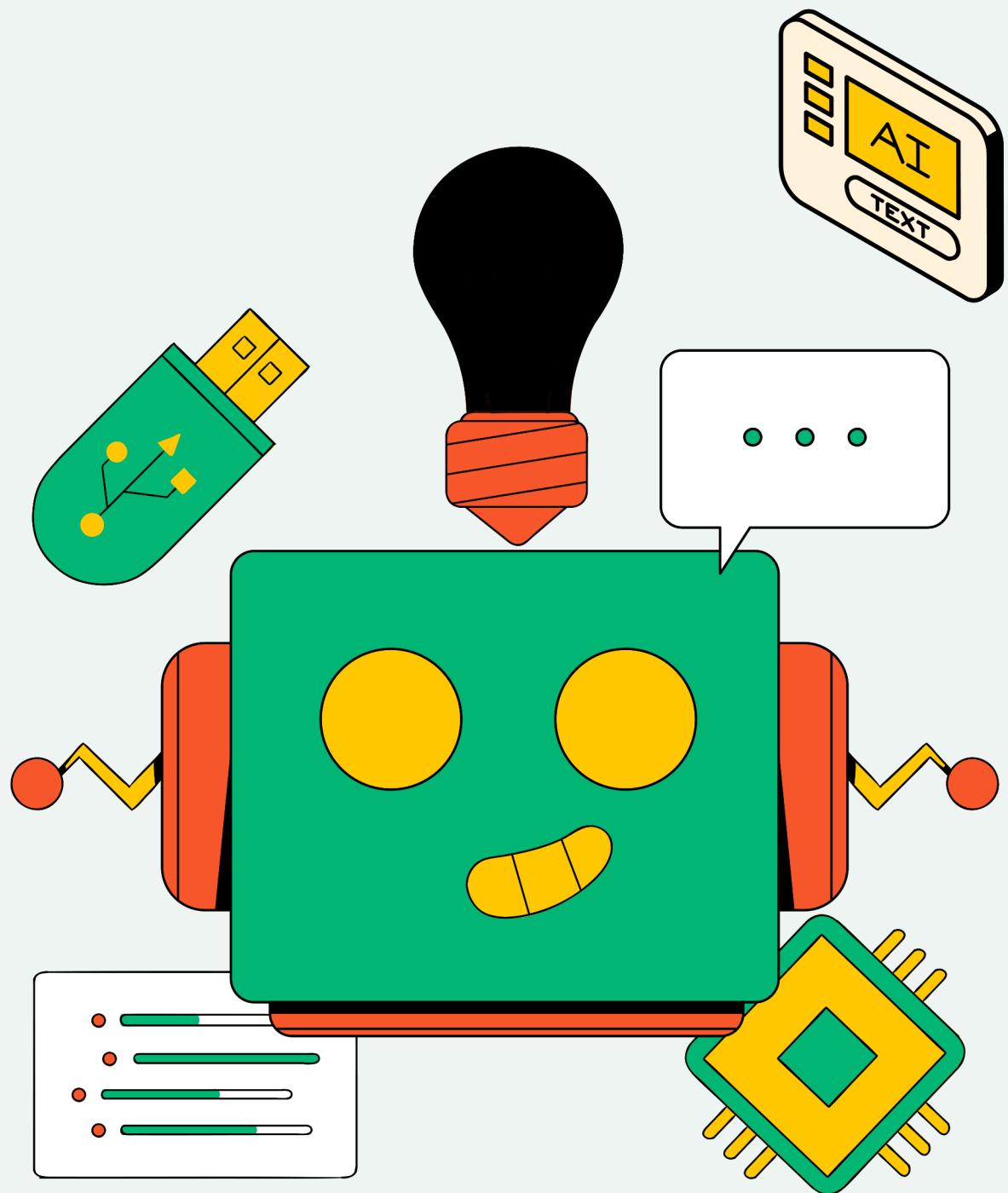


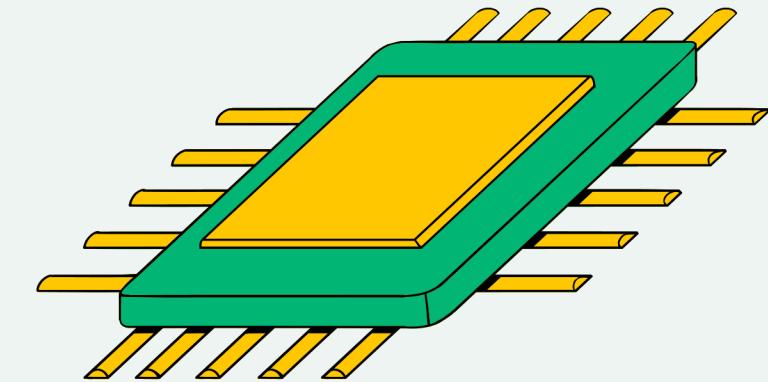
MENTORNESS



FAST TAG FRAUD DETECTION USING MACHINE LEARNING

PRESENTED BY:

LOKESH SINGH



PRESENTATION OUTLINE

- Problem Statement
- Data Description
- Project Objective
- Data Exploration
- Exploratory Data Analysis
- Feature Engineering
- Model Development
- Real - Time Fraud Detection
- Deliverables
- Expected Outcomes
- References



