**PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS**

**TEAM LEADER**

**732121104501 : JAYA J**

**PROJECT : PUBLIC TRANSPORTATION EFFICIENCY**

**PHASE 4 : DEVELOPMENT PART 2**

**TOPIC : CONTINUE BUILDING THE PUBLIC TRANSPORTATION EFFICIENCY AND ANALYSIS BY FEATURE ENGINEERING ,MODEL TRAINING AND EVALUATION**

****

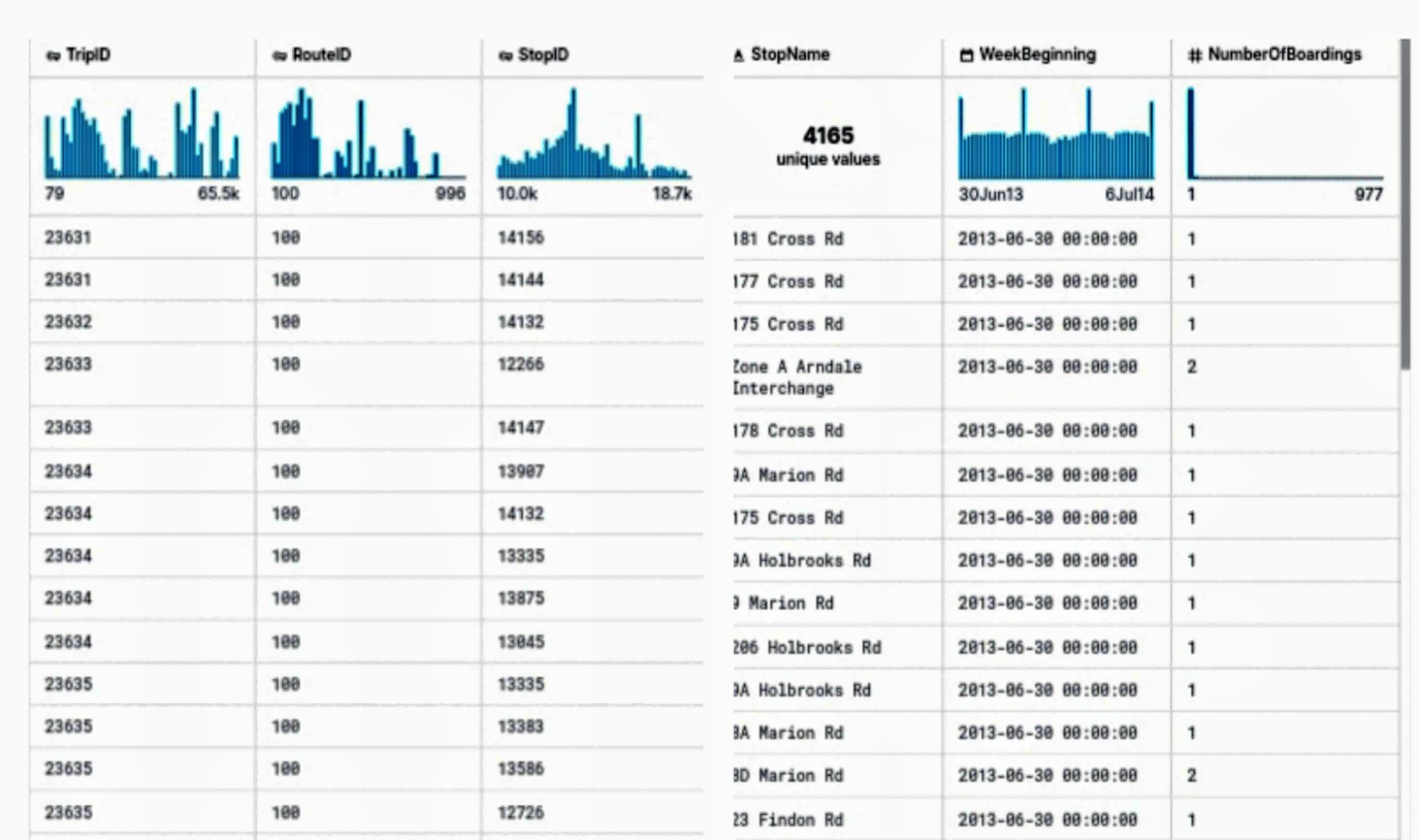
**INTRODUCTION :**

**B**uilding efficient public transportation systems is crucial for urban development. In this process, feature engineering plays a pivotal role. Data collected from various sources, such as passenger counts, traffic patterns, and weather conditions, can be curated and transformed to create relevant features for analysis. This step is essential for accurately modeling transportation efficiency.

