**PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS**

**TEAM MEMBER**

**732121104074 : SRIKANTH L**

**PROJECT : PUBLIC TRANSPORTATION EFFICIENCY**

**PHASE 5 : PROJECT DOCUMENTATION & SUBMISSION**

**TOPIC :** In this section we will document the complete project and prepare it for submission



**ANALYSIS OBJECTIVES:**

 Define specific objectives for analyzing public transportation data, such as assessing on-time performance, passenger satisfaction, and service efficiency.

**DATA SOURSE:**

Identify the sources and methods for collecting transportation data, including schedules, real-time updates, and passenger feedback.

**Dataset Link**: https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV

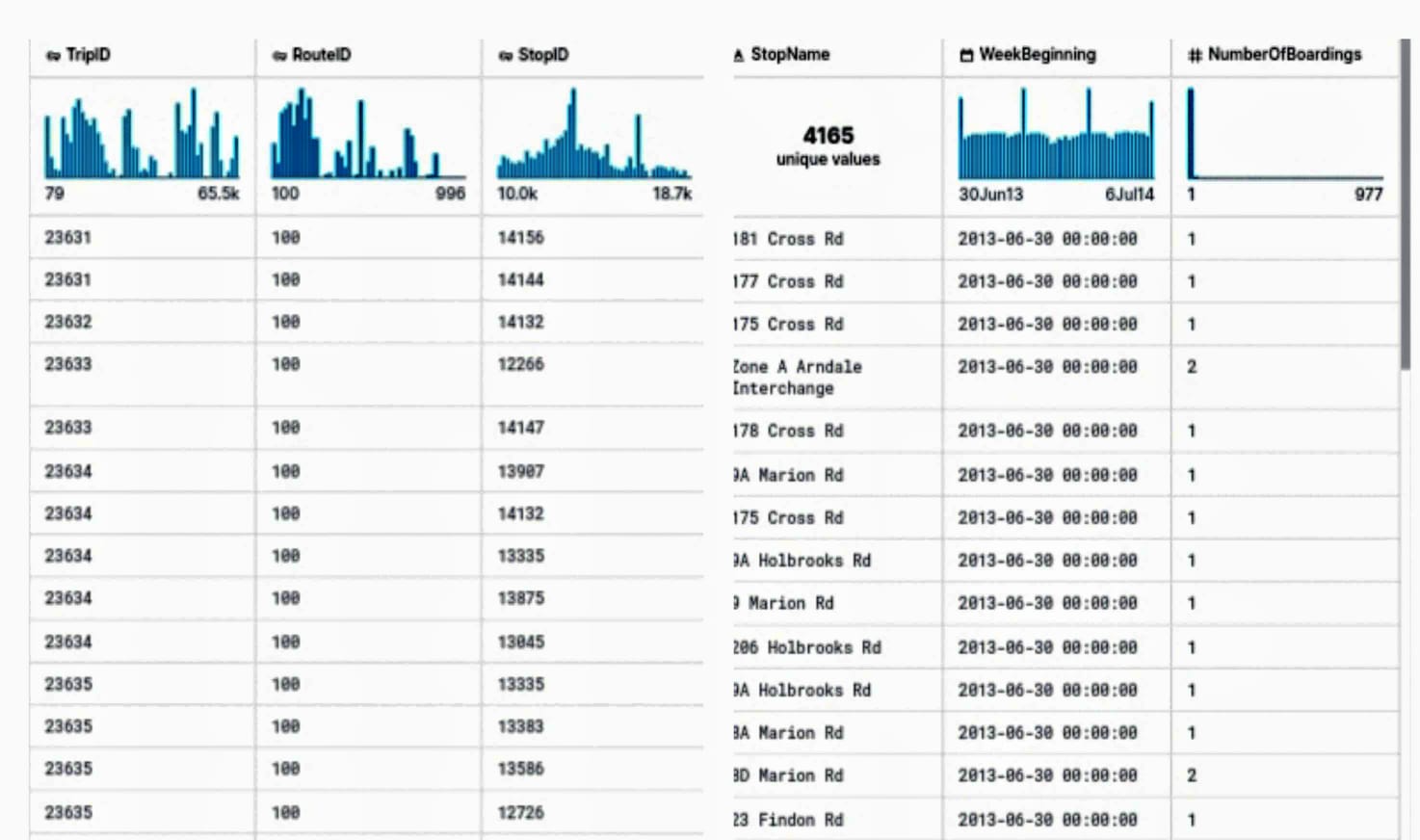
**ABSTRACT:**

Public transportation systems are essential components of modern urban infrastructure, aiming to provide convenient, cost-effective, and sustainable mobility solutions. To enhance the performance of these systems and optimize resource allocation, a comprehensive analysis of their efficiency is crucial. This paper proposes a structured approach to assess public transportation efficiency through various modules, each addressing specific aspects of the system. The analysis modules include ridership analysis, route optimization, resource utilization, environmental impact assessment, and passenger satisfaction evaluation. By systematically evaluating these components, transportation authorities and planners can make informed decisions to improve public transportation services, reduce operational costs, and contribute to the overall well-being of urban communities.

**INTRODUCTION:**

* In an era defined by urbanization and environmental consciousness, the efficiency of public transportation systems has become paramount to the well-being of our cities and the planet. The vitality of any metropolis hinges on its ability to move its citizens seamlessly, sustainably, and economically. This project delves into the heart of this challenge, aiming to assess and enhance the efficiency of public transportation networks to meet the growing demands of the modern world.
* Our analysis is driven by the understanding that efficient public transportation not only reduces traffic congestion, air pollution, and energy consumption but also fosters economic growth, accessibility, and social equity. As our cities expand and evolve, so too must our transit systems adapt and optimize.
