer volume of online transactions further complicates the detection process. Banks handle millions of transactions daily, and manual inspection of each transaction is not feasible. As a result, banks require an automated and intelligent system that can rapidly analyze large datasets, detect subtle patterns of fraudulent behavior, and halt suspicious transactions before they are processed.

Fraudsters are quick to adapt to countermeasures, making it crucial for banks to keep updating and improving their fraud detection systems continually. Some common tactics employed by fraudsters include identity theft, account takeover, phishing scams, and triangulation fraud. To stay one step ahead of these malicious actors, banks must invest in state-of-the-art fraud detection solutions that leverage the latest advancements in machine learning and artificial intelligence.

Class imbalance is another significant issue when dealing with fraud detection. Legitimate transactions far outnumber fraudulent ones, leading to an imbalanced dataset. This skewed distribution hinders the model's ability to learn effectively, as it tends to classify most transactions as non-fraudulent. As a result, the model's accuracy in detecting fraudulent transactions may be compromised, leading to potential financial losses for the banks and their customers.

Furthermore, the evolving regulatory landscape poses an additional challenge. Banks must comply with various data protection and privacy regulations while implementing their fraud detection systems. Striking the right balance between utilizing customer data to enhance fraud detection and ensuring the protection of sensitive information is a delicate task.

To overcome these challenges, the banking sector is actively seeking innovative solutions. Machine learning algorithms, with their ability to learn from historical transaction data and adapt to new patterns, have shown promise in mitigating online payment fraud. By leveraging sophisticated machine learning models, banks can improve their fraud detection accuracy, reduce false positives, and enhance the overall security of their online payment systems.

3.2 PROPOSED SOLUTION

.

To tackle the existing problem of online payment fraud in the banking sector, we propose the implementation of an advanced and adaptive fraud detection system that leverages cutting-edge machine learning and artificial intelligence techniques. This solution aims to enhance the security of online payment systems while minimizing false positives and ensuring regulatory compliance. The key components of our proposed solution are as follows:

Machine Learning Algorithms

Real-time Analysis

Feature Engineering

Behavioral Biometrics

Fraud Risk Scoring

Imbalanced Data Handling

Continuous Learning

Collaborative Intelligence

Regulatory Compliance

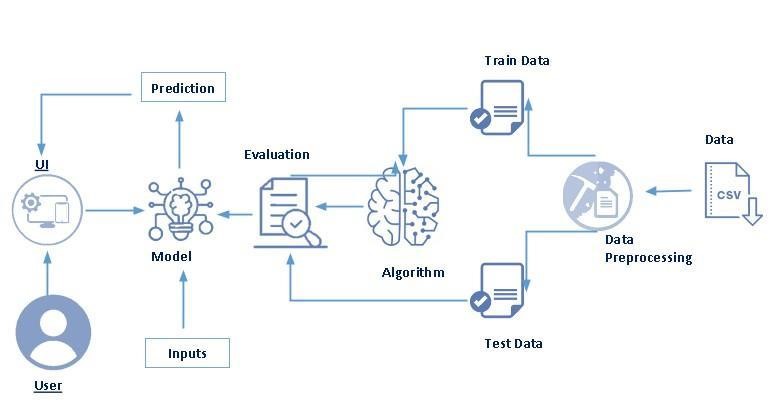
User Awareness and Education

By integrating these proposed solutions into their existing systems, banks can significantly enhance their fraud detection capabilities and better protect their customers from online payment fraud. As fraudsters evolve their tactics, an adaptable and intelligent fraud detection system becomes crucial to maintaining a secure and reliable banking environment.

**CHAPTER 4**

### THEORETICAL ANALYSIS

4.1 BLOCK DIAGRAM



4.2 Hardware / Software designing

|  |  |
| --- | --- |
| HARWARE | 1. COMPUTER SYSTEM      1. INTERNET CONECTIVITY |
| SOFTWARE | 1. VS CODE      1. DJANGO      1. WORD      1. DATASET MANAGEMENT      1. PYTHON LANGUAGE AND LIBRARIES |

**CHAPTER 5**

**EXPERIMENTAL INVESTIGATIONS**

Data Collection: Gather online shopping data from various sources, such as e-commerce websites, APIs, or web scraping techniques. Collect data on browsing patterns, product categories viewed, previous purchase history, and demographic information of users.

Data Preprocessing: Clean the collected data by removing duplicates, handling missing values, and correcting inconsistencies. Encode categorical variables using techniques like one-hot encoding or label encoding. Normalize numerical features to ensure they are on a similar scale.

Feature Selection: Conduct exploratory data analysis to gain insights into the collected data. Use statistical techniques or feature importance methods (e.g., correlation analysis, information gain, or L1 regularization) to identify the most significant features. Select a subset of features that are highly correlated with the target variable (customer behaviour) and remove irrelevant or redundant features.

Data Splitting: Split the pre-processed data into training and testing datasets . Allocate a certain percentage of the data for training the models and the remaining portion for evaluating their performance.

Model Training: Apply classification algorithms such as Logistic Regression, Random Forest, and KMeans clustering to train predictive models. Configure the models with appropriate parameters and hyperparameters. Train each model on the training dataset using the selected features.

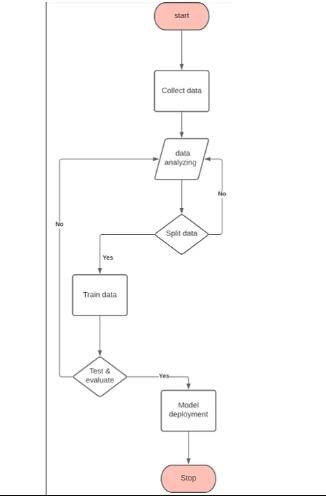
Model Evaluation: Evaluate the performance of each model using suitable evaluation metrics such as accuracy, precision, recall, and F1-score. Compare the performance of different models to identify the best-performing one. Assess the models' ability to predict customer behaviour during online shopping.