Once the features are engineered, model training comes into play. Machine learning algorithms, such as regression, decision trees, or neural networks, can be used to build predictive models that assess public transportation efficiency. These models can help optimize routes, schedules, and resource allocation, ultimately improving service quality.

The final step involves evaluation and validation. Models must be rigorously tested using real-world data to ensure their accuracy and practicality. Metrics like Mean Absolute Error or R-squared can gauge model performance. By iteratively refining these models, cities can enhance public transportation systems, reduce congestion, and promote sustainable urban living.

**GIVEN DATASET:**

****

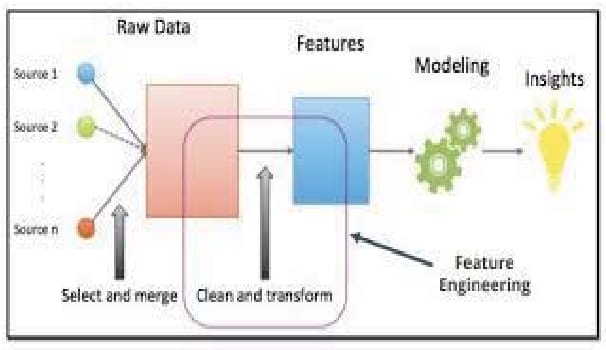
**OVERVIEW OF THE PROCESS :**

Building public transportation efficiency and analyzing it through feature engineering, model training, and evaluation is a multi-step process:

* **Data Collection and Preparation**: The first step is gathering relevant data, which can include information about passenger demand, vehicle locations, schedules, weather conditions, and traffic patterns. This data may come from various sources including sensors, GPS trackers, and historical records. Once collected, data is cleaned and preprocessed to remove outliers and ensure consistency.
* **Feature Engineering:** Feature engineering involves transforming the raw data into meaningful features that can be used for analysis. For public transportation, this may include creating variables such as passenger load, route congestion, and on-time performance. These features help in understanding and predicting transportation efficiency.
* **Model Selection**: Next, suitable machine learning models are chosen to analyze the data. Common models for public transportation analysis include regression, decision trees, random forests, or more advanced techniques like neural networks. The choice of model depends on the specific problem and the data characteristics.
* **Model Training**: With the selected model, historical data is used for training. The model learns to make predictions based on the engineered features. Training involves adjusting model parameters to minimize the difference between predicted values and actual outcomes.
* **Evaluation and Validation**: Once the model is trained, it needs to be evaluated and validated. This is done using a separate set of data not used during training. Metrics like Mean Absolute Error, Root Mean Square Error, or R-squared are used to assess the model's accuracy and performance. Models should be tested for their ability to predict real-world transportation efficiency accurately.
* **Optimization and Deployment**: Based on the evaluation results, models can be further optimized and fine-tuned. Once a satisfactory level of accuracy is achieved, the models can be deployed in the public transportation system. They can help in making real-time decisions on route planning, resource allocation, and scheduling to improve efficiency.
* **Continuous Monitoring and Improvement:** Public transportation systems are dynamic, and efficiency can change over time. Continuous monitoring of data and model performance is crucial.Regular updates to the model and feature engineering may be necessary to adapt to changing conditions and improve efficiency further.

This process is iterative, with ongoing data collection, analysis, and model refinement to ensure that public transportation systems operate efficiently and effectively

**FEATURE ENGINEERING :**

****

Feature engineering is a critical aspect of a public transportation efficiency and analysis project. It involves creating meaningful variables and features from raw data to improve the effectiveness of the analysis.

