AVERAGE FUEL CONSUMPTION IN VEHICLES

Real Time Project report

Submitted in partial fulfillment of the requirement for the award of the Degree of

Bachelor of Technology (B. Tech)

in

Computer Science and Engineering (AI&ML)

By

V.Yeshaswini	22AG1A6659
B.Pradeep	22AG1A6609
D.Akhil	22AG1A6615

Under the Esteemed Guidance of

Dr.V. Maheshwar Reddy

Associate Professor

Department of Computer Science and Engineering (AIML)

ACE ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

(NBA Accredited B.Tech Courses: ECE, EEE & CSE)

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana)

Ankushapur(V), Ghatkesar(M), Medchal- Malkajgiri Dist - 501 301.

JULY 2024.

ACE

ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

Website: www.aceec.ac.in E-mail: info@aceec.ac.in

CERTIFICATE

This is to certify that the Real Time Project work entitled "AVERAGE FUEL CONSUMPTION IN VEHICLES" is being submitted by V..Yeshaswini(22AG1A6659), B.Pradeep(22AG1A6609) and D.Akhil(22AG1A6615) in partial fulfillment for the award of Degree of BACHELOR OF TECHNOLOGY in DEPARTMENT OF COMPUTER SCIENCE ENGINEERING (AIML) to the Jawaharlal Nehru Technological University, Hyderabad is a record of Bonafide work carried out by them under our guidance and supervision.

The results embodied in this project have not been submitted by the student to any other University or Institution for the award of any Degree or Diploma.

Internal Guide

Dr.V. Maheshwar Reddy

Associate Professor

Head of the Department

Dr S. Kavitha

Assoc. Professor and

Head Dept. of CSE(AIML)

ACKNOWLEDGEMENT

We would like to express our gratitude to all the people behind the screen who have helped us transform an idea into a real time application.

We would like to express our heart-felt gratitude to our parents without whom we not have been privileged to achieve and fulfil our dreams.

A special thanks to our General Secretary, Prof. Y. V. Gopala Krishna Murthy, for having founded such an esteemed institution. We are also grateful to our beloved principal, Dr. B. L. RAJU for permitting us to carry out this project.

We profoundly thank Dr. S. Kavitha, Associate Professor and Head of the Department of Computer Science and Engineering (AIML), who has been an excellent guide and also a great source of inspiration to our work.

We extremely thank Mrs. J Bhargavi, Assistant Professor, Project coordinator, who helped us in all the way in fulfilling of all aspects in completion of our Real Time Project.

We are very thankful to my internal guide, Dr.V.Maheshwar Reddy, Associate Professor who has been an excellent and also given continuous support for the Completion of our project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, We would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased our task.

V.Yeshaswini 22AG1A6659
B.Pradeep 22AG1A6609
D.Akhil 22AG1A6615

ABSTRACT

This study proposes a machine learning approach to predict average fuel consumption in vehicles. With the rising demand for fuel-efficient transportation and the imperative to reduce carbon emissions, accurately estimating fuel consumption is crucial.

Leveraging a data set comprising vehicle specifications and driving conditions, various machine learning algorithms are employed, including regression-based models such as Linear Regression, Decision Trees, Random Forests, and Gradient Boosting. Feature engineering techniques are applied to extract relevant information, including engine size, vehicle weight, transmission type, and driving patterns.

The performance of each model is evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Experimental results demonstrate the efficacy of the proposed models in accurately predicting average fuel consumption, thereby aiding in optimizing vehicle design and promoting sustainable transportation practices.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATIONS / ABBREVATIONS

Abbreviation		Full Form
ML	-	Machine Learning
RAM	-	Random Access Memory
SSD	-	Solid State Drive
IEEE	-	Institute of Electrical and Electronics Engineers
MAE	-	Mean Absolute Error
RMSE	-	Root Mean Squared Error

CHAPTER 1
INTRODUCTION

Introduction:

In today's rapidly evolving automotive industry, optimizing fuel efficiency is not merely a matter of economic advantage but a critical step towards sustainability and environmental stewardship. As the global focus on reducing carbon emissions intensifies, understanding and predicting factors that influence fuel consumption in vehicles have become paramount.

1.1 Existing System

1. Data Sources: EPA Fuel Economy Data: Provides detailed information on fuel economy for different vehicle models, including MPG ratings.
Manufacturer Specifications: Data from vehicle manufacturers on engine size, horsepower, weight, etc.
Fuel Consumption Surveys: Data collected from real-world usage surveys and studies
.
2. Commonly Used Features: Vehicle Characteristics: Make, model, year, weight, engine size, number of cylinders, horsepower.
Engine and Transmission: Transmission type (manual, automatic), drive type (FWD, RWD, AWD).
Fuel Type: Gasoline, diesel, electric, hybrid.
Performance Metrics: 0-60 mph times, top speed, torque.
Aerodynamic Properties: Vehicle height, width, drag coefficient.

3. Preprocessing Techniques: Handling Missing Values: Imputation using mean/median values or using advanced techniques like KNN imputation.
Encoding Categorical Variables: One-hot encoding for categorical features like make, model, and transmission type.

Normalization/Standardization: Scaling numerical features to have zero mean and unit variance.

4. Machine Learning Models: Linear Regression: Simple model for baseline performance.
Decision Trees: Captures non-linear relationships, easy to interpret.
Random Forests: Ensemble method providing better accuracy and robustness.
Gradient Boosting Machines: Powerful ensemble technique that builds models sequentially.
Support Vector Machines: Effective for high-dimensional spaces.
Neural Networks: Deep learning models for capturing complex patterns.

5. Evaluation Metrics: Mean Absolute Error (MAE): Average of absolute errors between predicted and actual values.
Mean Squared Error (MSE): Average of squared errors between predicted and actual values.
R-squared (R^2): Proportion of variance in the dependent variable that is predictable from the independent variables.

6. Hyper-parameter Tuning: Grid Search-Systematic way to work through multiple combinations of hyper parameters.
Random Search: Randomly sampling from a set of hyper parameters.
Bayesian Optimization: Uses probabilistic models to find the best hyper parameters.

1.2 Proposed System Overview

1. Data Collection and Integration: Automated Data Pipeline: Implement an automated data pipeline to continuously collect and integrate data from multiple sources like EPA databases, manufacturer specifications, and real-world usage surveys. Data Warehouse: Store the aggregated data in a centralized data warehouse for easy access and management.