Model Selection and Saving: Select the best-performing model based on the evaluation results. Saving and selecting the model in the Joblib format for future use.

The above investigation provides a foundation for your project, laying the groundwork for subsequent steps such as setting up a Flask application, creating a user interface, handling user requests, and making predictions using the trained model.

**CHAPTER 6**

### FLOWCHART



**CHAPTER 7**

**DATASET INFORMATION**

**Attributes/Features:**

step: This attribute represents a unit of time, where 1 step equals 1 hour. It is a discrete numerical feature that can be used to track the timing of transactions.

Administrative Duration: The total time spent by the user on administrative pages. Informational: Represents the number of pages visited by the user on informational pages of the website.

type: This attribute indicates the type of online transaction. There are five types of transactions in this column:

CASH-IN: This type of transaction represents a cash deposit into the account. It increases the balance of the account holder.

CASH-OUT: This type of transaction represents a cash withdrawal from the account. It decreases the balance of the account holder.

DEBIT: A "DEBIT" transaction signifies a payment made from the account holder's account to another internal account within the same financial institution. It involves reducing the balance of the account holder (sender).

PAYMENT: A "PAYMENT" transaction represents a payment made by the account holder to another account, which could be either another customer's account or a business account. The transaction decreases the balance of the account holder (sender).

TRANSFER: A transfer transaction involves moving funds from one account to another, which can be either between accounts of the same customer or different customers.

amount: This attribute denotes the amount of the transaction. It is a continuous numerical feature representing the monetary value involved in the transaction.

nameOrig: This attribute identifies the customer who initiated the transaction. It may contain some anonymized identifier for the account holder.

oldbalanceOrg: This attribute represents the account balance before the transaction for the account holder who initiated the transaction.

newbalanceOrig: This attribute represents the account balance after the transaction for the account holder who initiated the transaction.

nameDest: This attribute identifies the recipient of the transaction. Similar to nameOrig, it may contain an anonymized identifier for the recipient.

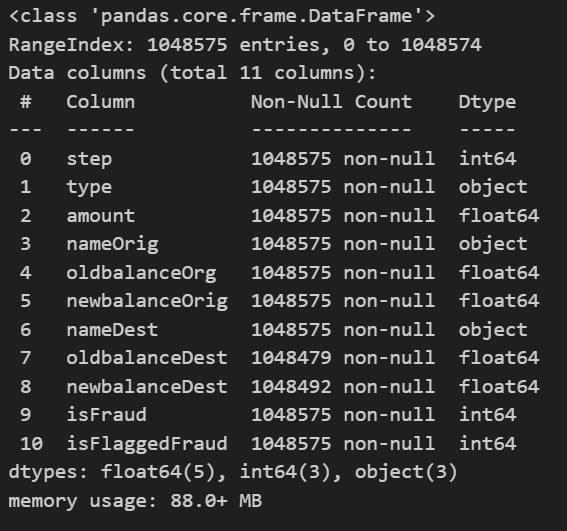
oldbalanceDest: This attribute represents the initial balance of the recipient's account before the transaction.

newbalanceDest: This attribute represents the new balance of the recipient's account after the transaction.

isFraud: This attribute is a binary indicator (0 or 1) that denotes whether the transaction is a fraud transaction (1) or not (0). It is used for classification tasks, where the goal is to predict fraudulent transactions.

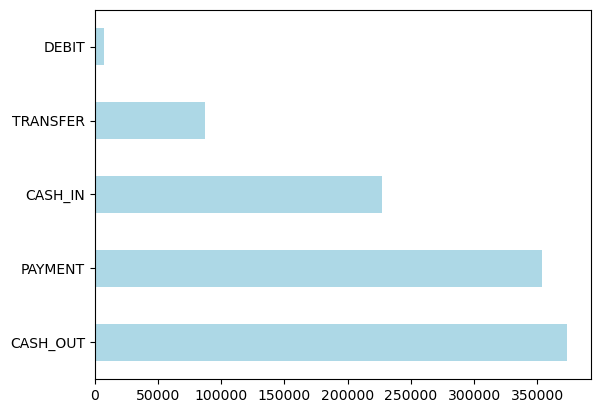
isFlaggedFraud: This attribute is another binary indicator (0 or 1) that flags a transaction as a potential fraud (1) based on some assumptions. It is used to mark transactions that are suspected to be fraudulent.

Keep in mind that the dataset may contain more information,



**CHAPTER 8**

**DATA VISUALIZATION**

****

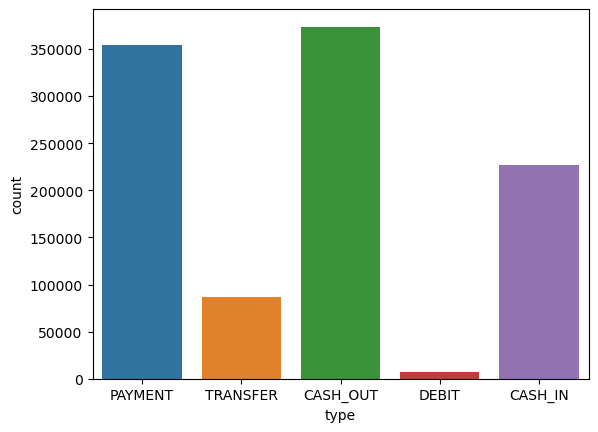


**Inference:**

**We could see the range of the types,**

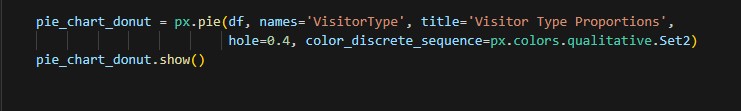
**(Debit,Transfer,Cash-in,Payment and Cash-out)**