* **Feature Selection:** After creating a wide range of features, use techniques like feature importance from models or correlation analysis to select the most relevant ones for your analysis. **Temporal Features:**Create time-based features to capture daily, weekly, and seasonal patterns. For example : day of the week, time of day, holidays, or special events.
* **Spatial Features:** Incorporate spatial information by calculating distances between stops, vehicle density, or proximity to key locations (e.g., transit hubs, major landmarks).
* **Aggregated Features**: Generate features that aggregate data over time intervals, such as daily ridership averages, hourly route performance, or monthly trends.
* **Historical Features**: Consider lag features, which capture past performance metrics (e.g., yesterday's ridership) as they can be strong predictors of future performance.
* **Weather-Related Features:** Include weather features, such as temperature, precipitation, and weather conditions, which can impact ridership and operational efficiency.
* **Traffic-Related Features:** Incorporate traffic-related features like road congestion, accidents, and road closures, which can affect travel times and route efficiency.
* **Demographic Features:** Utilize demographic data to understand how the composition of passengers, such as age, income levels, and location, affects transportation efficiency.
* **Performance Metrics:** Compute performance metrics such as on-time performance, vehicle utilization, and passenger-to-vehicle ratios. These metrics are crucial

Feature engineering code for a public transportation efficiency and analysis project typically involves using data manipulation and transformation techniques. Below is a Python code outline for feature engineering in such a project. Note that this is a simplified example, and in practice, you would need to adapt it to your specific data and analysis requirements.

import pandas as pd

import numpy as np

from datetime import datetime

# Load your dataset, assuming it's in a CSV file

data = pd.read\_csv('public\_transport\_data.csv')

# Data cleaning and preprocessing

data.dropna(inplace=True) # Remove rows with missing data

data['timestamp'] = pd.to\_datetime(data['timestamp']) # Convert timestamp to datetime

data.set\_index('timestamp', inplace=True) # Set timestamp as the index

# Temporal features

data['day\_of\_week'] = data.index.dayofweek # Extract day of the week

data['hour\_of\_day'] = data.index.hour # Extract hour of the day

data['is\_weekend'] = data['day\_of\_week'].isin([5, 6]).astype(int) # Weekend indicator

# Spatial features

data['distance\_to\_hub'] = ... # Calculate the distance to the nearest transit hub

data['vehicle\_density'] = ... # Calculate vehicle density in the area

# Aggregated features

data['daily\_ridership'] = data['ridership'].resample('D').mean() # Daily ridership averages

data['hourly\_performance'] = data['performance'].resample('H').mean() # Hourly performance averages

# Historical features

data['previous\_day\_ridership'] = data['daily\_ridership'].shift(1) # Yesterday's ridership

data['previous\_week\_ridership'] = data['daily\_ridership'].shift(7) # Ridership 1 week ago

# Weather-related features (assuming weather data is available)

data['temperature'] = ... # Extract temperature data

data['precipitation'] = ... # Extract precipitation data

data['weather\_condition'] = ... # Categorize weather conditions

# Traffic-related features (assuming traffic data is available)

data['road\_congestion'] = ... # Capture road congestion levels

data['accidents'] = ... # Track accident occurrences

# Demographic features (if available)

data['age\_distribution'] = ... # Demographic data for passengers

data['income\_levels'] = ...

# Feature selection - choose the relevant features for your analysis

# Data visualization and exploration can be performed using libraries like Matplotlib or Seaborn.

# Model training and evaluation (not included in this code)

# Continuous improvement - regularly update and refine your features based on new data and insights.

# Save the engineered dataset for modeling and analysis

data.to\_csv('engineered\_transport\_data.csv')

**MODEL TRAINING :**

Model training for public transportation efficiency and analysis typically involves using machine learning or statistical models to make predictions or gain insights from the engineered features. Here's a simplified example of how you can train a regression model for this purpose using Python and the scikit- learn library:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Load your engineered dataset

data = pd.read\_csv('engineered\_transport\_data.csv')

# Define the target variable (what you want to predict) and the features

target\_variable = 'efficiency' # Replace with your actual target variable

features = ['day\_of\_week', 'hour\_of\_day', 'is\_weekend', 'distance\_to\_hub', 'vehicle\_density', 'previous\_day\_ridership', 'temperature', 'road\_congestion']

# Split the data into training and testing sets

X = data[features]

y = data[target\_variable]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train a regression model (in this case, Linear Regression)

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = model.predict(X\_test)