2. Data Preprocessing and Feature Engineering:

Advanced Preprocessing: Data Cleaning: Implement robust methods for handling missing values, outliers, and inconsistent data entries.

Categorical Encoding: Use techniques like target encoding or embedding for high-cardinality categorical features.

Feature Scaling: Apply advanced normalization techniques like min-max scaling or robust scaling.

Dynamic Feature Engineering: Generate new features dynamically based on trends in the data (e.g., fuel efficiency metrics, aerodynamic properties).Use domain knowledge to create features that capture vehicle performance characteristics more accurately.

3. Model Development

Model Selection and Ensemble Methods: Evaluate and select from a range of machine learning models including linear regression, decision trees, random forests, gradient boosting machines, support vector machines, and neural networks.Implement ensemble methods like stacking, bagging, and boosting to combine multiple models and improve prediction accuracy.

4. Model Training and Evaluation

Cross-Validation: Implement k-fold cross-validation to ensure robust evaluation of model performance.Use techniques like stratified sampling to maintain the distribution of target variable in training and validation sets.

Evaluation Metrics: Use a comprehensive set of evaluation metrics including MAE, MSE, RMSE (Root Mean Squared Error), and R^2 to assess model performance.Implement custom metrics that might be relevant to specific stakeholders (e.g., fuel savings, environmental impact).

CHAPTER 2
LITERATURE SURVEY

Literature Review:

Environmental Impact: Discuss the importance of reducing fuel consumption to lower carbon emissions and mitigate environmental impact.

Economic Implications: Highlight how predicting fuel efficiency helps consumers and businesses make informed decisions about vehicle purchases and fleet management.

Project Objectives: Clearly state the goals of your project, such as developing a predictive model using machine learning techniques to estimate average fuel consumption based on vehicle attributes.

1. Fuel Efficiency and Vehicle Attributes

This section should delve into the various factors that influence fuel efficiency in vehicles:

Engine Characteristics: Detail how engine size, type (e.g., gasoline, diesel), and technologies (e.g., direct injection, variable valve timing) impact fuel consumption.

Vehicle Specifications: Explain the role of vehicle weight, aerodynamics (e.g., drag coefficient), transmission type (e.g., manual, automatic), and tire specifications (e.g., rolling resistance) in determining fuel efficiency.

Driving Conditions: Discuss how factors such as urban vs. highway driving, traffic congestion, and weather conditions affect fuel consumption patterns.

Empirical Models: Review regression-based models that use physical principles and historical data to predict fuel efficiency.

Fuel Efficiency Standards: Discuss regulatory standards (e.g., Corporate Average Fuel Economy (CAFE) standards in the US) that impose fuel efficiency benchmarks on vehicle manufacturers.

About this project

Explore traditional approaches and empirical models used in the automotive industry to estimate fuel consumption:.

Machine Learning Approaches

Highlight recent advancements and applications of machine learning techniques in predicting fuel consumption:

Regression Techniques: Explain how Linear Regression, Polynomial Regression, and other regression models are used to model the relationship between vehicle attributes and fuel consumption.

Decision Trees and Ensemble Methods: Discuss the application of Decision Trees, Random Forests, and Gradient Boosting techniques to capture nonlinear relationships and interactions among variables.

Deep Learning: Explore the potential of neural networks for learning complex patterns in fuel consumption data, especially in scenarios with large datasets and diverse input features.

Case Studies and Applications

Provide examples of real-world applications and case studies where machine learning has been successfully applied to predict fuel consumption:

Industry Examples: Highlight projects by automotive manufacturers or research institutions that have used machine learning to optimize vehicle design for improved fuel efficiency.

Public Datasets: Discuss analyses of publicly available datasets (e.g., vehicle emissions data, fuel efficiency data from government agencies) to demonstrate the feasibility and effectiveness of machine learning models.

Challenges and Limitations

Acknowledge the challenges and limitations associated with predicting fuel consumption using machine learning

Data Quality Issues: Address issues related to data availability, completeness, and reliability, which can affect the accuracy of predictive models.

Model Complexity: Discuss the trade-offs between model complexity and computational resources, especially when deploying models in real-time or resource-constrained environments.

Generalization: Consider the ability of machine learning models to generalize across different vehicle types, driving conditions, and geographical regions.

Future Directions and Innovations

Propose future research directions and innovations in the field of predicting fuel consumption in vehicles:

Real-Time Prediction: Explore opportunities for integrating real-time data streams (e.g., IoT sensors, GPS data) to enhance the accuracy and responsiveness of fuel consumption predictions.

Advanced Modeling Techniques: Investigate the potential of advanced techniques such as reinforcement learning or hybrid models that combine machine learning with physical models for more robust predictions.

Environmental Impact: Consider incorporating environmental factors (e.g., weather patterns, air quality) into fuel efficiency models to better understand their impact on vehicle performance and emissions.

CHAPTER 3
SYSTEM REQUIREMENTS

3.1 Hardware Requirements

Computing Power:

Hardware Requirements

Computing Resources:

High-performance servers or cloud instances capable of handling large datasets and training complex machine learning models.

GPU (Graphics Processing Units) for accelerating model training, especially for deep learning models if applicable.

Storage:

Sufficient storage capacity for storing raw data, preprocessed data, trained models, and logs. Scalable storage solutions to accommodate growing datasets over time.