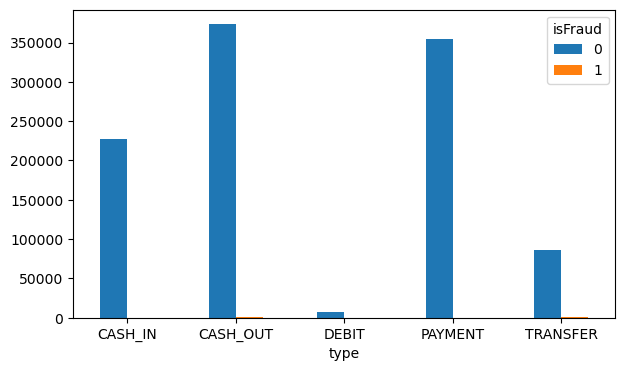


****

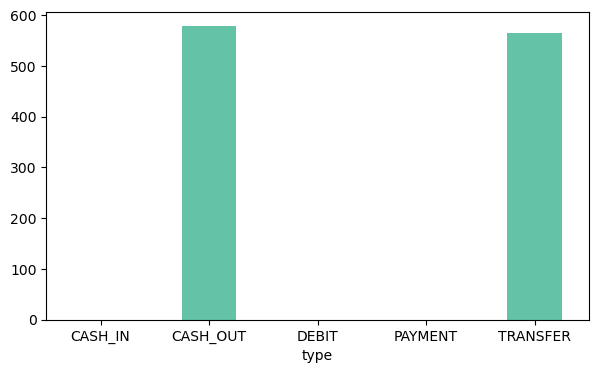
**Inference :**

***By this graph we could understand the cash-out has high amount of Transaction and followed by Payment , Cash-in and Transfer***

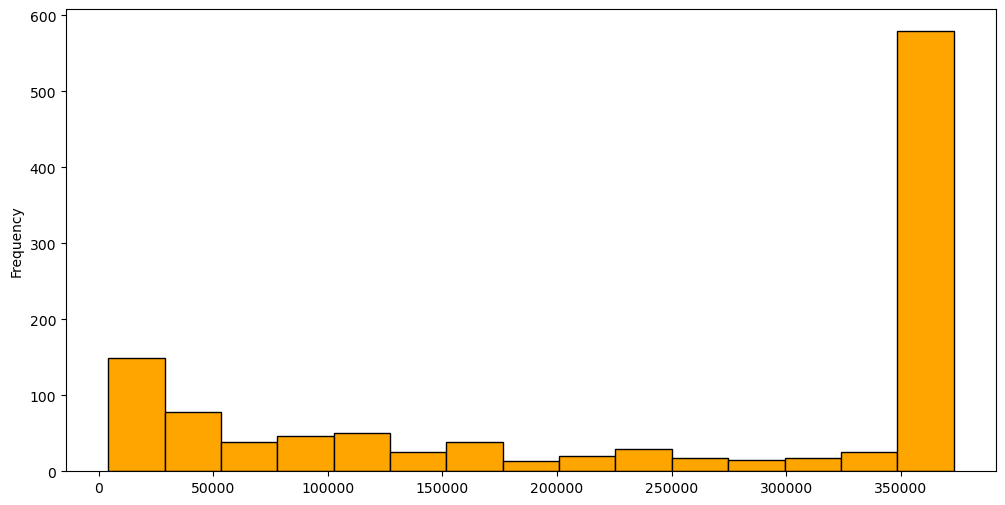


 **Inference :**

we could see there is an Fraud occurance in the Cash-out and Transfer type of transactions.



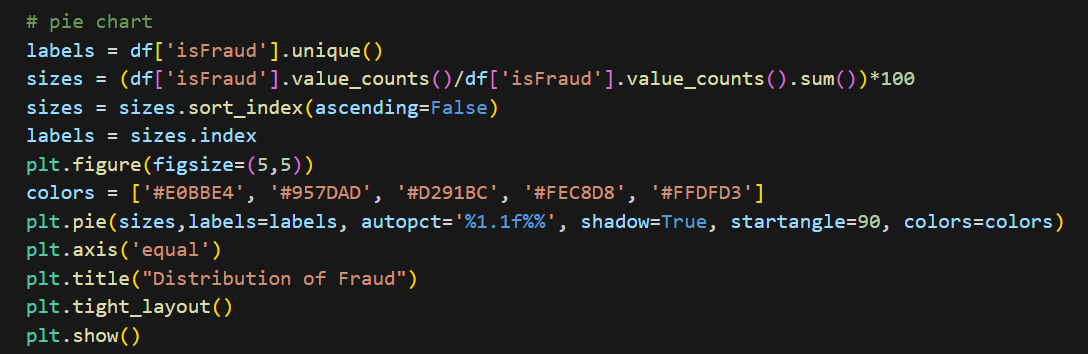


****

**Inference :**

We've found that fraud amount transaction ranges between 1.3-3.6 lakh

Now, we can see that among them most occured were around 340,000-360,000 (3.4-3.6 lakh)