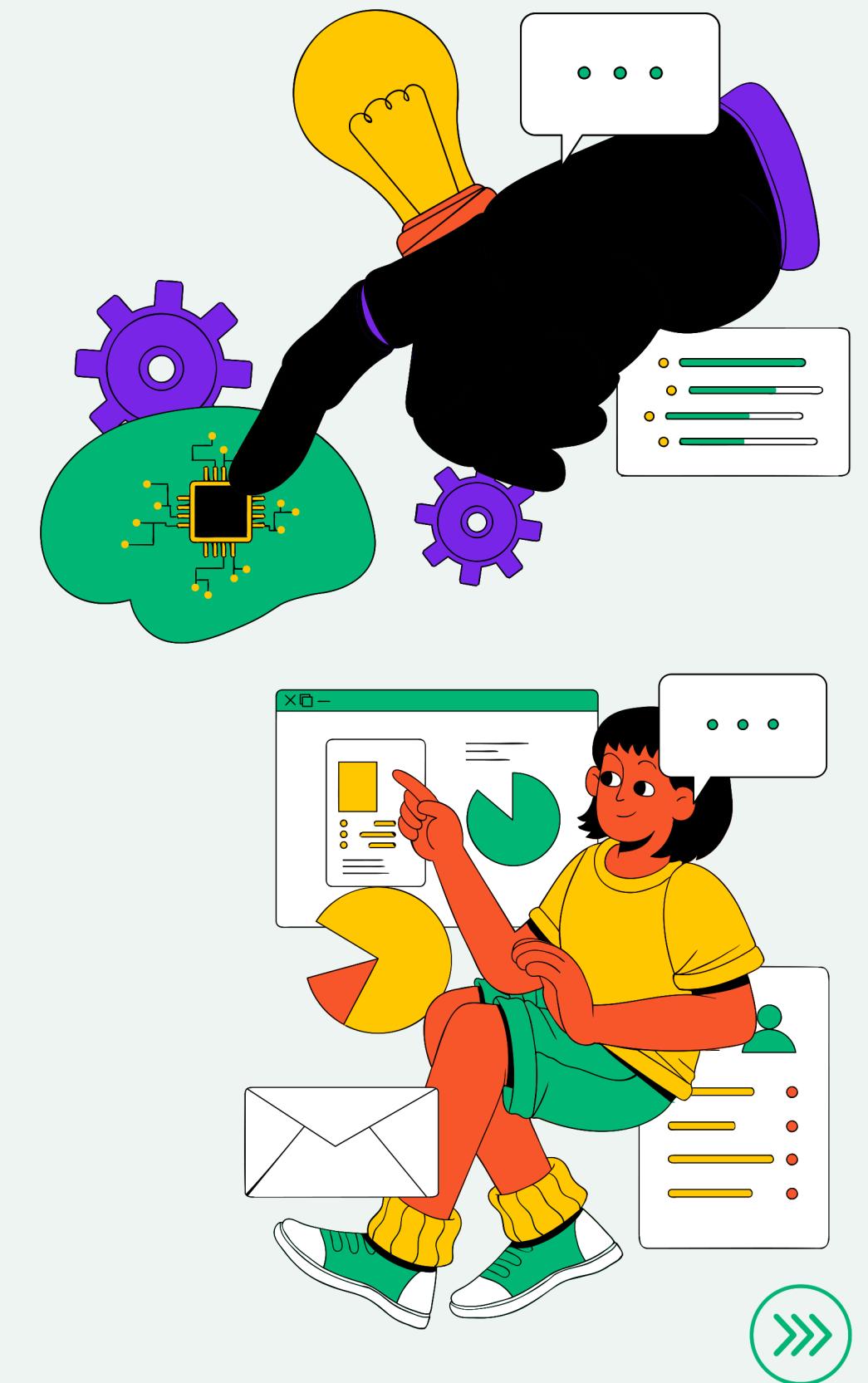
PROBLEM STATEMENT

This project focuses on leveraging machine learning classification techniques to develop an effective fraud detection system for Fastag transactions. The dataset comprises key features such as transaction details, vehicle information, geographical location, and transaction amounts. The goal is to create a robust model that can accurately identify instances of fraudulent activity, ensuring the integrity and security of Fastag transactions.

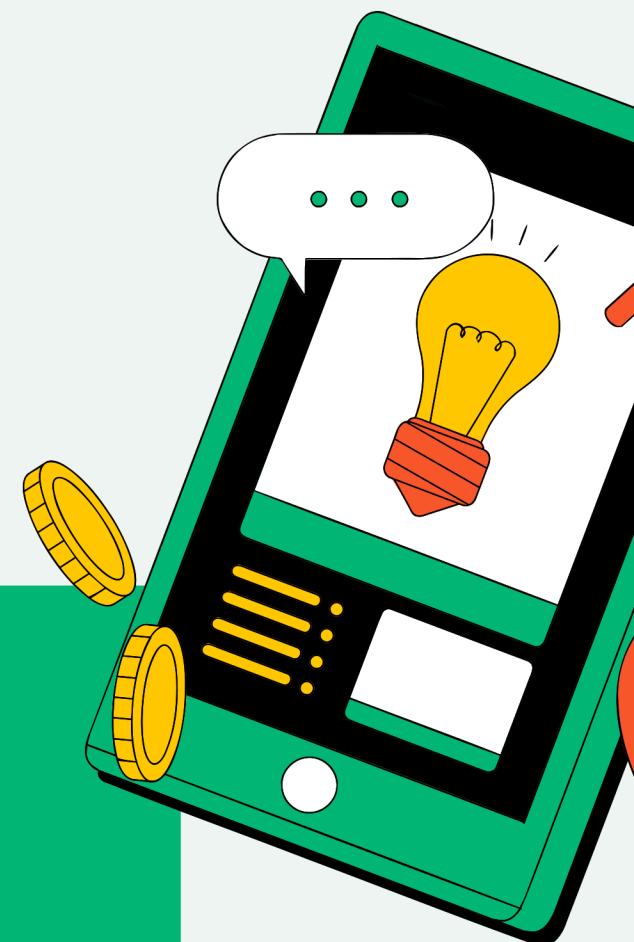
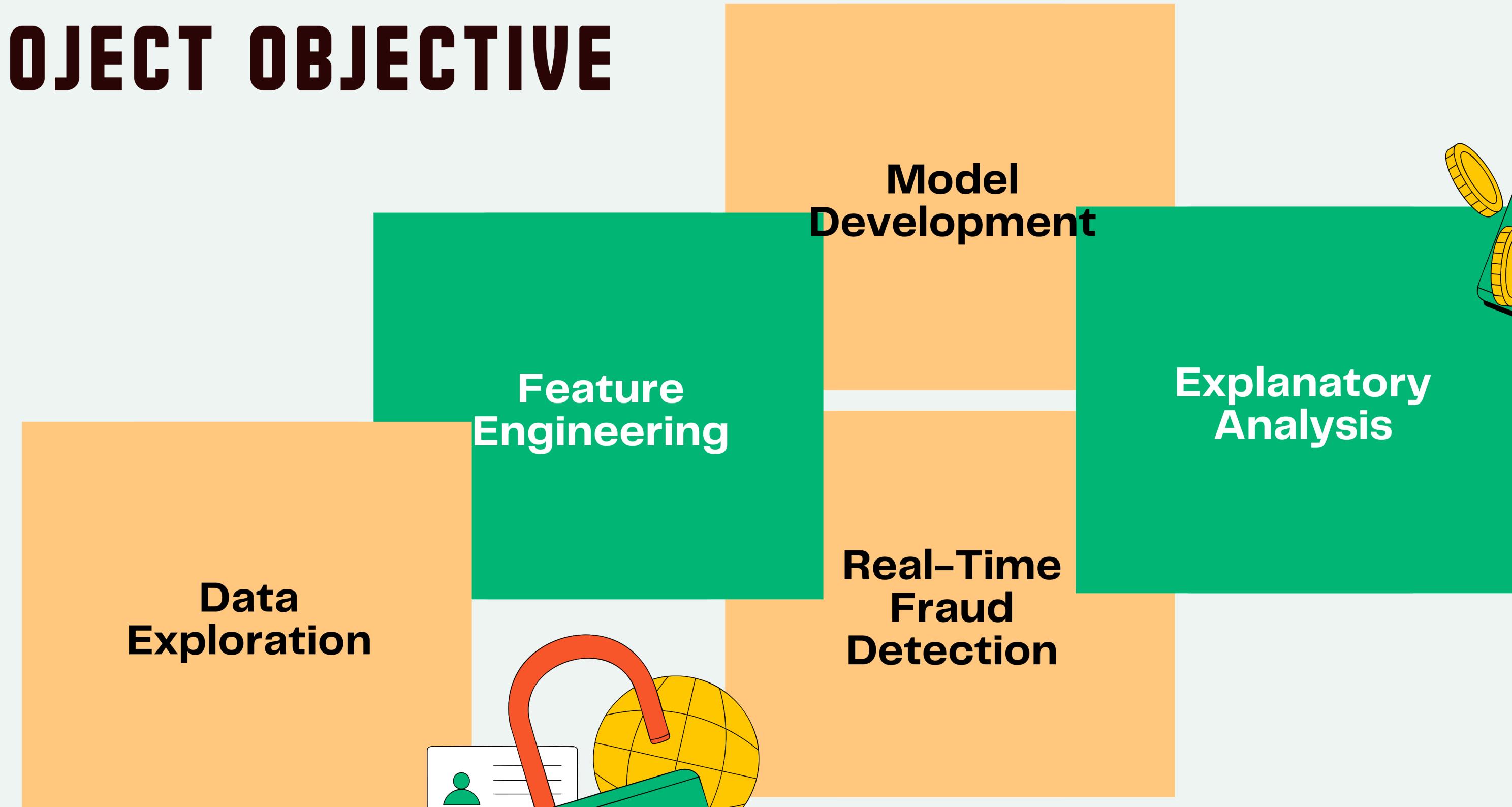