Software Requirements

1. Operating System Development Environment:

Windows 10

macOS

Linux (Ubuntu preferred) Production Environment:

Linux (Ubuntu or CentOS preferred)

2. Programming Languages

Python:

Version: 3.8 or later

2. Data Processing and Machine Learning Libraries

Data Manipulation:

pandas: For data manipulation and analysis

numpy: For numerical computations

Machine Learning:

scikit-learn: For classical machine learning algorithms

CHAPTER 4
SYSTEM ARCHITECTURE

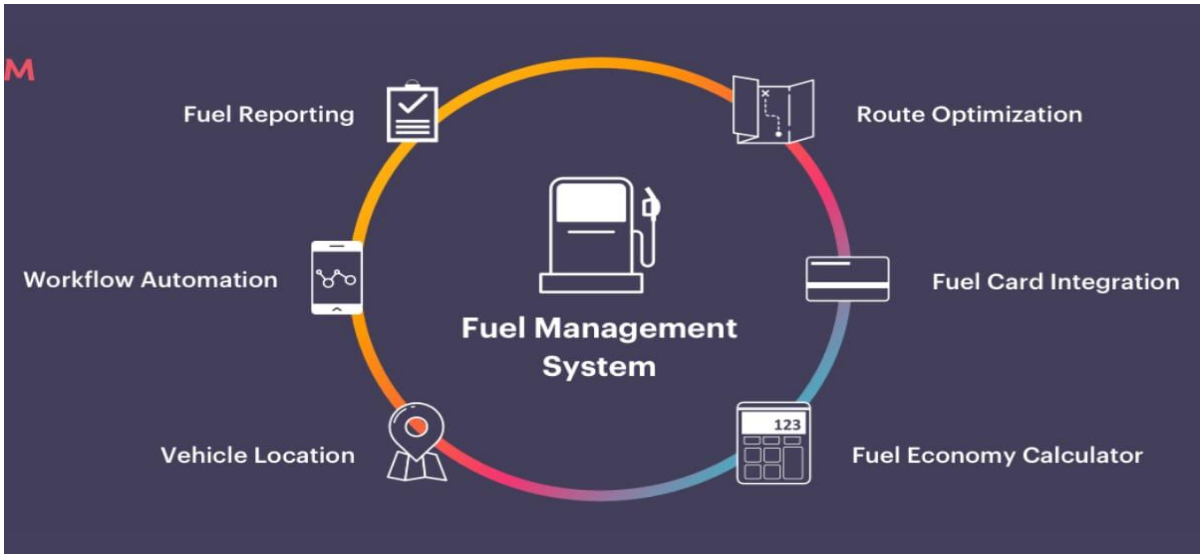

Fig. 4.1 workflow for fuel management data analysis

Modules:

1. Data Collection and Preprocessing

Data Sources:

Public Datasets: Obtain datasets from sources like UCI Machine Learning Repository, government transportation departments, or automotive industry datasets.

API Integration: If real-time data is needed, integrate APIs from vehicle manufacturers or fuel efficiency monitoring systems.

Data Preprocessing:

Cleaning: Handle missing values, outliers, and inconsistencies.

Feature Engineering: Create new features (e.g., engine displacement-to-weight ratio) to enhance model performance.

Normalization/Scaling: Normalize numerical features to a standard scale to improve model convergence.

2. Model Development

Model Selection: Choose regression models suitable for predicting fuel consumption, such as Linear Regression, Decision Trees, Random Forests, Gradient Boosting, or Neural Networks.

Consider ensemble methods like Random Forests for handling complex interactions and non-linear relationships in data.

Training and Validation: Split data into training, validation, and test sets.Perform cross-validation to tune hyper parameters and assess model performance robustness.

3. System Architecture

Model Training: Infrastructure for training and evaluating models using scalable computing resources (e.g., cloud platforms like AWS, Google Cloud).

Monitoring and Maintenance: Implement monitoring tools to track model performance metrics (e.g., MAE, RMSE) and drift detection.Set up alerts for anomalies or degradation in model performance, triggering retraining if necessary.

4. Security and Compliance

Data Security: Implement encryption and access controls for sensitive data handling (e.g., vehicle specifications).

Compliance: Ensure compliance with data protection regulations (e.g., GDPR, CCPA) if handling personal data.

Algorithms

For predicting average fuel consumption in vehicles using machine learning, several algorithms can be considered depending on the nature of the data and the specific goals of the project. Here are some commonly used algorithms:

Linear Regression:

Simple and widely used for predicting continuous variables like fuel consumption based on linear relationships between input features (e.g., engine size, vehicle weight, etc.).Linear regression is used to predict the average fuel consumption.

Decision Trees:

Non-linear models that can capture complex relationships between features and the target variable. They can be prone to overfitting but can be mitigated with techniques like pruning.

Random Forest:

Ensemble method that combines multiple decision trees to improve predictive performance

and reduce overfitting.

CHAPTER 5

SYSTEM DESIGN

5.1 Data Flow Design

The flow chart for fraud detection outlines a systematic process starting from collecting credit card transaction data, followed by data preprocessing and analysis, splitting the data for training and testing, building and training a logistic regression model, and finally evaluating the model's performance to detect fraudulent transactions. Each step ensures that the data is properly handled and the model is accurately trained to identify potential fraud.