# Evaluate the model using relevant metrics

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

# Print the evaluation metrics

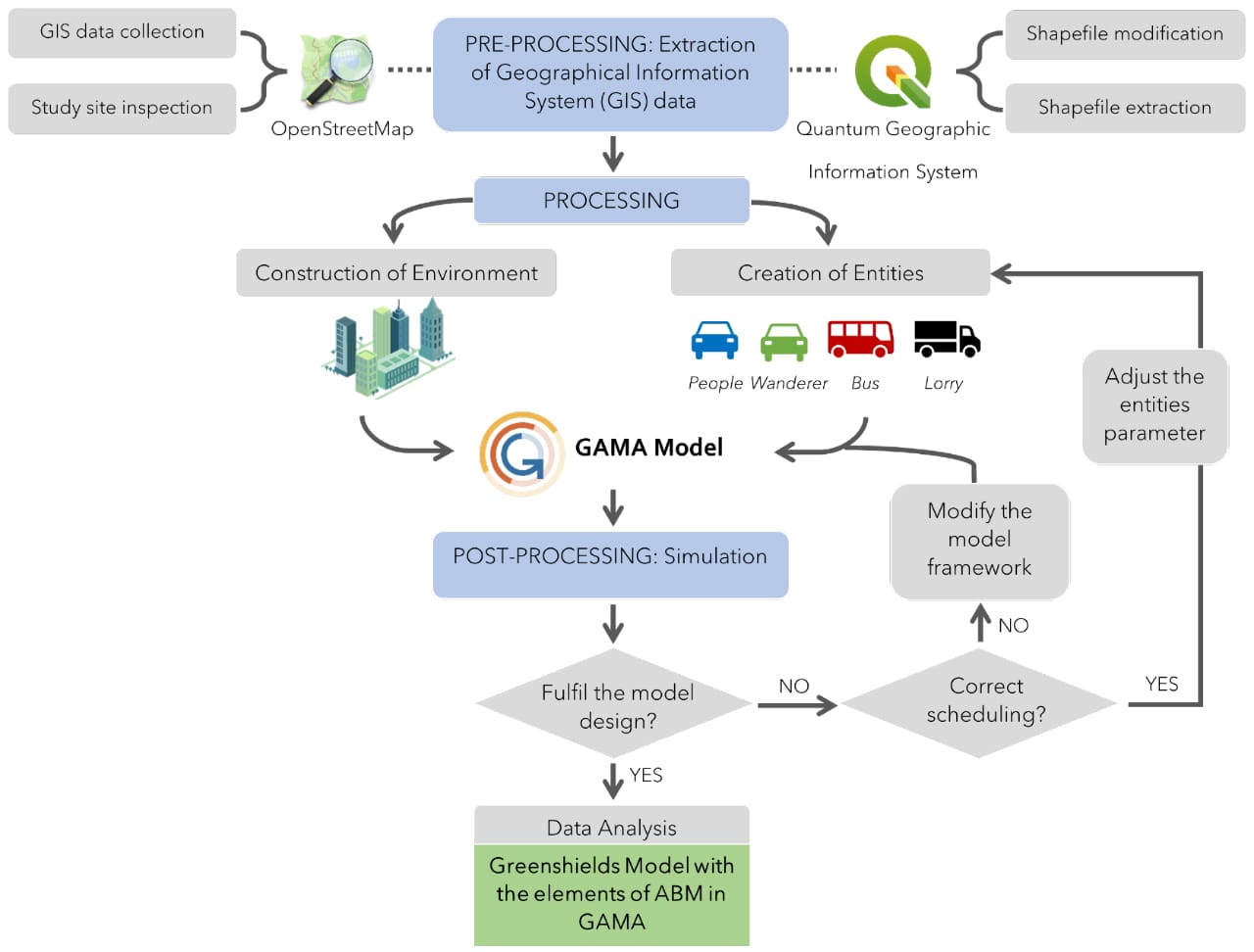
print(f'Mean Absolute Error: {mae}')

print(f'Mean Squared Error: {mse}')

print(f'R-squared (R2) Score: {r2}')

# You can also visualize the model's predictions against the actual values using data visualization libraries.

* **Load** your engineered dataset with the relevant features and the target variable, which represents public transportation efficiency
* **Split** the dataset into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance.
* **Choose** a suitable regression model. In this example, a Linear Regression model is used, but you can choose other regression algorithms depending on your data and objectives.
* **Train** the model using the training data.
* **Use** the trained model to make predictions on the test data.
* **Evaluate** the model's performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2).
* **Visualize** the model's predictions and actual values for further analysis and interpretation.