DATASET DESCRIPTION

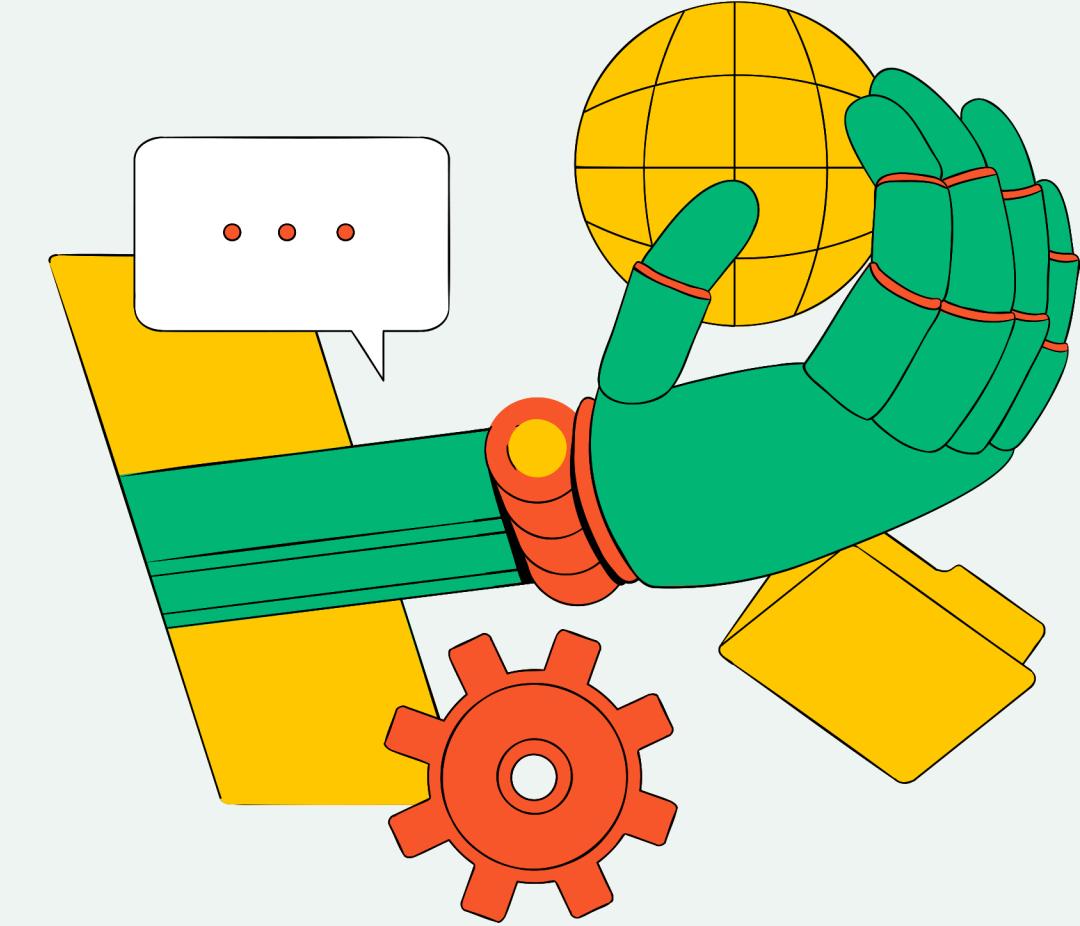
- **Transaction_ID:** Unique identifier for each transaction.
- **Timestamp:** Date and time of the transaction.
- **Vehicle_Type:** Type of vehicle involved in the transaction.
- **FastagID:** Unique identifier for Fastag.
- **TollBoothID:** Identifier for the toll booth.
- **Lane_Type:** Type of lane used for the transaction.
- **Vehicle_Dimensions:** Dimensions of the vehicle.
- **Transaction_Amount:** Amount associated with the transaction.
- **Amount_paid:** Amount paid for the transaction.
- **Geographical_Location:** Location details of the transaction.
- **Vehicle_Speed:** Speed of the vehicle during the transaction.
- **Vehicle_Plate_Number:** License plate number of the vehicle.
- **Fraud_indicator:** Binary indicator of fraudulent activity (target variable).



PROJECT OBJECTIVE



DATA EXPLORATION



DATA PREPARATION

- Data collection, cleaning, and preprocessing
- Ensuring data quality and reliability
- Data labeling and annotation

The main goal of data exploration are to identify potential relationship between variables, detect outliers or anomalies, understand the distribution of data, and gain insights that can guide further analysis or modelling tasks.



DATA SUMMARY

Importing CSV File

```
[1]: import pandas as pd\n\n#Load theb dataset\ndata = pd.read_csv("FastagFraudDetection.csv")\n\nprint("Datatset loaded successfully")\n\nDatatset loaded successfully\n\n[2]: #Display first few rows of dataset\nprint("First few rows of the dataset: ")\nprint(data.head())\n\nFirst few rows of the dataset:\n    Transaction_ID      Timestamp Vehicle_Type        FastagID TollBoothID \\\n0            1  1/6/2023 11:20         Bus  FTG-001-ABC-121       A-101\n1            2  1/7/2023 14:55        Car  FTG-002-XYZ-451       B-102\n2            3  1/8/2023 18:25  Motorcycle        NaN       D-104\n3            4  1/9/2023 2:05        Truck  FTG-044-LMN-322       C-103\n4            5  1/10/2023 6:35        Van   FTG-505-DEF-652       B-102\n\n    Lane_Type Vehicle_Dimensions Transaction_Amount Amount_paid \\\n0   Express           Large             350          120\n1  Regular           Small             120          100\n2  Regular           Small              0             0\n3  Regular           Large             350          120\n4  Express           Medium            140          100\n\n    Geographical_Location  Vehicle_Speed Vehicle_Plate_Number \\\n0  13.059816123454882, 77.77068662374292            65  KA11AB1234\n1  13.059816123454882, 77.77068662374292            78  KA66CD5678\n2  13.059816123454882, 77.77068662374292            53  KA88EF9012\n3  13.059816123454882, 77.77068662374292            92  KA11GH3456\n4  13.059816123454882, 77.77068662374292            60  KA44IJ6789\n\n    Fraud_indicator\n0\n1\n2\n3\n4
```