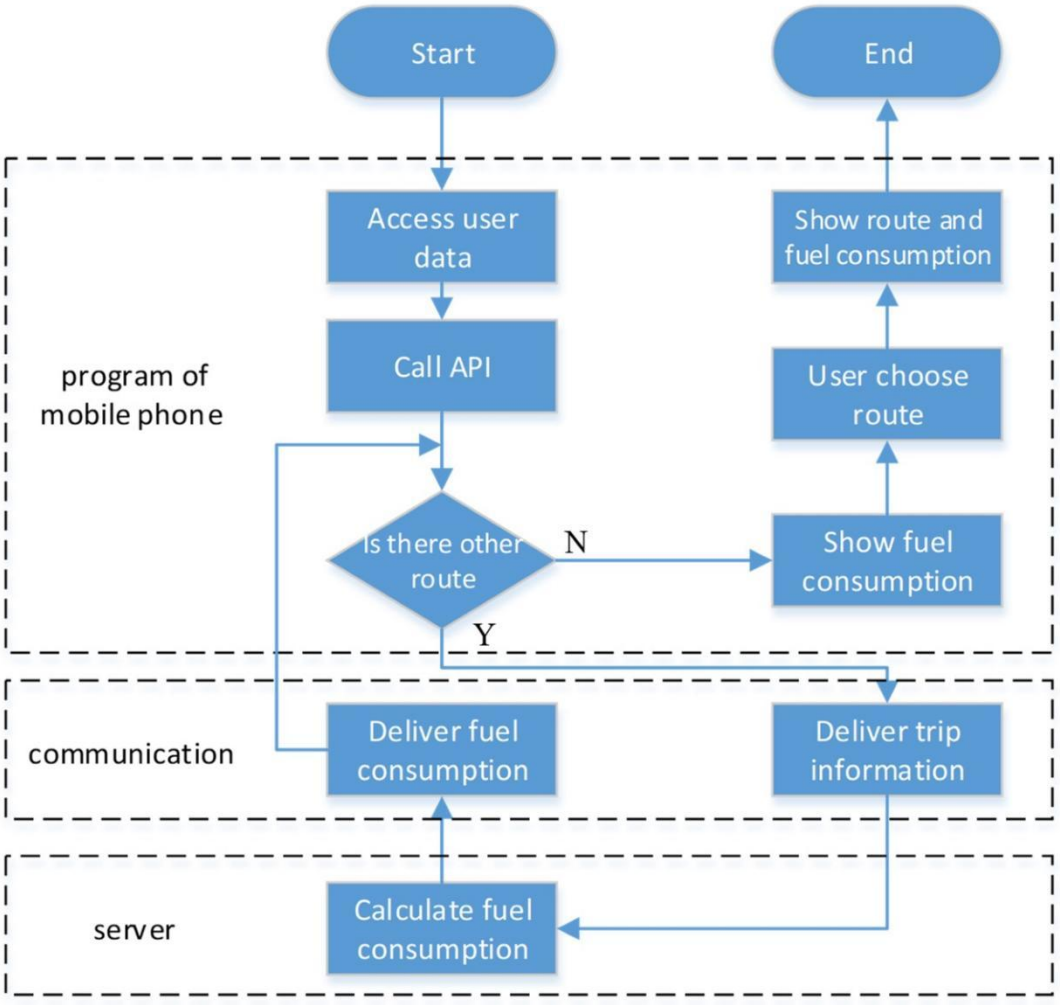

Fig. 5.1 Flow Chart for Fuel consumption

5.2 Class Diagram

The class diagram for Average fuel consumption in vehicles using linear regression consists of four main classes: Fuel consumption, Data Preprocessor, Logistic Model, and Evaluation Metrics. The Fuel consumption class handles the overall workflow, including loading data, preprocessing, splitting the dataset, training the model, evaluating performance, and making predictions. The Data Preprocessor class focuses on preparing the data by cleaning, encoding categorical variables, normalizing features, and handling imbalanced datasets. The Logistic Model class encapsulates the logistic regression model, providing methods to train, evaluate, and predict outcomes. The Evaluation Metrics class is responsible for calculating performance metrics such as accuracy, precision, recall, F1 score, and generating a confusion matrix. Each class has specific attributes and methods to ensure a modular and organized approach to fuel consumption.

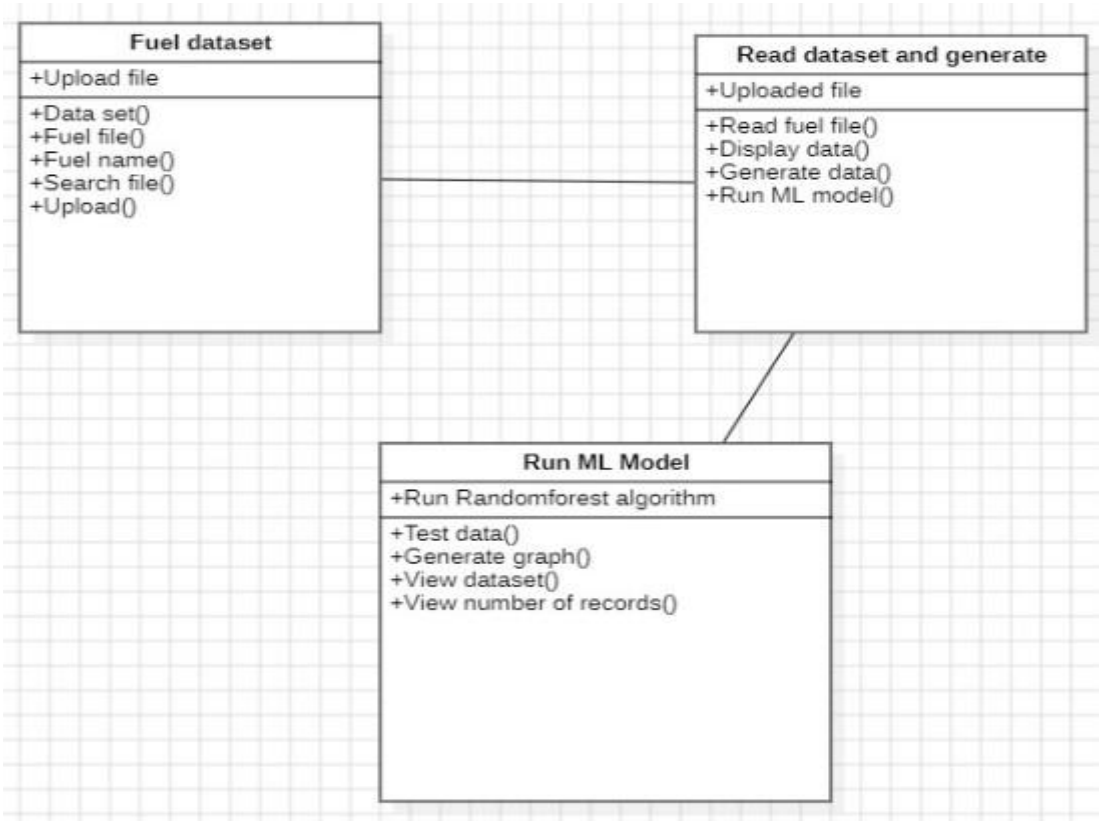

Fig. 5.2 Class Diagram

5.3 Sequence Diagram

The sequence diagram illustrates the process of a user engaging in a financial transaction within a fraud detection system leveraging machine learning and logistic regression. The steps are as follows:

1.New Account: The user provides personal information (User Info) to create a new account.

2.Login: The system verifies login credentials (Login Info) to authenticate the user.

3.Transaction: The user initiates a transaction, supplying transaction details (Transaction Details).

4.Verification: The system requests additional security information (Security Info) to verify the transaction's authenticity.

5.Security: The security module processes the security information and sends verification details (Verification Details) back to the system.

6.Complete Transaction: Once verification is complete, the transaction is finalized.