* This submission explores the essential components of public transportation efficiency, encompassing route planning, passenger flow management, technology integration, and sustainability practices. By scrutinizing these elements, we seek to identify opportunities for improvement and develop strategies that enhance the overall performance of urban transit systems.
* Through rigorous analysis, data-driven insights, and innovative solutions, we aim to contribute to the advancement of public transportation efficiency, offering a roadmap for cities to create cleaner, more accessible, and ultimately more livable environments for their residents. Our work reflects a commitment to the future of urban mobility, where the journey is as important as the destination.

**GIVEN DATASET:**

****

**PUBLIC TRANSACTION AND EFFICIENCY:**

Public transportation efficiency refers to the ability of a transit system to provide reliable, convenient, and cost-effective services that meet the transportation needs of a community. Efficient public transportation systems should accomplish the following objectives:

**Accessibility:** Provide access to key destinations such as employment centers, educational institutions, healthcare facilities, and recreational areas.

**Reliability:** Offer reliable and consistent services that minimize waiting times and delays for passengers.

**Affordability:** Maintain reasonable fares that are accessible to a broad range of residents, ensuring that public transportation remains an attractive option.

**Safety and Comfort:** Ensure the safety and comfort of passengers, addressing concerns related to security, cleanliness, and passenger experience.

**Environmental Sustainability:** Minimize the environmental impact of transit operations, including reducing emissions and resource consumption

**PROGRAM:**

matplotlib inline

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import datetime

import os

from math import sqrt

import warnings

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

warnings.filterwarnings('ignore')

data = pd.read\_csv('../input/unisys/ptsboardingsummary/20140711.CSV')

data.shape

data.head(10)

**Output** : ( 10857234,6)



out\_geo = pd.read\_csv('../input/outgeo/output\_geo.csv')

out\_geo.shape

out\_geo.head()

**Output:**

(4165, 10)



**EXTERNAL FEATURES:**

from math import sin, cos, sqrt, atan2, radians

def calc\_dist(lat1,lon1):