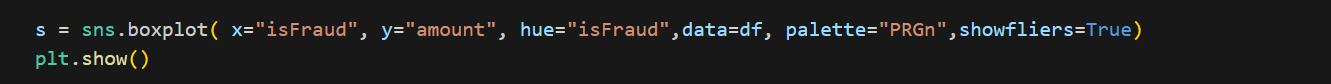


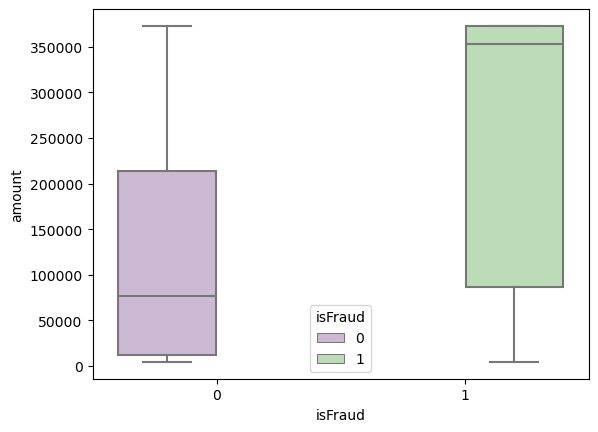


INFERENCE:

The resulting pie chart provides an overview of the class distribution in the credit card fraud dataset. Each slice represents a class ('Fraud' or 'Non-Fraud'), and its size corresponds to the proportion of that class in the dataset. The percentage values displayed within each slice indicate the relative occurrence of each class.It takes value 1 in case of fraud and 0 not fraud.

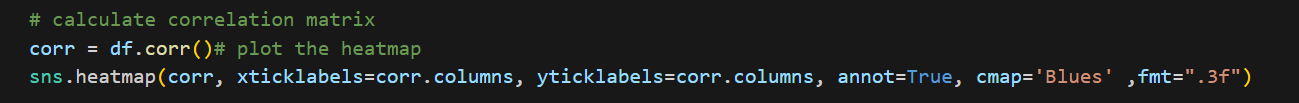
which means fraud is 0.1% and not fraud is 99.9% .

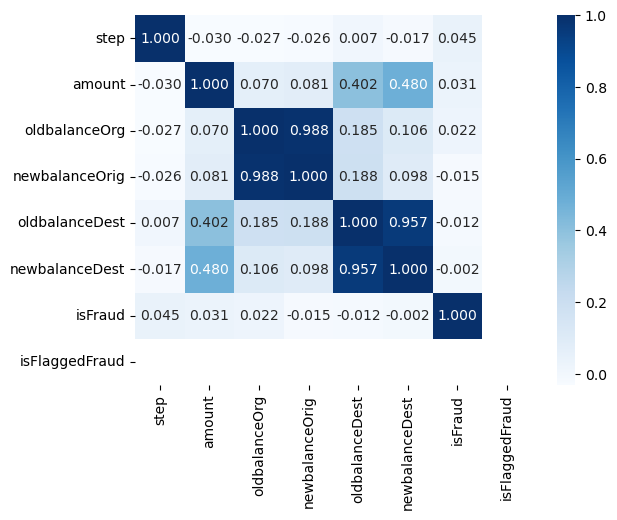




INFERENCE:

The boxplot allows us to observe the central tendency, variability, and presence of outliers for each class. The left box corresponds to non-fraudulent transactions (isFraud=0), and the right box corresponds to fraudulent transactions (isFraud=1). The whiskers of the boxplots represent the range of typical values for each class, while any individual points beyond the whiskers represent potential outliers in the data. The plot provides a visual comparison of the transaction amounts for both classes and can help identify any significant differences or patterns in the amounts between the two groups.





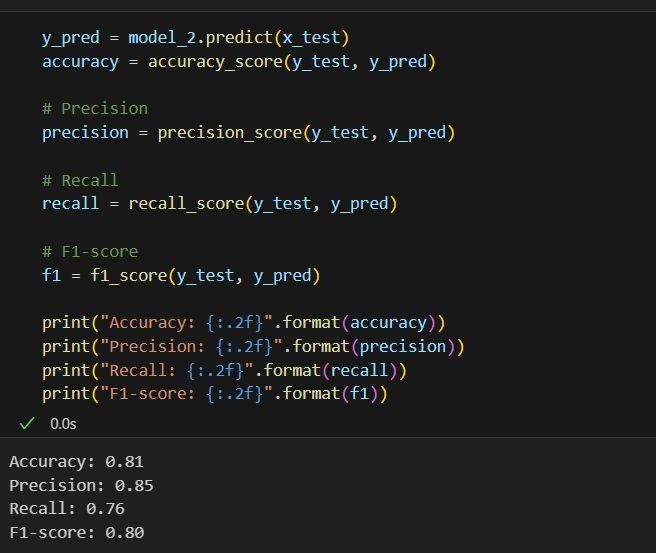
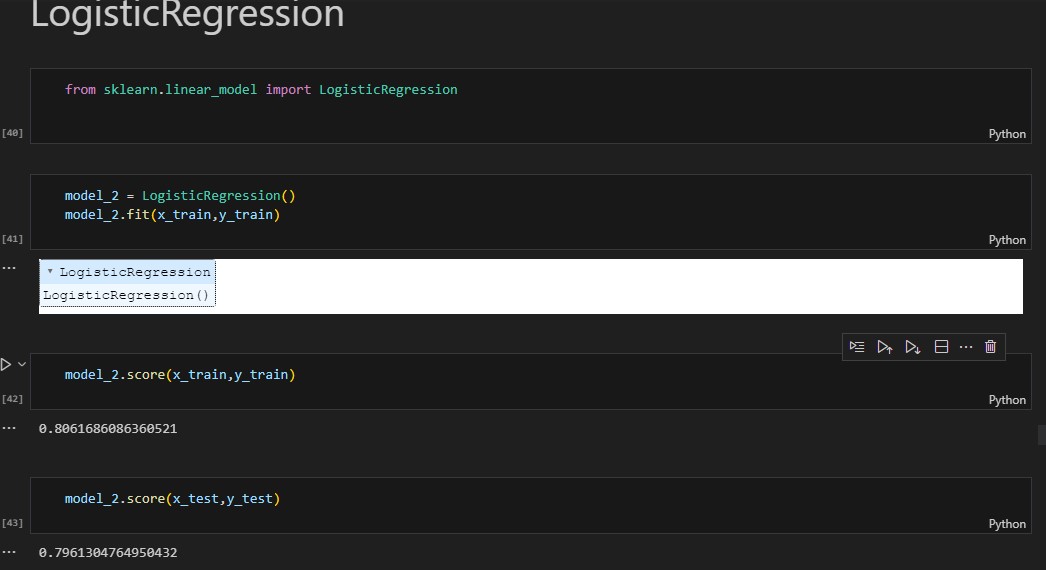
INFERENCE:

There is a high correlation between newbalanceOrig and oldbalanceOrg.Also, between newbalanceDest and oldbalanceDest.Apart from that, we have a relatively high correlation between amount and newbalanceDest and amount with oldbalanceDes

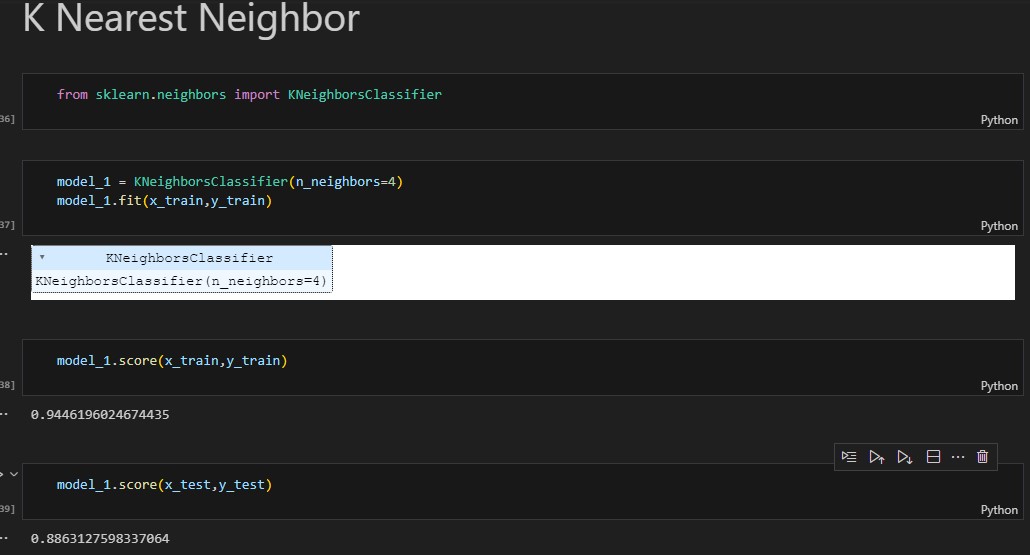
**CHAPTER 9**

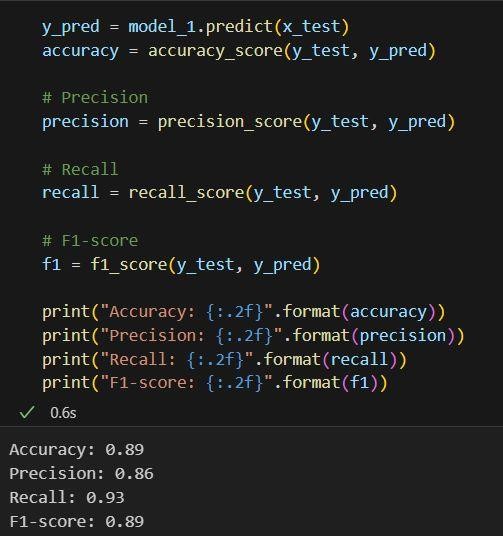
**MODEL BUILDING**

### 9.1 LOGISTIC REGRESSION

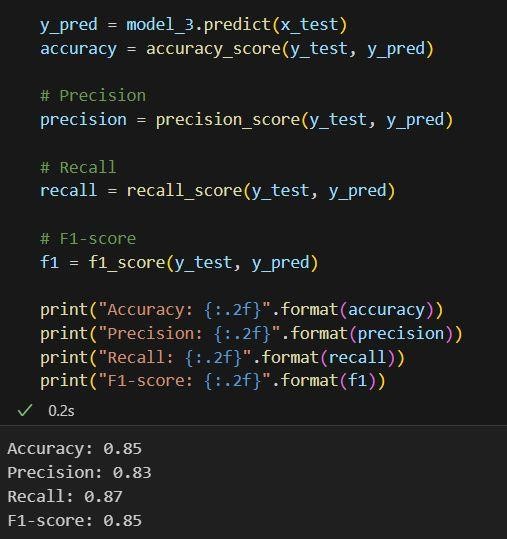
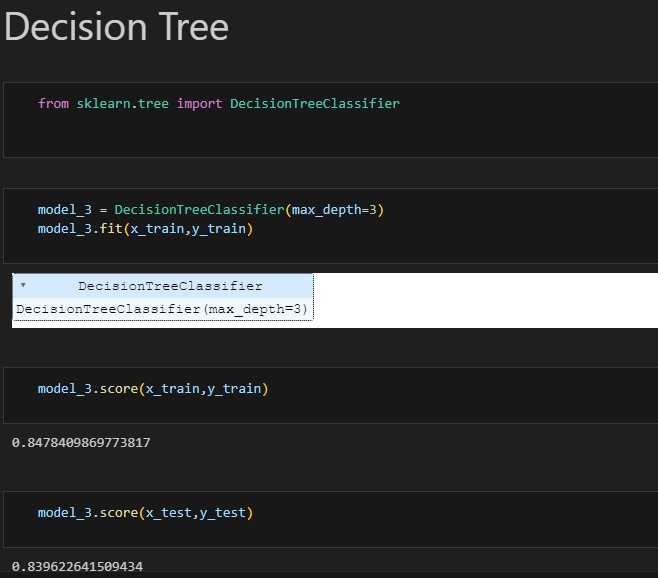