Summary Statistics/ Missing Values

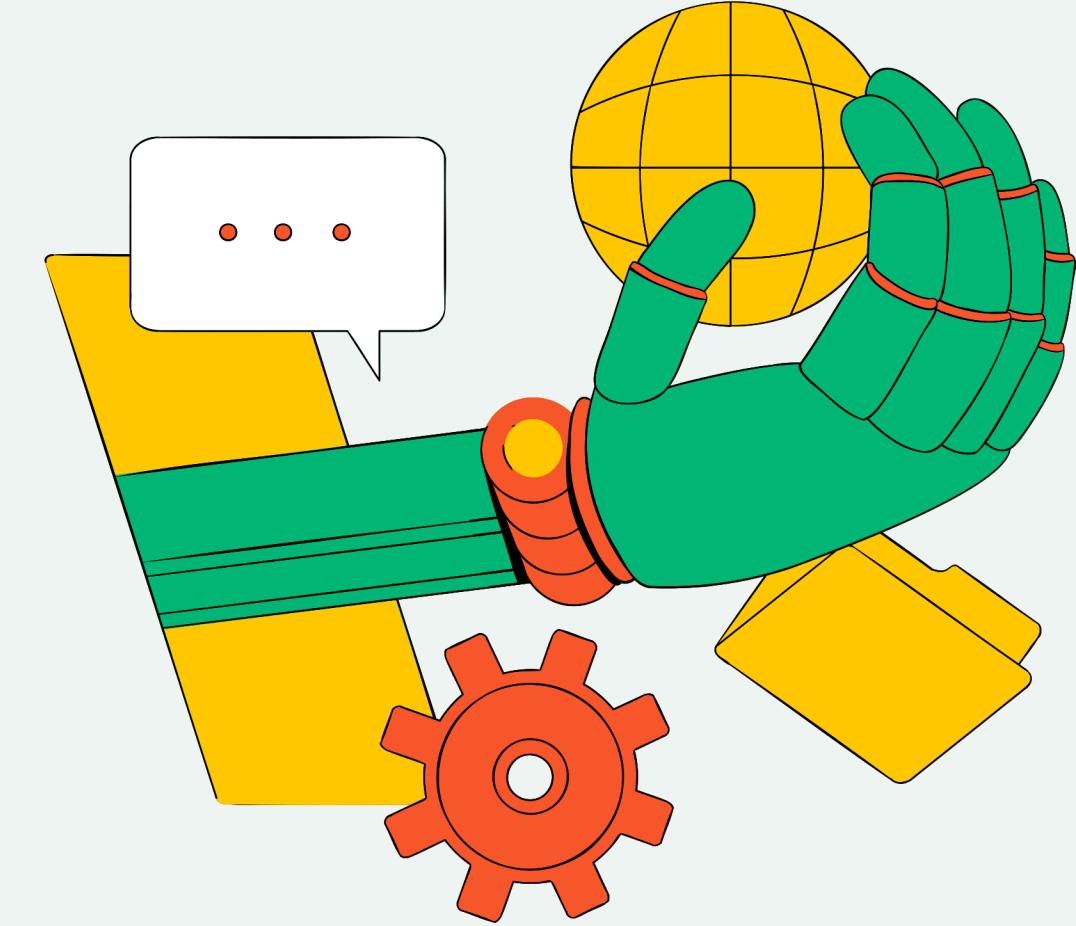
```
[4]: # Summary statistics of numerical columns\nprint("\nSummary statistics:")\nprint(data.describe())\n\nSummary statistics:\n    Transaction_ID  Transaction_Amount  Amount_paid  Vehicle_Speed\ncount      5000.000000      5000.000000      5000.000000      5000.000000\nmean      2500.500000     161.062000     141.261000      67.851200\nstd       1443.520003     112.44995     106.480996     16.597547\nmin        1.000000      0.000000      0.000000      10.000000\n25%     1250.750000     100.00000     90.000000      54.000000\n50%     2500.500000     130.00000     120.000000      67.000000\n75%     3750.250000     290.00000     160.000000      82.000000\nmax      5000.000000     350.00000     350.000000     118.000000\n\n[5]: #check for missing values\nprint("\n Missing Values")\nprint(data.isnull().sum())\n\nMissing Values\nTransaction_ID          0\nTimestamp                0\nVehicle_Type              0\nFastagID                 549\nTollBoothID               0\nLane_Type                  0\nVehicle_Dimensions          0\nTransaction_Amount          0\nAmount_paid                  0\nGeographical_Location          0\nVehicle_Speed                  0\nVehicle_Plate_Number          0\nFraud_indicator                 0\ndtype: int64
```

EXPLANATORY ANALYSIS

EXPLANATORY ANALYSIS

- Provide insights into factors contributing to fraudulent transactions.

Explanatory analysis involves exploring and understanding data to gain insights into patterns, relationships, and trends. Unlike predictive analysis, which focuses on making predictions or classifications, explanatory analysis aims to explain the underlying structure and characteristics of the data.



DATA VISUALIZATION

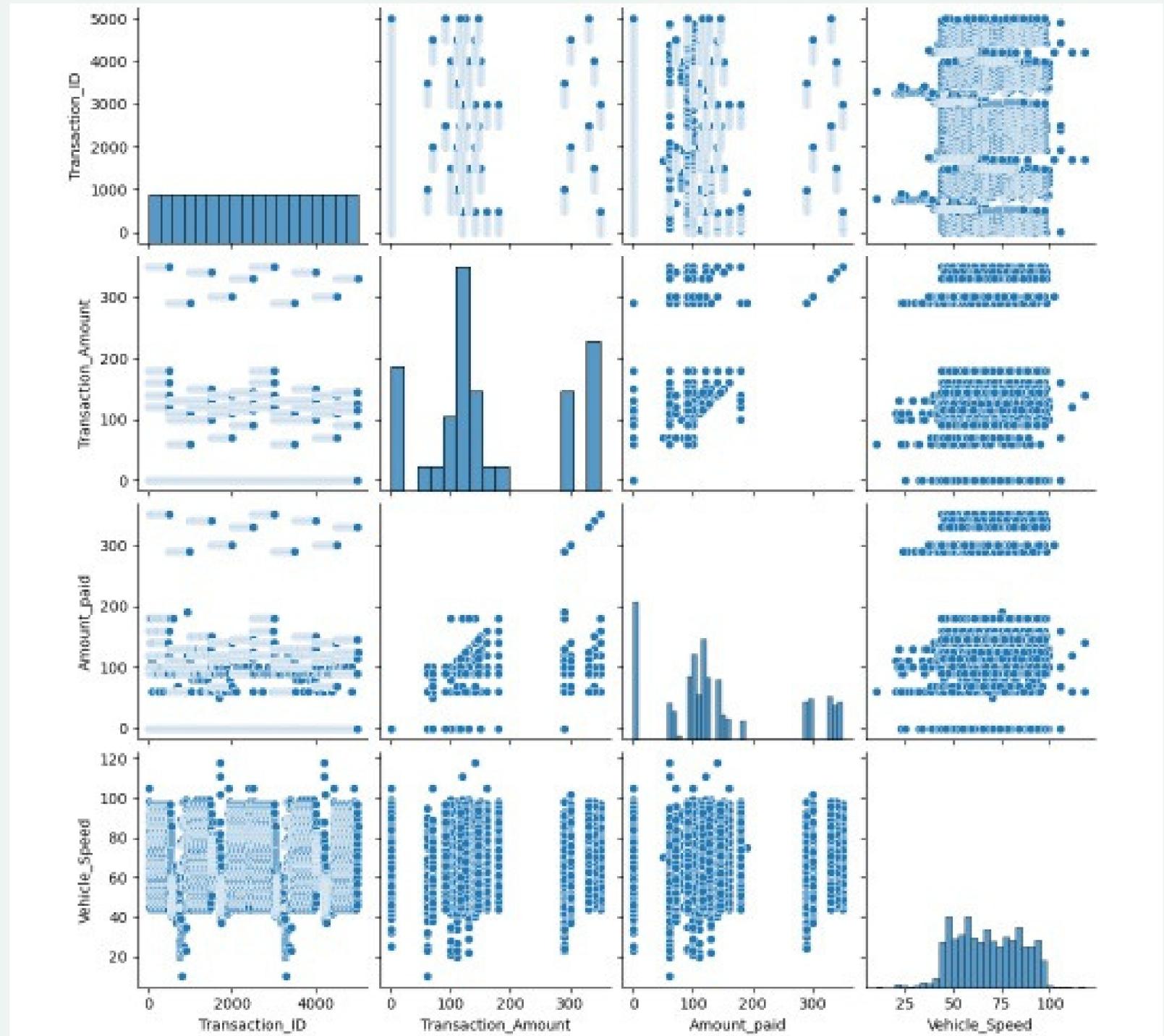
Pairwise Scatter plots for Numerical Values

```
#unique values in categorical columns
print("\nUnique values in categorical columns:")
for col in data.select_dtypes(include='object').columns:
    print(f"(col): {data[col].unique()}")

#Visualize distribution of numerical columns
import matplotlib.pyplot as plt
import seaborn as sns

sns.pairplot(data)
plt.show()

Unique values in categorical columns:
(col): ['1/6/2023 11:20' '1/7/2023 14:55' '1/8/2023 18:25' ... '2/5/2023 5:08'
 '2/20/2023 20:34' '3/10/2023 0:59']
(col): ['Bus' 'Car' 'Motorcycle' 'Truck' 'Van' 'Sedan' 'SUV']
(col): ['FTG-001-ABC-121' 'FTG-002-XYZ-451' nan ... 'FTG-447-PLN-109'
 'FTG-458-VFR-876' 'FTG-459-WSX-543']
(col): ['A-101' 'B-102' 'D-104' 'C-103' 'D-105' 'D-106']
(col): ['Express' 'Regular']
(col): ['Large' 'Small' 'Medium']
(col): ['13.059816123454882, 77.77068662374292'
 '13.042660878688794, 77.47580097259879'
 '12.84197701525119, 77.67547528176169'
 '12.936687032945434, 77.53113977439017'
 '13.21331620748757, 77.55413526894684']
(col): ['KA11AB1234' 'KA66CD5678' 'KA88EF9012' ... 'KA33WX6789' 'KA35YZ0123'
 'KA37AB3456']
(col): ['Fraud' 'Not Fraud']
```