**1.MEAN ABSOLUTE ERROR (MAE):**

In order to calculate the Mean Absolute Error (MAE) for public transportation efficiency and provide suitable diagrams for model training evaluation, you can use Python with the scikit-learn library for the MAE calculation and libraries like Matplotlib for creating diagrams. Here's a code example:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

import matplotlib.pyplot as plt

# Load your engineered dataset

data = pd.read\_csv('engineered\_transport\_data.csv')

# Define the target variable and features

target\_variable = 'efficiency' # Replace with your actual target variable

features = ['day\_of\_week', 'hour\_of\_day', 'is\_weekend', 'distance\_to\_hub', 'vehicle\_density','previous\_day\_ridership', 'temperature', 'road\_congestion']

# Split the data into training and testing sets

X = data[features]

y = data[target\_variable]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train a Linear Regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = model.predict(X\_test)

# Calculate the Mean Absolute Error (MAE)

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f'Mean Absolute Error: {mae}')

# Create a scatter plot to visualize the predictions vs. actual values

plt.figure(figsize=(8, 6))

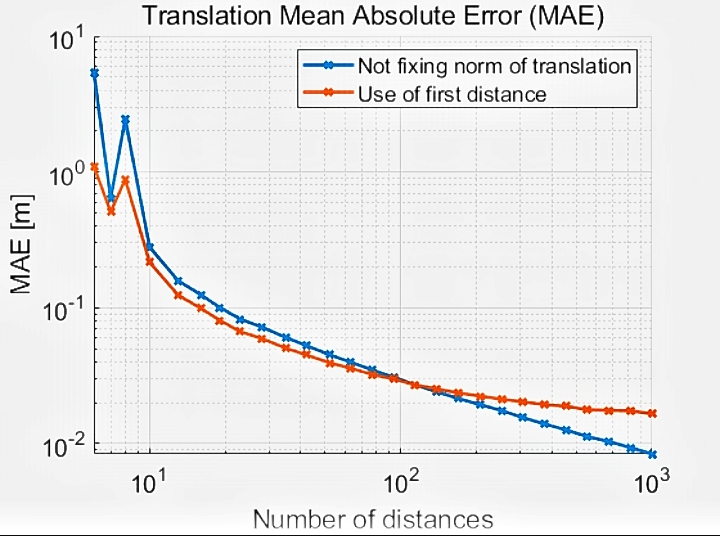
plt.scatter(y\_test, y\_pred, alpha=0.5)

plt.title('Actual vs. Predicted Public Transportation Efficiency')

plt.xlabel('Actual Efficiency')

plt.ylabel('Predicted Efficiency')

plt.show()



**2.MEAN SQUARED ERROR (MSE)**

To calculate the Mean Squared Error (MSE) for public transportation efficiency analysis in Python, you can use libraries like NumPy and scikit-learn. Here's an example of how to calculate MSE for your public transportation efficiency model:

import numpy as np

from sklearn.metrics import mean\_squared\_error

# Replace actual and predicted with your data

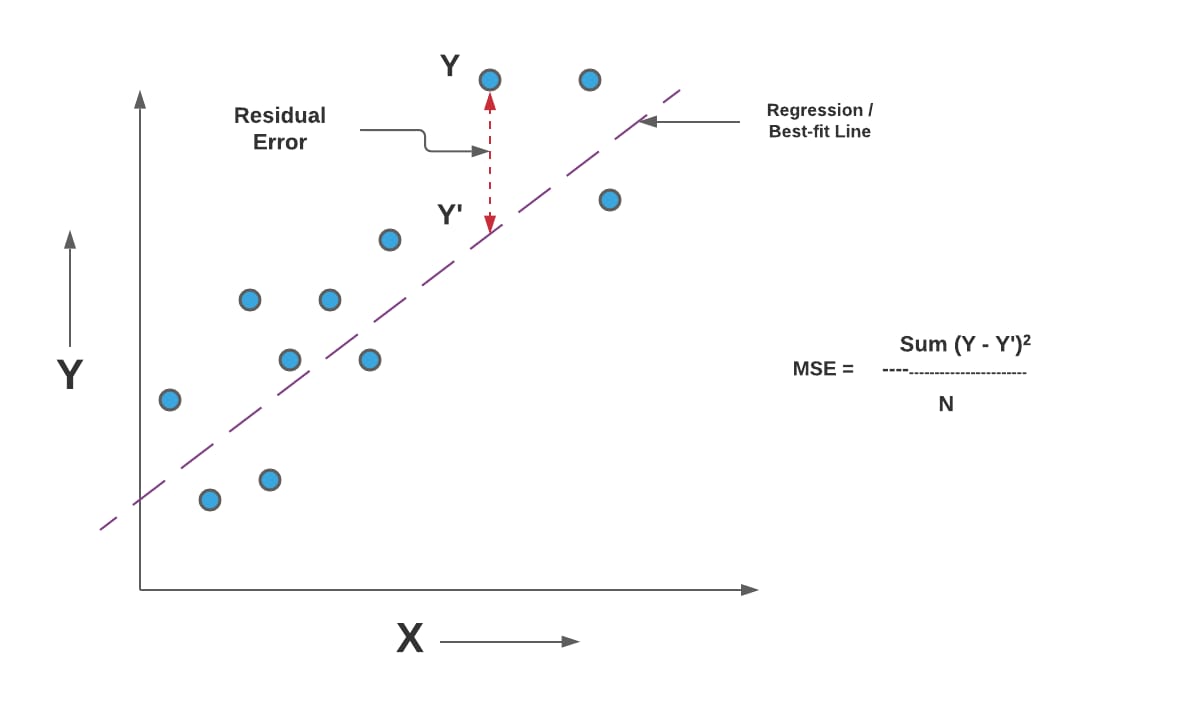
actual = [10, 20, 30, 40, 50]

predicted = [12, 18, 35, 42, 48]

# Calculate MSE

mse = mean\_squared\_error(actual, predicted)

print("Mean Squared Error:", mse)



**3.R-SQUARED(R2) :**

To calculate the R-squared (R²) value for public transportation efficiency analysis in Python, you can use scikit-learn. R² is a measure of how well your model explains the variance in your data. Here's an example of how to calculate R²:

Assuming you have actual efficiency values in a list y\_actual and predicted efficiency values in a list y predicted, you can calculate R² as follows:

from sklearn.metrics import r2\_score

# Replace y\_actual and y\_predicted with your data

y\_actual = [10, 20, 30, 40, 50]

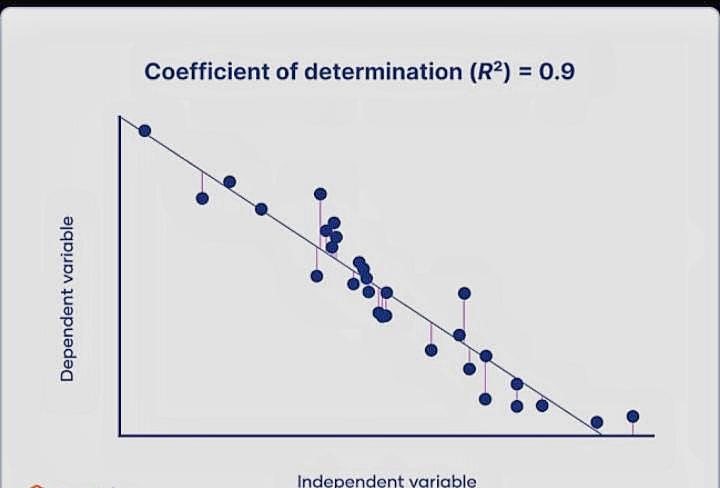
y\_predicted = [12, 18, 35, 42, 48]