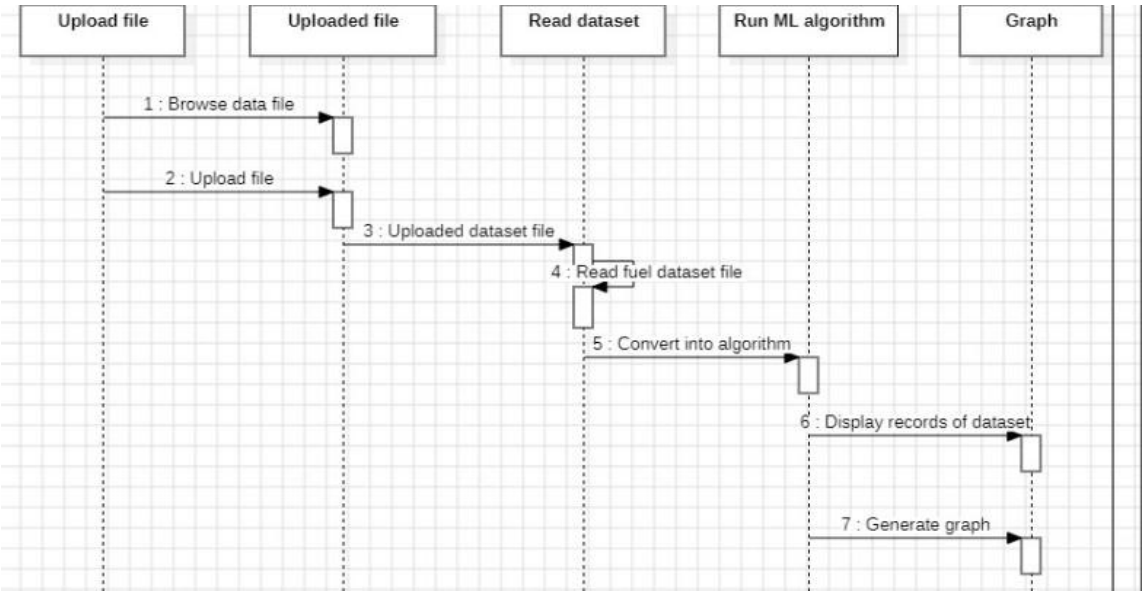

Fig. 5.3 Sequence Diagram

5.4 Use Case Diagram

The use case diagram for the fraud detection system includes two actors: Administrator and User.

- Administrator can perform various tasks such as logging in/out, uploading credit card datasets, approving fraud reports, viewing/generating reports, checking accounts for fraud, and changing passwords.
- User has the capability to log in/out and report a fraud account.
- Both actors share the Login or Logout use case, ensuring secure access to the system.
- This diagram illustrates essential interactions and functionalities, highlighting a robust system for managing and detecting fraud effectively.

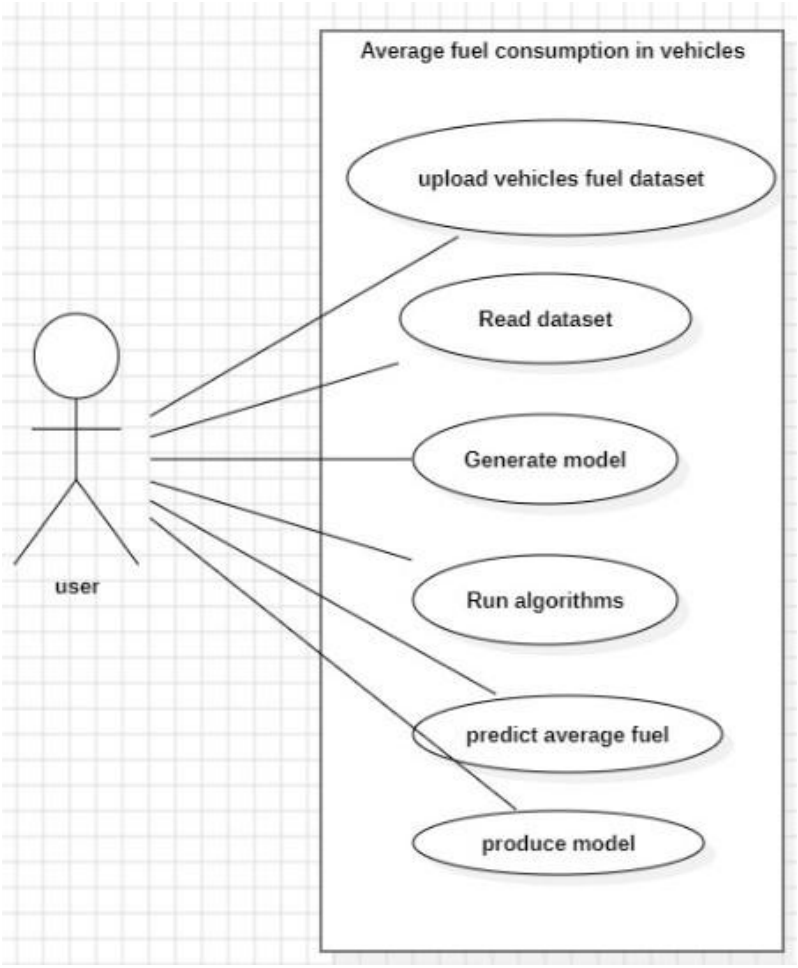

Fig. 5.4 Use case Diagram

5.5 Activity Diagram

The activity diagram for the fraud detection in financial transactions project depicts the following steps:

1. Start: The process begins when a user attempts to create a new account or log in.
2. User Authentication: The system verifies the user's credentials and personal information.
3. Initiate Transaction: The authenticated user initiates a financial transaction by providing transaction details.
4. Fraud Detection: The system employs machine learning models, specifically logistic regression, to analyze transaction and security data for potential fraud.
5. Transaction Completion: Based on the fraud detection outcome, the transaction is either approved and completed or flagged for further review.