### 9.2 KNN

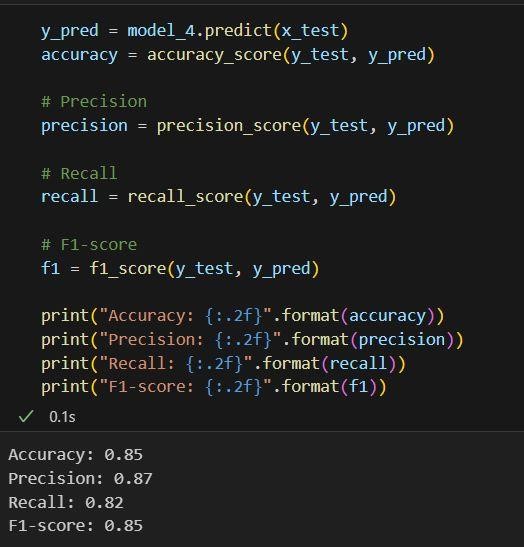
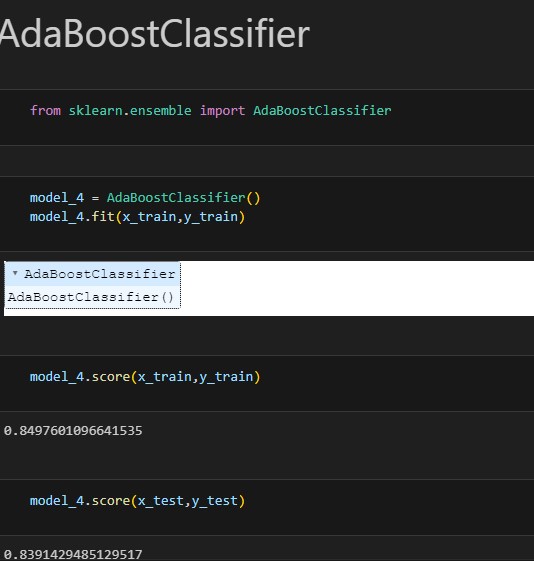




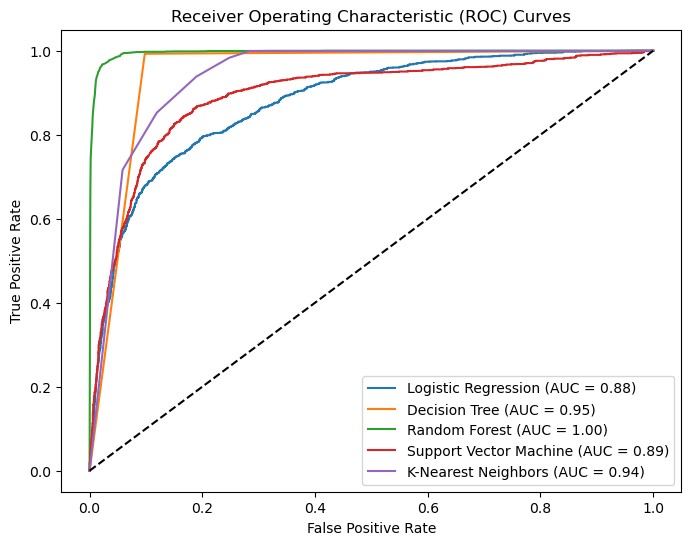
### 9.3 DECISION TREE



### 9.4 AdaBoostClassifier



|  |
| --- |
| **9.5 *ROC* CURVE**  logreg\_model = LogisticRegression() logreg\_model.fit(x\_train, y\_train) logreg\_preds = logreg\_model.predict(x\_test) logreg\_accuracy = accuracy\_score(y\_test, logreg\_preds) logreg\_probs = logreg\_model.predict\_proba(x\_test)[:, 1] logreg\_auc = roc\_auc\_score(y\_test, logreg\_probs) logreg\_fpr, logreg\_tpr, \_ = roc\_curve(y\_test, logreg\_probs)  # Decision Tree dt\_model = DecisionTreeClassifier() dt\_model.fit(x\_train, y\_train) dt\_preds = dt\_model.predict(x\_test) dt\_accuracy = accuracy\_score(y\_test, dt\_preds) dt\_probs = dt\_model.predict\_proba(x\_test)[:, 1] dt\_auc = roc\_auc\_score(y\_test, dt\_probs) dt\_fpr, dt\_tpr, \_ = roc\_curve(y\_test, dt\_probs)    # Random Forest rf\_model = RandomForestClassifier() rf\_model.fit(x\_train, y\_train) rf\_preds = rf\_model.predict(x\_test) rf\_accuracy = accuracy\_score(y\_test, rf\_preds) rf\_probs = rf\_model.predict\_proba(x\_test)[:, 1] rf\_auc = roc\_auc\_score(y\_test, rf\_probs) rf\_fpr, rf\_tpr, \_ = roc\_curve(y\_test, rf\_probs)  # Support Vector Machine svm\_model = SVC(probability=True) svm\_model.fit(x\_train, y\_train) svm\_preds = svm\_model.predict(x\_test)  svm\_accuracy = accuracy\_score(y\_test, svm\_preds) svm\_probs = svm\_model.predict\_proba(x\_test)[:, 1] svm\_auc = roc\_auc\_score(y\_test, svm\_probs)  svm\_fpr, svm\_tpr, \_ = roc\_curve(y\_test, svm\_probs)  # K-Nearest Neighbors  knn\_model = KNeighborsClassifier() knn\_model.fit(x\_train, y\_train) knn\_preds = knn\_model.predict(x\_test) knn\_accuracy = accuracy\_score(y\_test, knn\_preds) knn\_probs = knn\_model.predict\_proba(x\_test)[:, 1] knn\_auc = roc\_auc\_score(y\_test, knn\_probs)  knn\_fpr, knn\_tpr, \_ = roc\_curve(y\_test, knn\_probs)    # Plot ROC curves plt.figure(figsize=(8, 6))  25 |