DATA VISUALIZATION

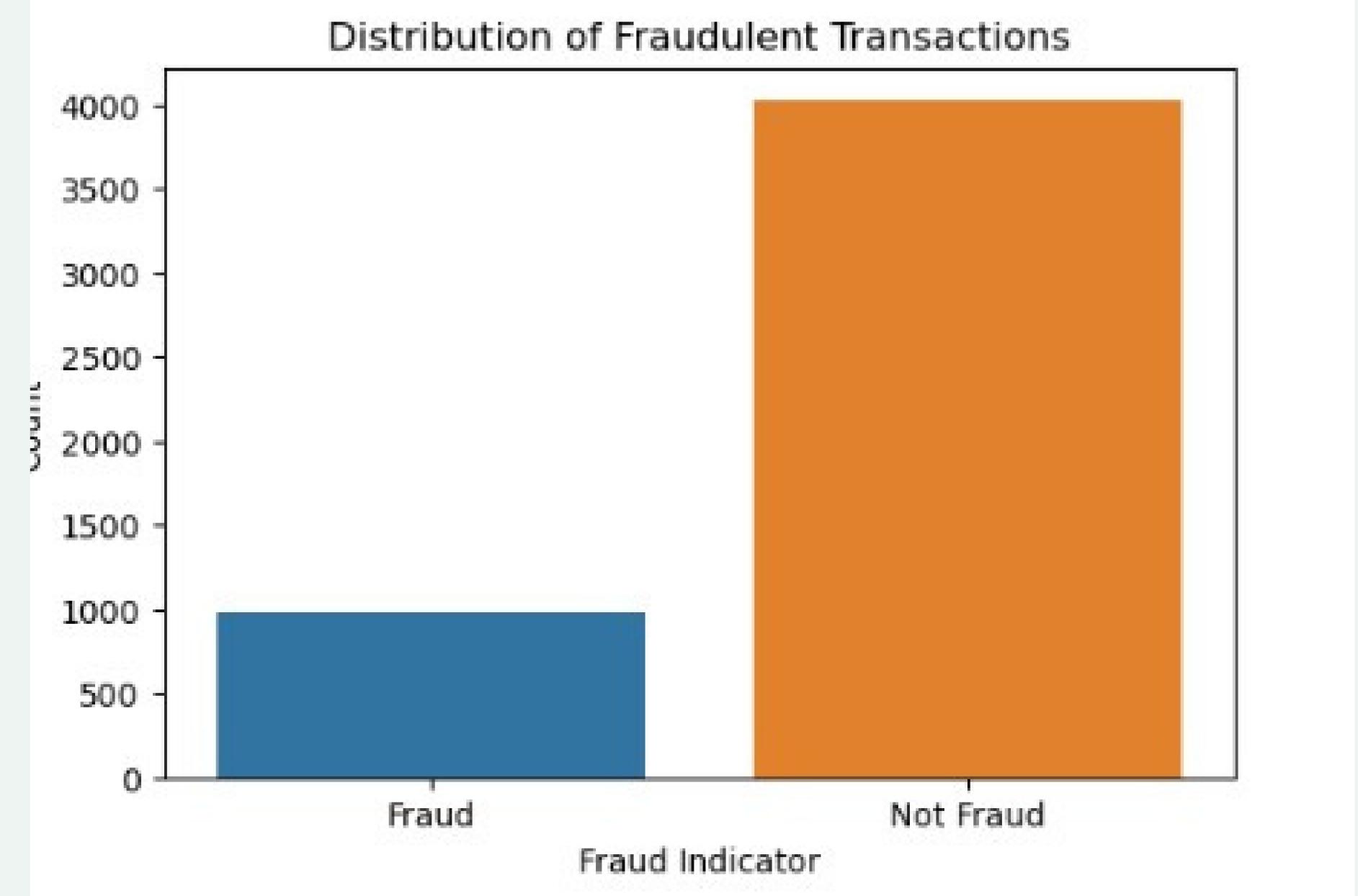
Bar plots for Categorical Features

```
# Summary statistics
print("\nSummary statistics:")
print(data.describe())

Summary statistics:
   Transaction_ID  Transaction_Amount  Amount_paid  Vehicle_Speed
count      5000.000000      5000.000000  5000.000000      5000.000000
mean     2500.500000     161.062000   141.261000      67.851200
std      1443.520003    112.449950   106.480996      16.597547
min       1.000000       0.000000     0.000000      10.000000
25%    1250.750000     100.000000    90.000000      54.000000
50%    2500.500000     130.000000   120.000000      67.000000
75%    3750.250000     290.000000   160.000000      82.000000
max      5000.000000     350.000000   350.000000     118.000000

import seaborn as sns
import matplotlib.pyplot as plt

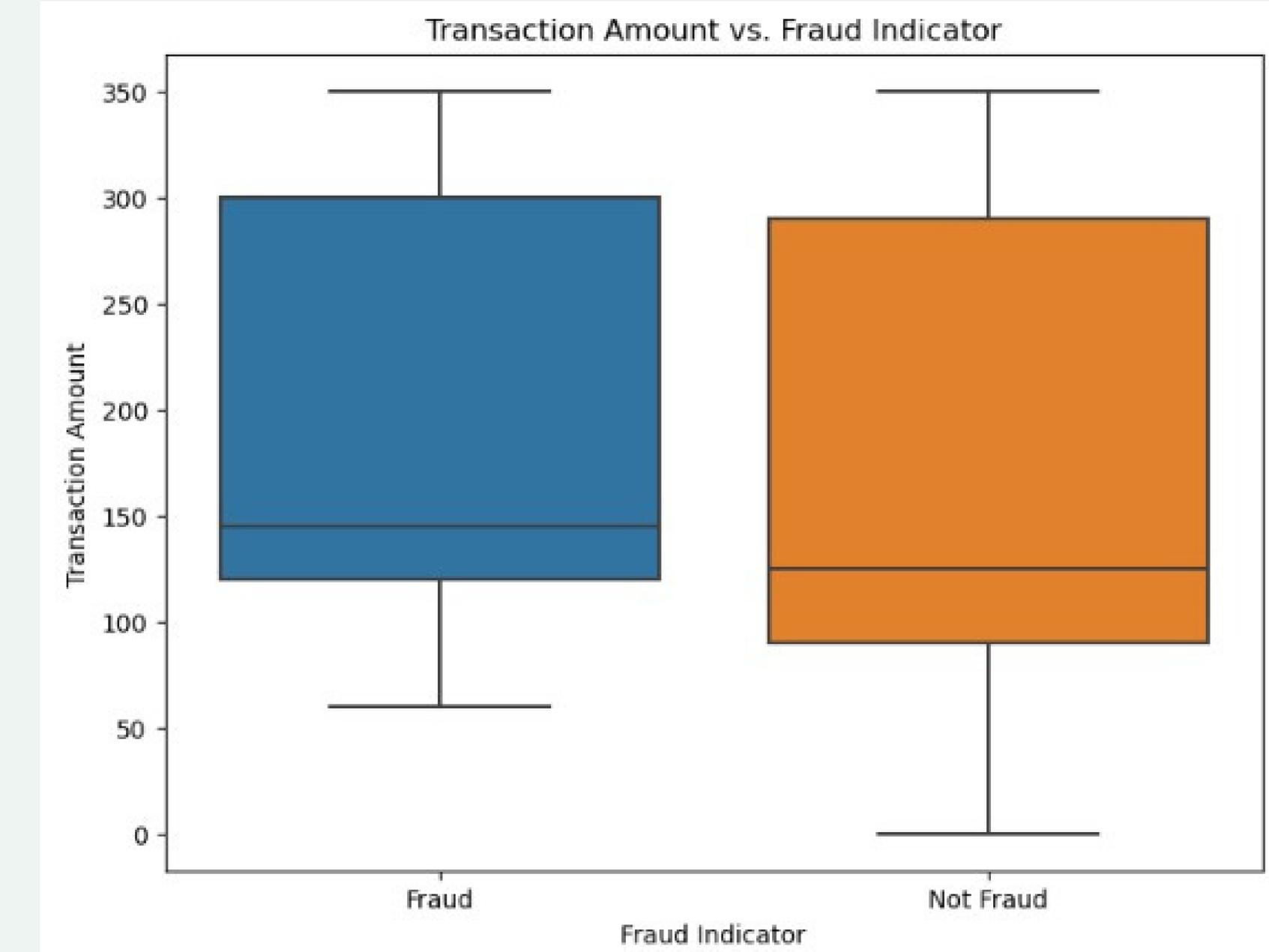
# Visualize the distribution of the target variable (Fraud_indicator)
plt.figure(figsize=(6, 4))
sns.countplot(x='Fraud_indicator', data=data)
plt.title('Distribution of Fraudulent Transactions')
plt.xlabel('Fraud Indicator')
plt.ylabel('Count')
plt.show()
```