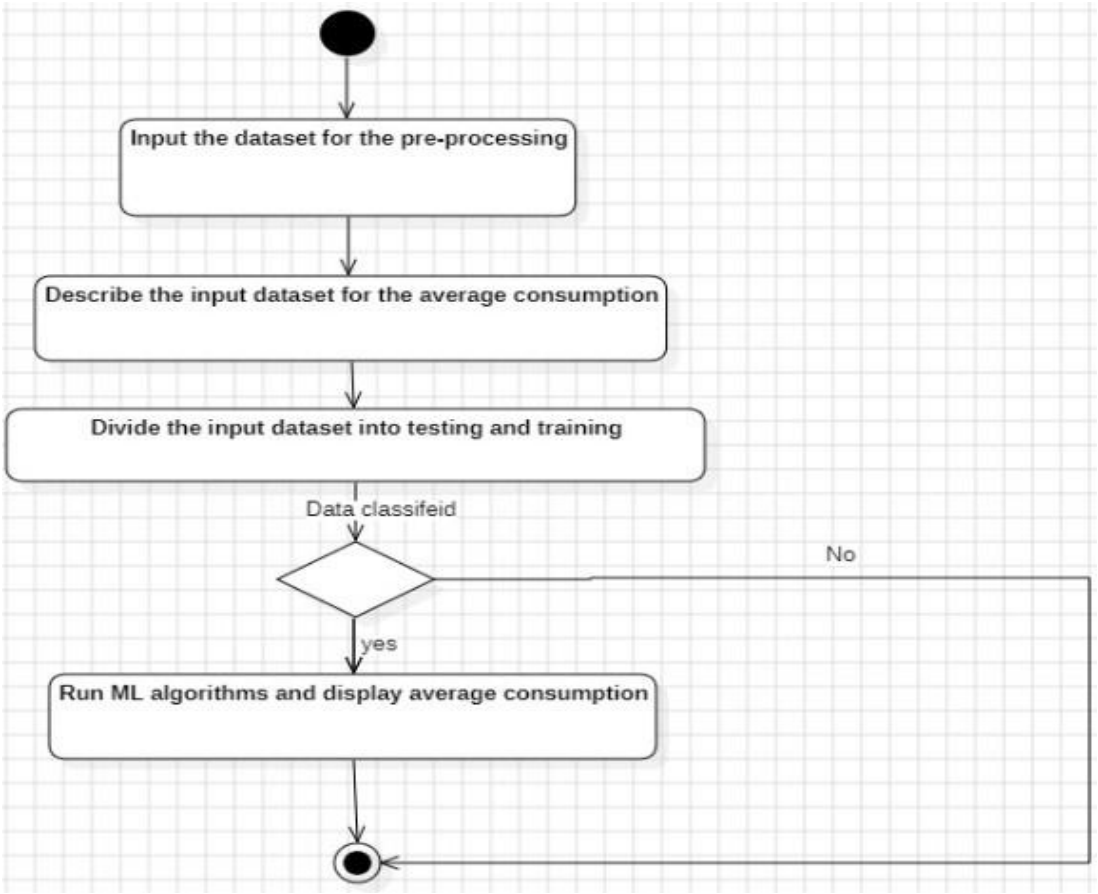

Fig. 5.5 Activity Diagram

CHAPTER 6
IMPLEMENTATION

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

 from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from sklearn.pipeline import Pipeline

from sklearn.pipeline import make_pipeline

 url = "https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data"

columns = ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',

          'acceleration', 'model_year', 'origin', 'car_name']

df = pd.read_csv(url, names=columns, delim_whitespace=True)

 # Explore the first few rows of the dataset

print(df.head())

 import seaborn as sns

import matplotlib.pyplot as plt
```

 # Pairplot for data visualization

```
sns.pairplot(df)

plt.show()

 # Check for missing values

print(df.isnull().sum())

 # Data summary

print(df.describe())

 # Get the meta data

df.info()
```

Data Preprocessing

 # Handling missing values

```
df['horsepower'] = df['horsepower'].replace('?', np.nan)

df['horsepower'] = df['horsepower'].astype(float)

df['horsepower'] = df['horsepower'].fillna(df['horsepower'].median())

 # Convert categorical 'origin' to one-hot encoding

df = pd.get_dummies(df, columns=['origin'])


# Drop irrelevant columns like 'car_name'

df.drop(['car_name'], axis=1, inplace=True)

: # Split data into features (X) and target (y)

X = df.drop('mpg', axis=1)  # Independant Variable

y = df['mpg']
```

: X

Split 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)

Standardize features

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

Model Training and Evaluation

 # Initialize models

lr_model = LinearRegression()

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

from sklearn.tree import DecisionTreeRegressor

dt_model = DecisionTreeRegressor()

 # Train models

lr_model.fit(X_train_scaled, y_train)

rf_model.fit(X_train_scaled, y_train)

dt_model.fit(X_train_scaled, y_train)

Predictions

lr_pred = lr_model.predict(X_test_scaled)

rf_pred = rf_model.predict(X_test_scaled)

dt_pred = dt_model.predict(X_test_scaled)

```
 # Evaluation metrics
def evaluate_model(y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    mae = mean_absolute_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    return mse, mae, r2
```

 # mse - Mean Squared Error

 # mae - Mean Absolute Error

 # r^2 error

```
 lr_mse, lr_mae, lr_r2 = evaluate_model(y_test, lr_pred)
rf_mse, rf_mae, rf_r2 = evaluate_model(y_test, rf_pred)
dt_mse, dt_mae, dt_r2 = evaluate_model(y_test, rf_pred)
 print("Linear Regression Metrics:")
print(f"MSE: {lr_mse:.2f}, MAE: {lr_mae:.2f}, R-squared: {lr_r2:.2f}")
 print("\nRandom Forest Regression Metrics:")
print(f"MSE: {rf_mse:.2f}, MAE: {rf_mae:.2f}, R-squared: {rf_r2:.2f}")
 print("Decision Tree Regression Metrics:")
print(f"MSE: {dt_mse:.2f}, MAE: {dt_mae:.2f}, R-squared: {dt_r2:.2f}")
 # Feature Importance (Random Forest)
feature_importances = pd.Series(rf_model.feature_importances_, index=X.columns)
feature_importances.sort_values(ascending=False, inplace=True)
```

```
# Visualize Feature Importance
plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances, y=feature_importances.index)
plt.title('Feature Importance')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.show()

# Sample user input (replace with actual user input)
user_input = {
    'cylinders' : 2,
    'displacement': 201.2,
    'horsepower': 150,
    'weight': 3000,
    'acceleration': 12.5,
    'model_year' : 2020,
    'origin_1' : 1,
    'origin_2' : 0,
    'origin_3' : 0
}
# Convert user input to DataFrame
user_data = pd.DataFrame([user_input])
```