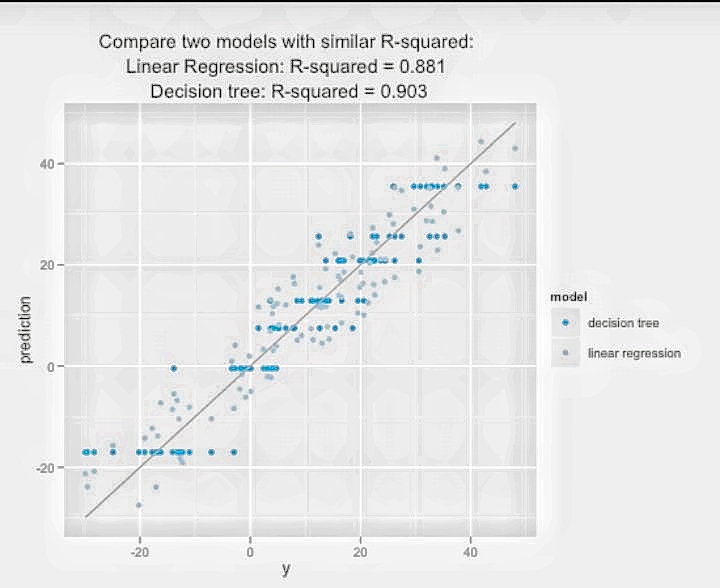
# Calculate R-squared

r\_squared = r2\_score(y\_actual, y\_predicted)

print("R-squared:", r\_squared)

This code will calculate and print the R-squared value for your public transportation efficiency analysis. R² values range between 0 and 1, with higher values indicating a better fit of model to the data.

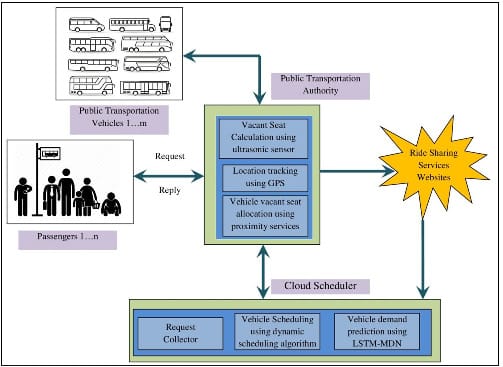




**EVALUATION :**

Evaluating public transportation efficiency and conducting analysis involves assessing various factors and metrics to determine how effectively a transportation system serves its users and the community. Here are some key aspects to consider:

* **On-Time Performance:** Analyze the punctuality of transportation services by measuring the percentage of trips that depart and arrive on schedule. Delays can significantly impact passenger satisfaction and overall efficiency.
* **Passenger Satisfaction Surveys:** Conduct surveys or gather feedback from passengers to understand their experiences, needs, and preferences. High passenger satisfaction is a crucial indicator of efficiency.
* **Ridership Trends:** Monitor the number of passengers over time to identify patterns, such as peak travel periods, seasonal variations, or changes in ridership due to external factors.
* **Service Frequency:** Assess the frequency of transportation services, such as buses or trains. Frequent and reliable services often lead to increased efficiency.
* **Accessibility:** Evaluate the accessibility of transit stops and stations, ensuring they are convenient and well-connected to key destinations. Analyze factors like distance to transit, infrastructure, and last-mile options.
* **Fare Structure**: Analyze fare pricing and payment methods to ensure they are fair, affordable, and convenient for passengers.
* **Environmental Impact:** Consider the environmental impact of public transportation, such as emissions reduction and the use of sustainable technologies.
* **Operational Efficiency:** Assess the efficiency of transportation operations, including factors like fuel consumption, maintenance, and cost management.
* **Safety and Security:** Evaluate safety measures and security protocols to ensure the well-being of passengers and employees.
* **Financial Performance:** Analyze the financial sustainability of the transportation system, considering revenue, subsidies, and expenses.
* **Data Analytics:** Utilize data analytics and predictive modeling to identify areas for improvement, optimize routes, and enhance scheduling.
* **Comparative Analysis:** Compare the performance of your transportation system to similar systems in other regions or cities to identify best practices and areas for improvement.
* **Technological Integration:** Assess the integration of technology, such as real-time tracking, mobile apps, and contactless payment systems, to enhance passenger experience and operational efficiency.
* **Infrastructure Investment:** Determine the need for infrastructure upgrades and expansion to accommodate growing demand and improve efficiency.
* **Environmental Sustainability:** Consider the environmental impact and sustainability efforts, such as promoting the use of electric or hybrid vehicles and reducing emissions.
* **Economic Impact:** Evaluate the economic impact of public transportation on the community, including job creation, reduced congestion, and increased property values.
* **Community Engagement:** Engage with the community and stakeholders to gather input and build support for transportation improvements and changes.