DATA VISUALIZATION

Box plots for Numerical Features

```
# Visualize the relationship between Transaction_Amount and Fraud_indicator
plt.figure(figsize=(8, 6))
sns.boxplot(x='Fraud_indicator', y='Transaction_Amount', data=data)
plt.title('Transaction Amount vs. Fraud Indicator')
plt.xlabel('Fraud Indicator')
plt.ylabel('Transaction Amount')
plt.show()
```



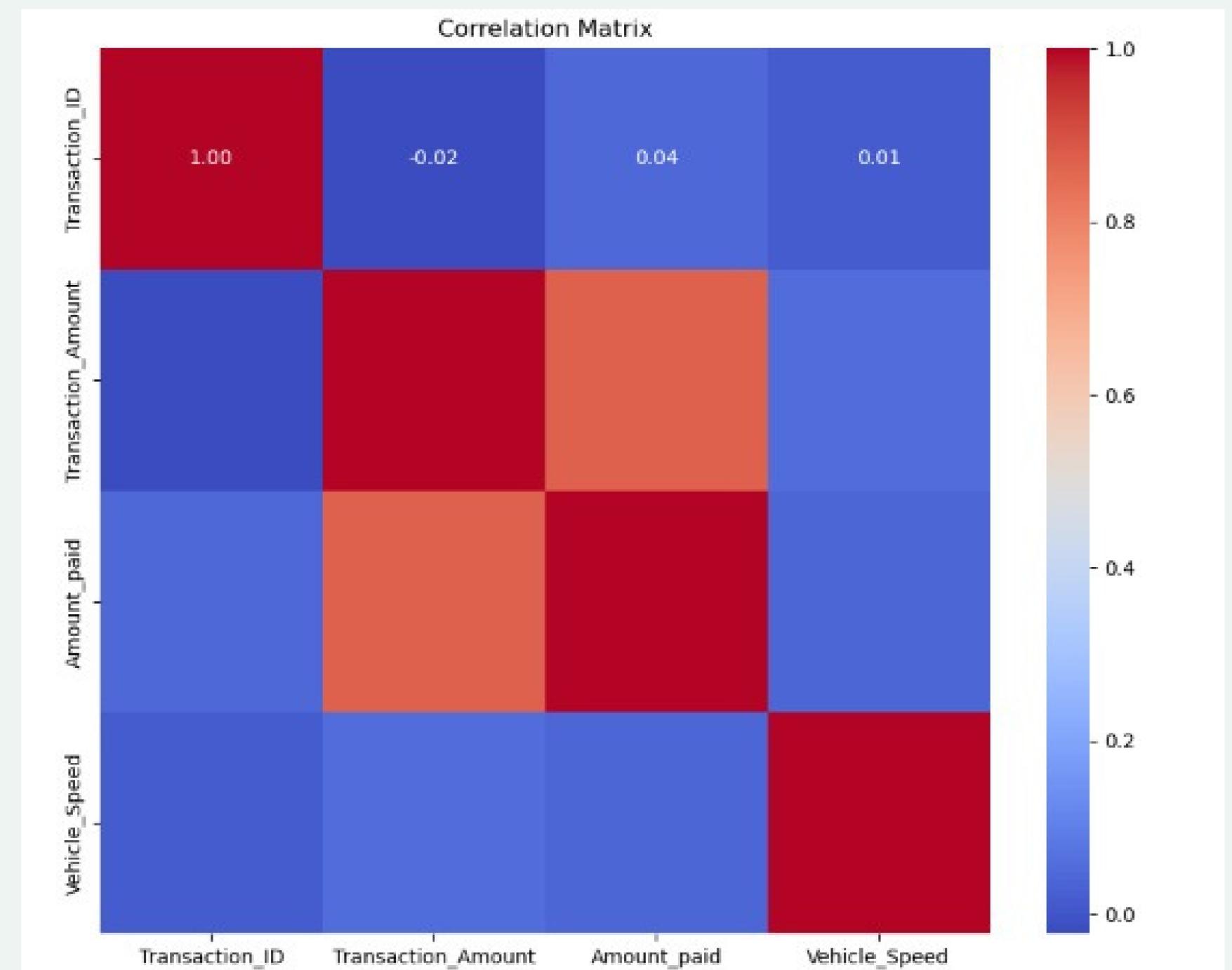
DATA VISUALIZATION

Correlation Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt

# Select only numerical columns for correlation matrix
numerical_data = data.select_dtypes(include=['float64', 'int64'])

# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(numerical_data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```