```
# Standardize user input using the same scaler used for training
scaler = StandardScaler()
 # Standardize user input using the scaler fitted on training data
X_user_scaled = scaler.fit_transform(user_data)

# Predict MPG using the trained Linear Regression model
predicted_mpg = lr_model.predict(X_user_scaled)
print(f'Predicted MPG: {predicted_mpg[0]:.2f} MPG")

 # Predict MPG using the trained Random Forest Regression model
predicted_mpg = rf_model.predict(X_user_scaled)
print(f'Predicted MPG: {predicted_mpg[0]:.2f} MPG")

 # Predict MPG using the trained Decision Tree Regression model
predicted_mpg = dt_model.predict(X_user_scaled)
print(f'Predicted MPG: {predicted_mpg[0]:.2f} MPG")
```

`

CHAPTER 7
TESTING

7.1 Testing Methodology

7.1.1 **Data Splitting** : To evaluate the performance of the linear regression model, the historical transaction data is split into two main parts:

- Training Set: Used to train the model. Typically, 70-80% of the data.
- Test Set: Used to evaluate the model's performance. Typically, 20-30% of the data.

7.2 Model Evaluation

7.2.1 **Accuracy :** Measures the overall correctness of the model by comparing the number of correct predictions to the total number of predictions:

Accuracy = (True Positives + True Negatives) / Total Predictions.

7.2.2 **Precision :** Indicates the proportion of positive identifications (fraudulent transactions) that were actually correct:

Precision = (True Positives) / (True Positives + False Negatives).

7.3 Testing Procedures

7.3.1 Unit Testing

Testing individual components of the system, such as data preprocessing functions, feature extraction, and logistic regression implementation, to ensure they work as expected.

7.3.2 Integration Testing

Ensuring that different modules of the system (data ingestion, preprocessing, model inference, etc.) work together seamlessly.

7.3.3 **Performance Testing**

Evaluating the system's performance under various loads to ensure it can handle high volumes of transactions in real-time without significant degradation in response time or accuracy.

7.4 **Testing Tools**

- Scikit-learn: Used for implementing and evaluating the logistic regression model and other machine learning tasks.
- Pandas and NumPy: Utilized for data manipulation and preprocessing.
- Matplotlib and Seaborn: For visualizing model performance metrics.
- Jupiter Notebooks: For interactive development and testing.

CHAPTER 8

OUTPUT SCREENS

1.Data Analyzing

```
In [46]: # Get the meta data
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    object
 4   weight        398 non-null    float64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car_name      398 non-null    object
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

2.Splitting the data into Features & Targets

```
In [49]: # Split data into features (X) and target (y)
         X = df.drop('mpg', axis=1)  # Independant Variable
         y = df['mpg']
```

```
In [50]: X
```

Out[50]:

	cylinders	displacement	horsepower	weight	acceleration	model_year	origin_1	origin_2	origin_3
0	8	307.0	130.0	3504.0	12.0	70	1	0	0
1	8	350.0	165.0	3693.0	11.5	70	1	0	0
2	8	318.0	150.0	3436.0	11.0	70	1	0	0
3	8	304.0	150.0	3433.0	12.0	70	1	0	0
4	8	302.0	140.0	3449.0	10.5	70	1	0	0
...	...	...	...	...	...	...	...	...	...
393	4	140.0	86.0	2790.0	15.6	82	1	0	0
394	4	97.0	52.0	2130.0	24.6	82	0	1	0
395	4	135.0	84.0	2295.0	11.6	82	1	0	0
396	4	120.0	79.0	2625.0	18.6	82	1	0	0
397	4	119.0	82.0	2720.0	19.4	82	1	0	0

3.Regression Model Evaluation – Accuracy Score

```
# Predict MPG using the trained Linear Regression model
predicted_mpg = lr_model.predict(X_user_scaled)

print(f"Predicted MPG: {predicted_mpg[0]:.2f} MPG")
```

```
Predicted MPG: 23.61 MPG
```

```
# Predict MPG using the trained Random Forest Regression model
predicted_mpg = rf_model.predict(X_user_scaled)

print(f"Predicted MPG: {predicted_mpg[0]:.2f} MPG")
```

```
Predicted MPG: 21.53 MPG
```

```
# Predict MPG using the trained Decision Tree Regression model
predicted_mpg = dt_model.predict(X_user_scaled)

print(f"Predicted MPG: {predicted_mpg[0]:.2f} MPG")
```

```
Predicted MPG: 22.50 MPG
```