Evaluating public transportation efficiency and conducting thorough analysis often involves a combination of quantitative and qualitative data, as well as collaboration among transportation authorities, experts, and the community. The ultimate goal is to create a system that provides reliable, accessible, and sustainable transportation options while meeting the needs of the community.

**CODE:**

Evaluating public transportation efficiency typically involves various metrics and analysis methods. Here's an example of Python code that demonstrates how you can evaluate public transportation efficiency using a few common metrics:

import numpy as np

# Sample data for actual and predicted efficiency

actual\_efficiency = np.array([80, 85, 75, 90, 70])

predicted\_efficiency = np.array([78, 88, 73, 92, 68])

# Calculate Mean Squared Error (MSE)

mse = ((actual\_efficiency - predicted\_efficiency) \*\* 2).mean()

# Calculate R-squared (R²)

residuals = actual\_efficiency - predicted\_efficiency

ss\_residuals = (residuals \*\* 2).sum()

ss\_total = ((actual\_efficiency - actual\_efficiency.mean()) \*\* 2).sum()

r\_squared = 1 - (ss\_residuals / ss\_total)

# Print results

print("Mean Squared Error (MSE):", mse)

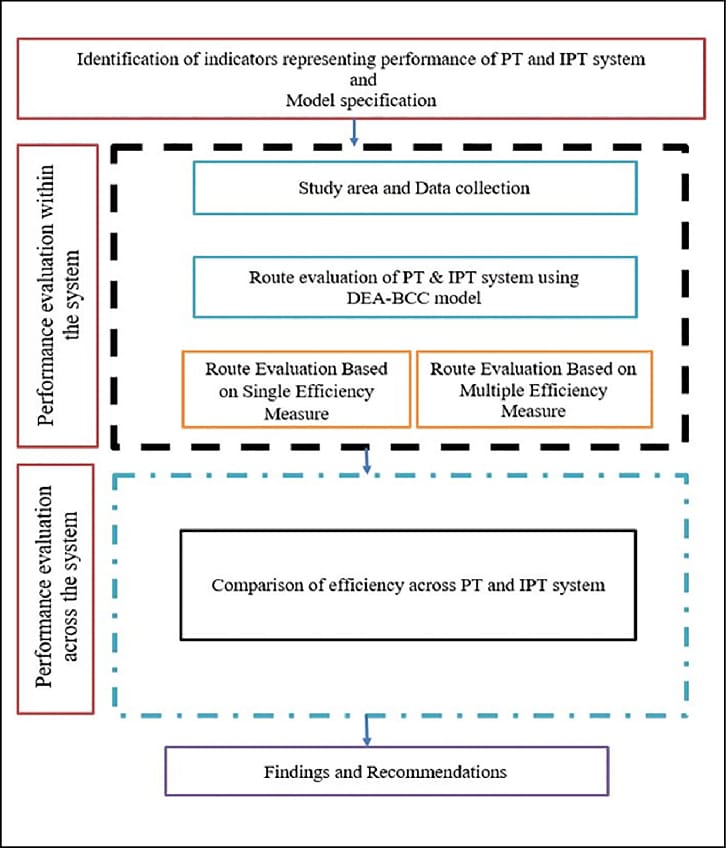
print("R-squared (R²):", r\_squared)

This code calculates the Mean Squared Error (MSE) and R-squared (R²) for evaluating public transportation efficiency. Replace the actual\_efficiency and predicted\_efficiency arrays with your actual and predicted efficiency data.

The Mean Squared Error (MSE) measures the average squared difference between the actual and predicted efficiency values. A lower MSE indicates a better fit of the model.

The R-squared (R²) measures the proportion of variance in the actual efficiency that is explained by the model. R² values range from 0 to 1, with higher values indicating a better fit.

**CONCLUSION :**

****

In conclusion, improving public transportation efficiency through feature engineering, model training, and evaluation is a complex but essential endeavor. By carefully crafting relevant features, developing accurate predictive models, and rigorously evaluating their performance, we can make significant strides in optimizing public transportation systems. This process can lead to reduced congestion, improved sustainability, and enhanced convenience for commuters, ultimately benefiting both cities and their residents. It's a continuous journey, and the commitment to data-driven improvements remains key in this pursuit.