**CHAPTER 10**

**DJANGO DEPLOYMENT AND WEB PAGE UI AND OUTCOME:**

**What is Django?**

Django is a high-level, open-source web framework written in Python that follows the "batteries-included" philosophy. It is designed to make web development faster and easier by providing a robust set of tools and libraries for common tasks, allowing developers to focus on building web applications rather than dealing with low-level details. Django was created in 2005 by Adrian Holovaty and Simon Willison and has since become one of the most popular web frameworks in the Python ecosystem.

**Key features and reasons for using Django**:

1. Rapid Development: Django's "batteries-included" approach provides readyto-use components for common web development tasks, such as URL routing, database ORM, user authentication, and more. This enables developers to build web applications quickly and efficiently.

2.Scalability: Django is designed to handle high-traffic and large-scale applications. It follows best practices to ensure that applications can scale up to meet increasing demands.

1. ORM (Object-Relational Mapping): Django's built-in ORM allows developers to interact with the database using Python classes and objects, rather than writing SQL queries directly. This abstraction simplifies database operations and enhances code readability.

1. Admin Interface: Django automatically generates an admin interface for managing database records. It allows developers to perform CRUD (Create, Read, Update, Delete) operations on database models without writing custom admin interfaces.

1. Security: Django emphasizes security and comes with built-in features to prevent common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

1. URL Routing and View Handling: Django's URL dispatcher maps URLs to views, making it easy to organize different parts of the web application. Views handle HTTP requests and return appropriate responses.

1. Template Engine: Django's template engine allows developers to create dynamic HTML templates, separating design from logic. This promotes code reusability and enhances front-end development.

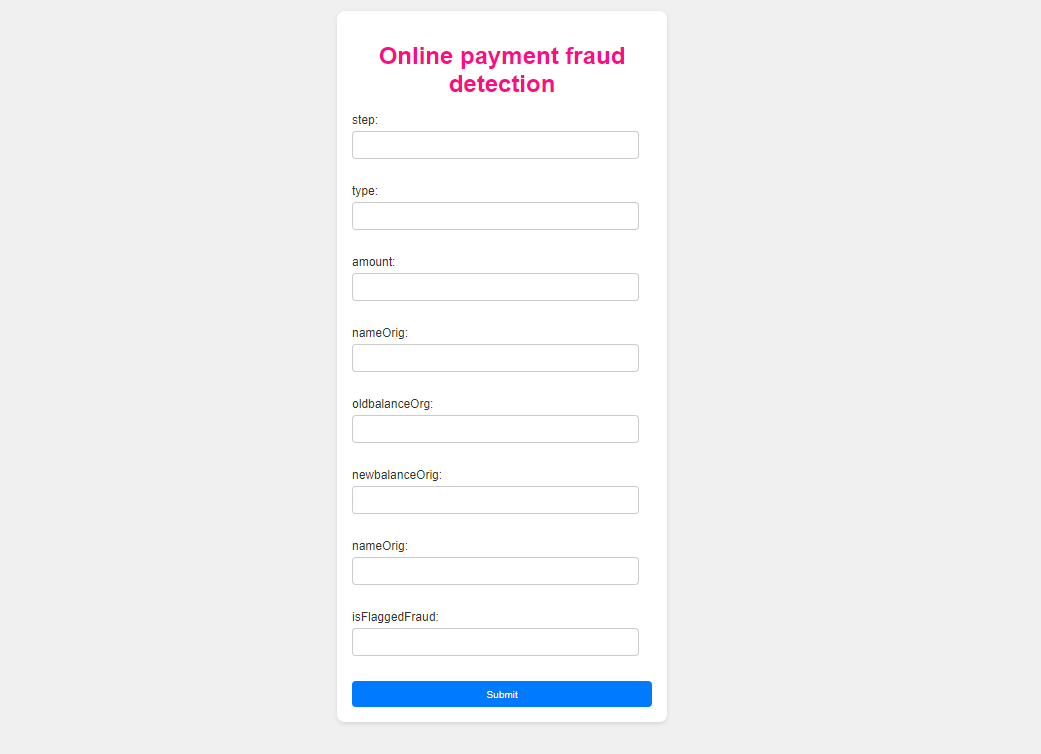
1. Form Handling: Django provides a form handling system that simplifies form creation, validation, and data cleaning. It streamlines user input and data processing.

1. Internationalization and Localization: Django supports multiple languages and provides tools for translating web applications into different languages, making it suitable for creating multilingual applications.