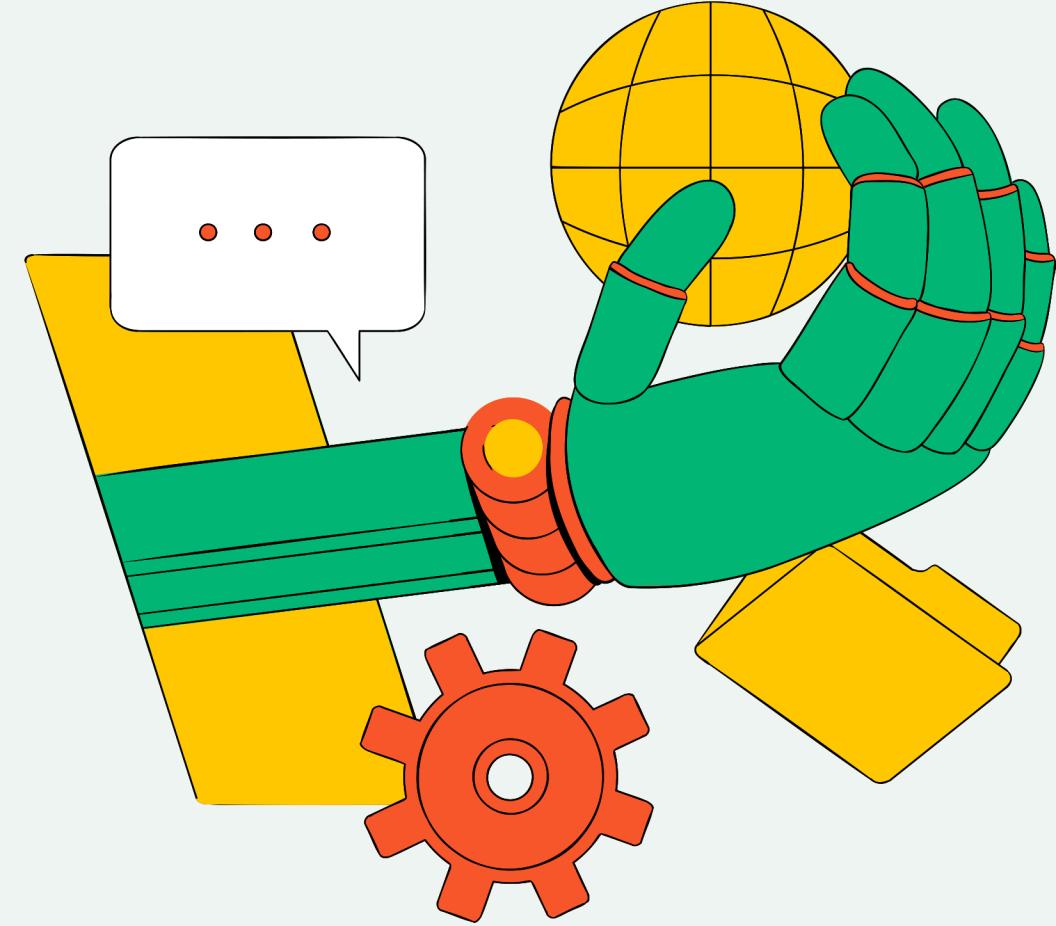


FEATURE ENGINEERING

FEATURE ENGINEERING

- Identify and engineer relevant features that contribute to fraud detection accuracy.

Feature engineering is the process of creating new features or modifying existing one in a dataset to improve the performance of machine learning models. It involves selecting, transforming and creating features that are relevant and informative for the task at hand.



FEATURE ENGINEERING

FEATURE ENGINEERING

```
from sklearn.preprocessing import LabelEncoder

# Load the dataset
fastag_data = pd.read_csv('FastagFraudDetection.csv')

# Feature engineering

# Convert Timestamp to datetime
fastag_data['Timestamp'] = pd.to_datetime(fastag_data['Timestamp'])

# Extract date and time features
fastag_data['Day'] = fastag_data['Timestamp'].dt.day
fastag_data['Month'] = fastag_data['Timestamp'].dt.month
fastag_data['Year'] = fastag_data['Timestamp'].dt.year
fastag_data['Hour'] = fastag_data['Timestamp'].dt.hour
fastag_data['Minute'] = fastag_data['Timestamp'].dt.minute

# Encode categorical variables using LabelEncoder
label_encoder = LabelEncoder()
fastag_data['Vehicle_Type'] = label_encoder.fit_transform(fastag_data['Vehicle_Type'])
fastag_data['Lane_Type'] = label_encoder.fit_transform(fastag_data['Lane_Type'])
fastag_data['Geographical_Location'] = label_encoder.fit_transform(fastag_data['Geographical_Location'])

# Extract features from Vehicle_Plate_Number
# For example, extracting the state code or vehicle category

# Drop unnecessary columns
fastag_data.drop(['Transaction_ID', 'Timestamp', 'FastagID', 'TollBoothID', 'Vehicle_Dimensions', 'Vehicle_Plate_Number'], axis=1, inplace=True)

# Save the engineered dataset
fastag_data.to_csv('engineered_fastag_data.csv', index=False)
```

Converting timestamp to datetime

Encoding categorical variables

Dropping unnecessary columns

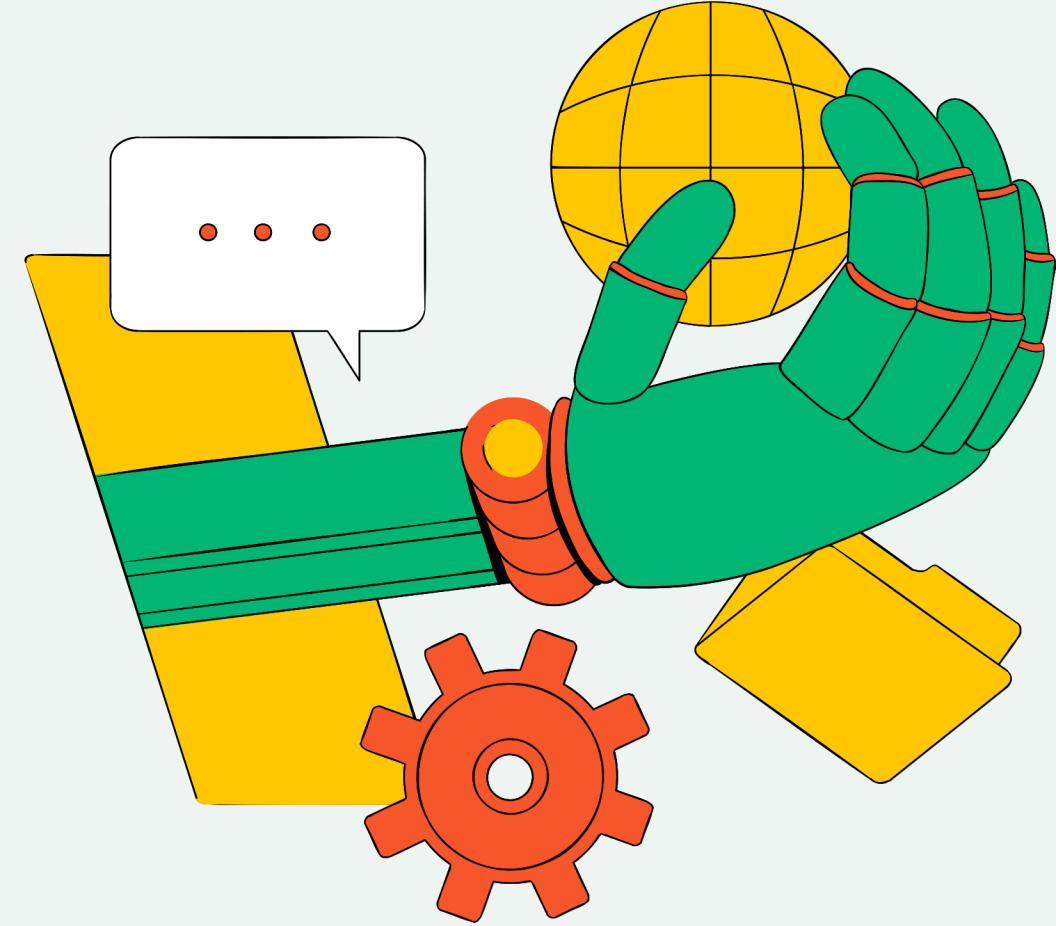
saved the engineered dataset

MODEL DEVELOPMENT

MODEL DEVELOPMENT

- Build a machine learning classification model to predict and fast tag transaction fraud.
- Evaluate and fine-tune model performance using appropriate metrics.

Model development is the process of creating, training, and evaluating machine learning models to solve specific problem or tasks. It involves selecting the appropriate algorithm, preprocessing the data, tuning model parameters and assessing the model's performance using various evaluation metrics.