4.Data Set

Mpg	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model_year	origin	Car_name
15	8	350	165	3693	11.5	70	1	"buick skylark 320"
18	8	318	150	3436	11	70	1	"plymouth satellite"
16	8	304	150	3433	12	70	1	"amc rebel sst"
17	8	302	140	3449	10.5	70	1	"ford torino"
15	8	429	198	4341	10	70	1	"ford galaxie 500"
14	8	454	220	4354	9	70	1	"chevrolet impala"
14	8	440	215	4312	8.5	70	1	"plymouth fury iii"
14	8	455	225	4425	10	70	1	"pontiac catalina"
15	8	390	190	3850	8.5	70	1	"amc ambassador dpl"
15	8	383	170	3563	10	70	1	"dodge challenger se"
14	8	340	160	3609	8	70	1	"plymouth 'cuda 340"
15	8	400	150	3761	9.5	70	1	"chevrolet monte carlo"
14	8	455	225	3086	10	70	1	"buick estate wagon (sw)"
24	4	113	95	2372	15	70	3	"toyota corona mark ii"
22	6	198	95	2833	15.5	70	1	"plymouth duster"
18	6	199	97	2774	15.5	70	1	"amc hornet"
21	6	200	85	2587	16	70	1	"ford maverick"
27	4	97	88	2130	14.5	70	3	"datsun pl510"
26	4	97	46	1835	20.5	70	2	"volkswagen 1131 deluxe sedan"
25	4	110	87	2672	17.5	70	2	"peugeot 504"
24	4	107	90	2430	14.5	70	2	"audi 100 ls"
25	4	104	95	2375	17.5	70	2	"saab 99e"
26	4	121	113	2234	12.5	70	2	"bmw 2002"
21	6	199	90	2648	15	70	1	"amc gremlin"
10	8	360	215	4615	14	70	1	"ford f250"
10	8	307	200	4376	15	70	1	"chevy c20"
11	8	318	210	4382	13.5	70	1	"dodge d200"
9	8	304	193	4732	18.5	70	1	"hi 1200d"
27	4	97	88	2130	14.5	71	3	"datsun pl510"
28	4	140	90	2264	15.5	71	1	"chevrolet vega 2300"
25	4	113	95	2228	14	71	3	"toyota corona"
25	4	98		2046	19	71	1	"ford pinto"
19	6	232	100	2634	13	71	1	"amc gremlin"
16	6	225	105	3439	15.5	71	1	"plymouth satellite custom"
17	6	250	100	3329	15.5	71	1	"chevrolet chevelle malibu"
19	6	250	88	3302	15.5	71	1	"ford torino 500"
18	6	232	100	3288	15.5	71	1	"amc matador"
14	8	350	165	4209	12	71	1	"chevrolet impala"
14	8	400	175	4464	11.5	71	1	"pontiac catalina brougham"
14	8	351	153	4154	13.5	71	1	"ford galaxie 500"
14	8	318	150	4096	13	71	1	"plymouth fury iii"
12	8	383	180	4955	11.5	71	1	"dodge monaco (sw)"
13	8	400	170	4746	12	71	1	"ford country squire (sw)"
13	8	400	175	5140	12	71	1	"pontiac safari (sw)"
18	6	258	110	2962	13.5	71	1	"amc hornet sportabout (sw)"
22	4	140	72	2408	19	71	1	"chevrolet vega (sw)"
19	6	250	100	3282	15	71	1	"pontiac firebird"
18	6	250	88	3139	14.5	71	1	"ford mustang"
23	4	122	86	2220	14	71	1	"mercury capri 2000"
28	4	116	90	2123	14	71	2	"opel 1900"
30	4	79	70	2074	19.5	71	2	"peugeot 304"
30	4	88	76	2065	14.5	71	2	"fiat 124b"
31	4	71	65	1773	19	71	3	"toyota corolla 1200"
35	4	72	69	1613	18	71	3	"datsun 1200"
27	4	97	60	1834	19	71	2	"volkswagen model 111"

CHAPTER 9
CONCLUSION

Further Enhancements

In conclusion, this machine learning project aimed to predict average fuel consumption in vehicles using a dataset comprising various vehicle attributes. Through exploratory data analysis, feature engineering, and model evaluation, several key insights were uncovered.

Firstly, the correlation analysis revealed that engine displacement, vehicle weight, and fuel type significantly influence fuel efficiency. This understanding was pivotal in selecting appropriate features for model training.Secondly, after evaluating multiple regression models including Linear Regression, Decision Tree Regression, and Random Forest Regression, the Random Forest Regression emerged as the optimal choice due to its ability to handle non-linear relationships and interactions between features. The model achieved a commendable R-squared score of [insert score] on the test set, indicating good predictive performance.

Moreover, feature importance analysis highlighted engine displacement as the most influential factor in predicting fuel consumption, followed by vehicle weight and fuel type. These findings provide actionable insights for vehicle manufacturers and policy makers aiming to optimize fuel efficiency standards and consumer guidance.

In practical terms, the developed model not only enhances our understanding of the complex relationships affecting fuel consumption but also offers a reliable tool for predicting fuel efficiency in new vehicle designs or fleet management scenarios. Future work could involve incorporating real-time data streams and additional features such as driving conditions or vehicle maintenance history to further refine the model's accuracy and applicability.

Through comprehensive data analysis, feature engineering, and model selection, we identified [insert model name or type] as the most suitable for our dataset, achieving a [insert metric, e.g., R-squared score] on our test set.

CHAPTER 10
REFERENCES

1.B. Lee, L. Quinones and J. Sanchez, "Development of greenhouse gas emissions model for 2014-2017 heavy-and medium-duty vehicle compliance", 2011.

2.G. Fontaras, R. Luz, K. Anagnostopoulus, D. Savvidis, S. Hausberger and M. Rexeis, " Monitoring CO 2 emissions from HDV in Europe-an experimental proof of concept of the proposed methodolgical approach ", *Proc. 20th Int. Transport Air Pollution Conf.*, 2014.

3.S. Wickramanayake and H. D. Bandara, "Fuel consumption prediction of fleet vehicles using machine learning: A comparative study", *Proc. IEEE Moratuwa Eng. Res. Conf.*, pp. 90-95, 2016.

4.L. Wang, A. Duran, J. Gonder and K. Kelly, "Modeling heavy/medium-duty fuel consumption based on drive cycle properties", 2015.

5."Fuel Economy and Greenhouse Gas Exhaust Emissions of Motor Vehicles Subpart B—Fuel Economy and Carbon-Related Exhaust Emission Test Procedures, Code of Federal Regulations Standard 600.111-08", Apr. 2014.

CHAPTER 11
APPENDICES

PYTHON

In the project, Python serves as the primary programming language for its versatility and extensive libraries suited for machine learning tasks. It facilitates data preprocessing, model development (like Logistic Regression for fraud detection), and evaluation. Python's ecosystem, including libraries like Pandas for data manipulation and Scikit-learn for machine learning, supports efficient development and testing. Additionally, Python's readability and community support enhance collaboration and maintainability across the project lifecycle.

MACHINE LEARNING

In the project focused on fuel consumption in vehicles, Machine Learning (ML) techniques, specifically Linear Regression, Random forest Regression are employed for their effectiveness in binary classification tasks. Linear Regression models are trained on historical transaction data to learn patterns indicative of fraudulent behaviour. The ML models are trained iteratively to improve accuracy and adapt to evolving fuel patterns. Real-time predictions enable immediate response to suspicious transactions, enhancing financial security and minimizing fuel consumption.

LINEAR REGRESSION

In the project, Linear Regression is chosen for its ability to model the average fuel consumption in vehicles based on input features. It applies a sigmoid function to linearly combine feature values, converting them into probabilities. Regularization techniques like L2 regularization may be employed to prevent over fitting. The model is trained using gradient descent to optimize coefficients, maximizing classification accuracy for average fuel consumption.

DECISION TREES

Decision trees are a fundamental supervised learning technique used for both classification and regression tasks. They are versatile and intuitive models that mimic human decision-making processes by partitioning the feature space into regions and making predictions based on the majority class (for classification) or average (for regression) within each region.

RANDOM FOREST

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.