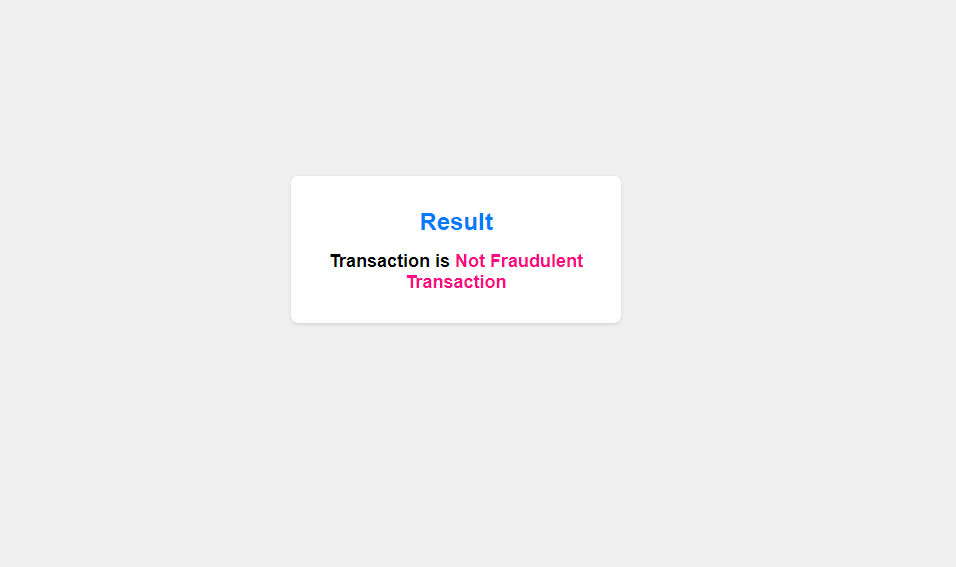
1. Versatility: Django can be used to build not only traditional web applications but also RESTful APIs, making it suitable for creating web services and integrating with other platforms.

1. Active Community: Django has a large and active community of developers, which means extensive documentation, third-party packages, and community support are available.

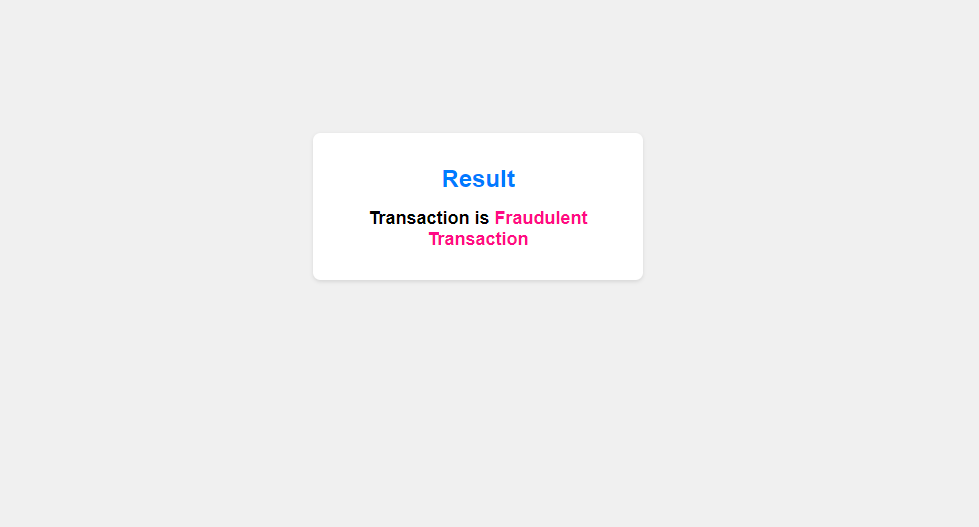
**USER INTERFACE:**



RESULT **1:**



**RESULT 2:**



**CHAPTER 11**

### ADVANTAGES & DISADVANTAGES

**Advantages of the proposed solution:**

* Accurate prediction of customer behaviour during online shopping, enabling businesses to optimize their marketing strategies and improve conversion rates.

* Insights into customer preferences and browsing patterns, allowing businesses to tailor their product offerings and enhance the overall shopping experience.

* The ability to make data-driven decisions based on customer segmentation and targeted marketing.

**Disadvantages of the proposed solution:**

* Dependency on the quality and availability of data for accurate predictions.

* The need for continuous monitoring and updating of the predictive model as customer behaviour and preferences change over time.

* Ethical consideration regarding the collection and usage of customer data, ensuring privacy and compliance with data protection regulations

### 11.1 APPLICATIONS

The proposed solution can be applied in various areas, including:

E-commerce platforms: Predicting customer behaviour to optimize product recommendations, personalize marketing campaigns, and improve customer satisfaction.

Digital marketing:

Targeting specific customer segments with tailored advertisements and promotional offers.

Market research:

Analyzing online shopping behaviour to identify trends, preferences, and patterns for strategic decision-making.

Customer relationship management:

Understanding customer preferences to provide personalized customer support and enhance customer loyalty

**CHAPTER 12**

## Conclusion

1. Dataset Overview: The dataset contains 6,362,620 rows and 11 columns. Each row represents a transaction, and the columns represent various attributes of the transactions, such as time, transaction type, transaction amount, customer information, and recipient information. The target variable is `isFraud`, which indicates whether a transaction is fraudulent (1) or not (0).

2. Features: The dataset includes features such as `step`, `type`, `amount`, `nameOrig`, `oldbalanceOrg`, `newbalanceOrig`, `nameDest`, `oldbalanceDest`, `newbalanceDest`, `isFraud`, and `isFlaggedFraud`.

3. Target Variable: The primary task is to predict whether a transaction is fraudulent (`isFraud` = 1) or not (`isFraud` = 0). This is a binary classification problem.

4. Fraud Detection: The goal is to develop a machine learning model that can effectively detect fraudulent transactions to prevent financial losses and protect customers and businesses.

5. Best Algorithm: According to the inference made from the provided code, the Random Forest classifier achieved the highest AUC score among all the classifiers evaluated. A higher AUC score indicates better performance in distinguishing between fraudulent and non-fraudulent transactions. Therefore, it can be concluded that the Random Forest algorithm is the best-performing model for this specific dataset and task of online payment fraud detection.

However, it's important to note that the choice of the "best" algorithm may depend on various factors, including data size, data quality, feature engineering, and specific performance metrics. Therefore, it's always recommended to perform thorough model evaluation and hyperparameter tuning to ensure the most suitable algorithm for a given problem. Additionally, you may want to explore other machine learning algorithms and techniques to further improve the model's performance.