MODEL DEVELOPMENT

MODEL DEVELOPMENT

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Split the dataset into features (X) and target variable (y)
X = fastag_data.drop('Fraud_indicator', axis=1)
y = fastag_data['Fraud_indicator']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Random Forest classifier
clf = RandomForestClassifier()

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = clf.predict(X_test)

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

# Convert accuracy to percentage
accuracy_percentage = accuracy * 100

# Print the evaluation metrics
print(f"Accuracy: {accuracy_percentage:.2f}%")
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

Evaluation Metrics Results

Accuracy: 98.00%

Confusion Matrix:

```
[[197  20]
 [  0 783]]
```

Classification Report:

	precision	recall	f1-score	support
Fraud	1.00	0.91	0.95	217
Not Fraud	0.98	1.00	0.99	783
accuracy			0.98	1000
macro avg	0.99	0.95	0.97	1000
weighted avg	0.98	0.98	0.98	1000

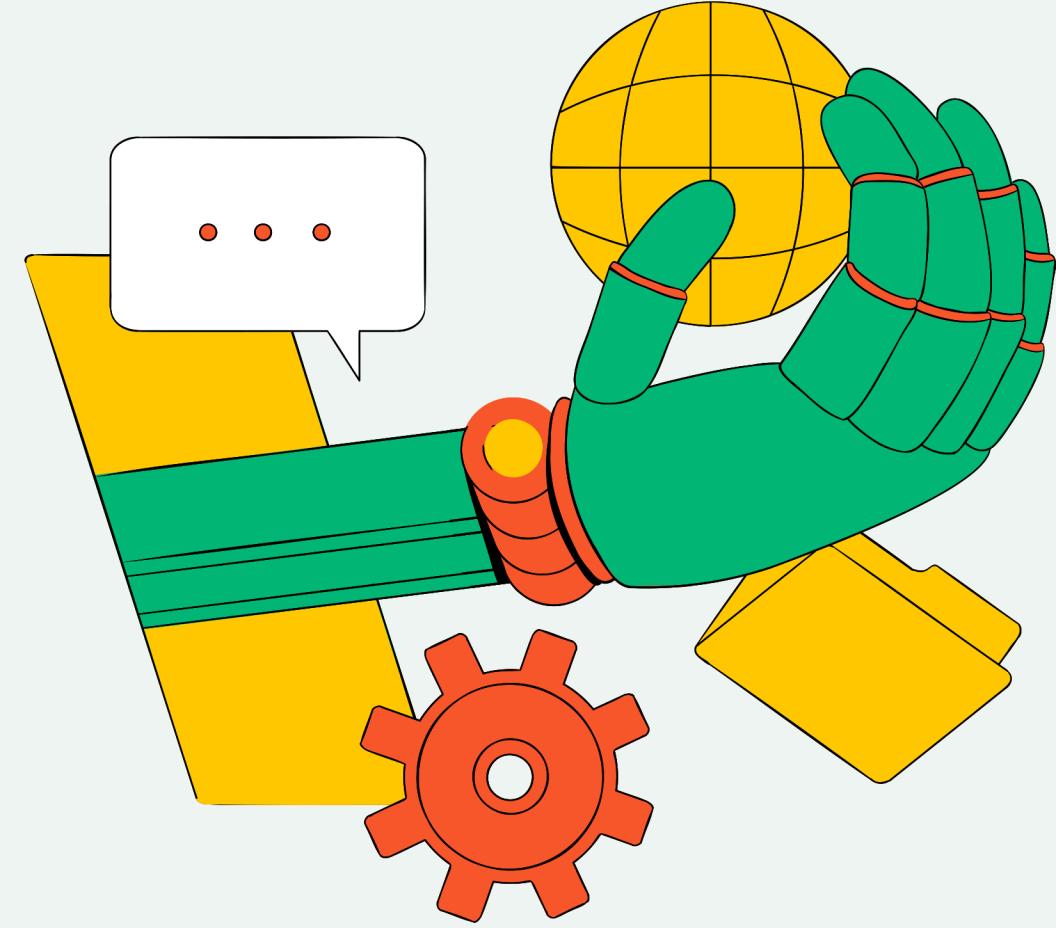
REAL-TIME FRAUD DETECTION

REAL-TIME FRAUD DETECTION

- Explore the feasibility of implementing the model for real-time fraud detection.

Real-Time fraud detection is the process of identifying and preventing fraudulent activities as they occur in a timely manner.

The goal is to detect fraud as quickly as possible to minimize financial losses and protect against security threats.



REAL-TIME FRAUD DETECTION

REAL - TIME FRAUD DETECTION

```

import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import joblib
import time

# Load the historical dataset
historical_data = pd.read_csv('engineered_fastag_data.csv')

# Feature engineering (if necessary)

# Separate features and target variable
X = historical_data.drop('Fraud_indicator', axis=1)
y = historical_data['Fraud_indicator']

# Initialize the model
model = RandomForestClassifier()

# Train the model on historical data
model.fit(X, y)

# Save the trained model
joblib.dump(model, 'fastag_fraud_detection_model.pkl')

# Simulate real-time data arrival
while True:
    # Get new data (replace this with code to receive real-time data)
    new_data = pd.read_csv('engineered_fastag_data.csv')

    # Separate features from the new data
    X_new = new_data.drop('Fraud_indicator', axis=1)

    # Predict fraud for the new data
    fraud_predictions = model.predict(X_new)

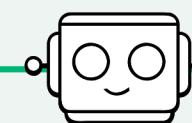
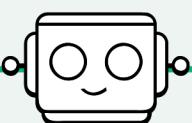
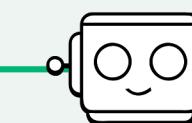
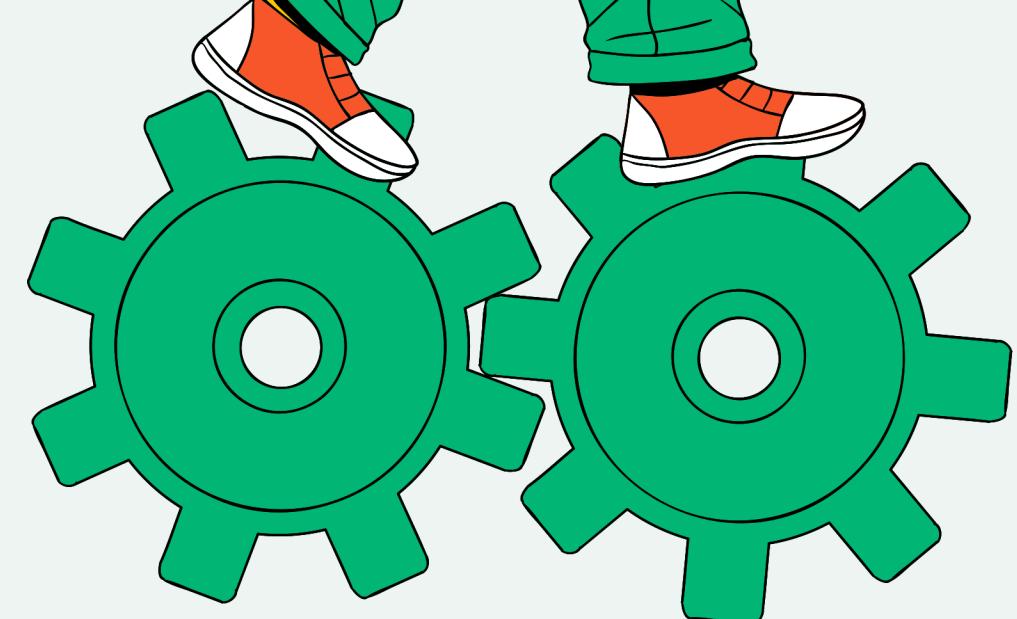
    # Print the predictions (or take further action)
    print("New data processed. Predicted fraud status:", fraud_predictions)

    # Wait for a specified time interval before processing next batch of data
    time.sleep(60) # wait for 1 minute

```

Result Analysis

DELIVERABLES



01

- Trained machine learning model for Fastag fraud detection.

02

- Evaluation metrics and analysis report.

03

- Documentation on relevant features and their impact on fraud detection.

EXPECTED OUTCOME



An effective and scalable Fastag fraud detection system capable of minimizing financial losses and ensuring the security of digital toll transactions.



THANKYOU

**For joining me on this journey of exploring Real-Time Fast tag Fraud Detection.
Our knowledge and skills will continue to evolve with practice and experimentation.**

FOLLOW ME



<https://www.linkedin.com/in/lokesh-pratap-singh-aab024203/>



<https://github.com/lokesh-pratap>