R = 6373.0

dlon = radians(138.604801) - radians(lon1)

dlat = radians(-34.921247) - radians(lat1)

a = sin (dlat / 2)\*2 + cos(radians(lat1)) \*cos(radians(-34.921247)) \* sin(dlon / 2)\*2

c = 2 \* atan2(sqrt(a), sqrt(1 - a))

return R \* c

out\_geo['dist\_from\_centre']=out\_geo[['latitude','longitude']].apply(lambda x: calc\_dist(\*x), axis=1)

out\_geo.head()

**Output:**



out\_geo['type'].fillna('street\_address',inplace=True)

out\_geo['type'] = out\_geo['type'].apply(lambda x: str(x).split(',')[-1])

out\_geo['type'].unique()

**Output:**

array(['street\_address', 'transit\_station', 'premise', 'political',

'school', 'route', 'intersection', 'point\_of\_interest',

'subpremise', 'real\_estate\_agency', 'university', 'travel\_agency',

'restaurant', 'supermarket', 'store', 'post\_office'], dtype=object)

**Input:**

data['WeekBeginning'] = pd.to\_datetime(data['WeekBeginning']).dt.date

data['WeekBeginning'][1]

**Output:**

datetime.date(2013, 6, 30)

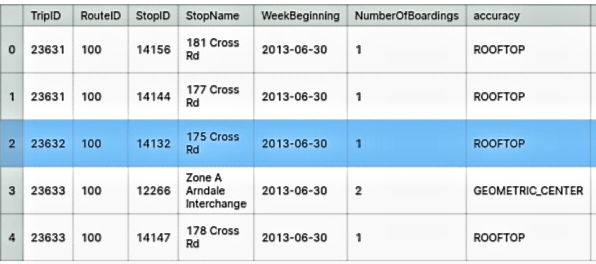
**DATA AGGREGATION:**

data= pd.merge(data,out\_geo,how='left',left\_on = 'StopName',right\_on = 'input\_string')

data.head(5)

data.shape

**Output:**

****

(10857234, 17)

**Input:**

col=['TripID','RouteID','StopID','StopName','WeekBeginning','NumberOfBoardings', 'latitude', 'longitude','postcode','type','dist\_from\_centre']

data = data[col]

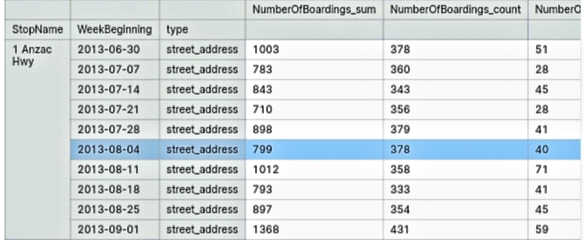
grouped = data.groupby(['StopName','WeekBeginning','type'])

grouped.columns = ["\_".join(x) for x in grouped.columns.ravel()]

grouped.head(10)

grouped.columns

**Output:**

****

Index(['NumberOfBoardings\_sum', 'NumberOfBoardings\_count','NumberOfBoardings\_max'],dtype='object')

**Input:**

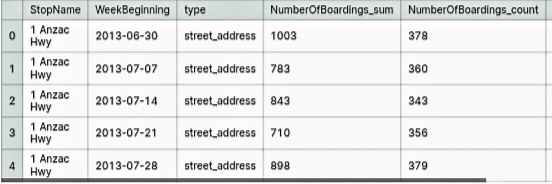
st\_week\_grp = pd.DataFrame(grouped).reset\_index()

st\_week\_grp.shape

st\_week\_grp.head()

**Output:**

(207864, 6)

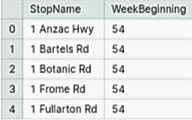


**Input:**

st\_week\_grp1 = pd.DataFrame(st\_week\_grp.groupby('StopName')["WeekBeginning"].count()).reset\_index()

st\_week\_grp1.head()

**Output:**



DATA EXPLORATION:

data.nunique()

Output:

TripID 39211

RouteID 616

StopID 5838

StopName 3127

WeekBeginning 54

NumberOfBoardings 359

latitude 2393

longitude 2379

postcode 138

type 8

dist\_from\_centre 2397

dtype: int64

**DATA VISUALIZATION:**

fig,axrr=plt.subplots(2,2,figsize=(15,15))

ax=axrr[0][0]

ax.set\_title("No of Boardings")

data['NumberOfBoardings'].value\_counts().sort\_index().head(20).plot.bar(ax=axrr[0][0])

ax=axrr[0][1]

ax.set\_title("WeekBeginning")

data['WeekBeginning'].value\_counts().plot.area(ax=axrr[0][1])

ax=axrr[1][0]

ax.set\_title("most Busiest Route")

data['RouteID'].value\_counts().head(10).plot.bar(ax=axrr[1][0])

ax=axrr[1][1]

ax.set\_title("least Busiest Route")

data['RouteID'].value\_counts().tail(10).plot.bar(ax=axrr[1][1])

**Output:**

Text(0.5,1,'No of Boardings')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff880af0940>

Text(0.5,1,'WeekBeginning')

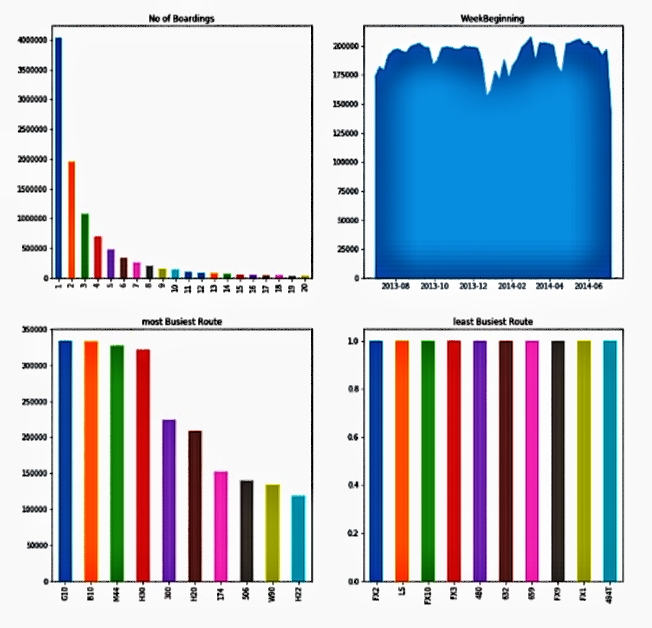
<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff709a6bb38>

Text(0.5,1,'most Busiest Route')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff709a48e10>

Text(0.5,1,'least Busiest Route')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff736bbafd0>



**IMPORTANCE OF LOADING AND PROCESSING DATASE:**

Loading and processing datasets are critical steps in any data analysis or machine learning project. The importance of these steps cannot be overstated for several reasons:

* **Data Quality Assurance:** Loading and preprocessing help ensure the quality and integrity of the data. By addressing missing values, outliers, and inconsistencies, you improve the reliability of the dataset. Low-quality data can lead to inaccurate analyses and misleading conclusions.
* **Data Consistency**: Datasets often come from various sources, and they may have different formats, units, or structures. Processing data allows you to standardize it, making it consistent and easier to work with. Consistency is crucial for valid and meaningful analyses.
* **Feature Engineering:** Preprocessing is essential for creating new features or variables that can enhance the analysis. These engineered features may hold valuable insights and are often used to improve model performance in machine learning projects.
* **Data Exploration:** Loading and preprocessing enable you to explore the dataset, gaining a better understanding of its characteristics. This exploration can help you identify patterns, relationships, and trends in the data, which are crucial for making informed decisions.
* **Data Size Reduction:** Large datasets can be computationally expensive to work with. Preprocessing can involve data reduction techniques, such as dimensionality reduction or downsampling, which can make the dataset more manageable without sacrificing important information.
* **Data Security and Privacy:** In many cases, sensitive or personal information may be present in the raw data. Preprocessing can include techniques for anonymization or de-identification to protect privacy while retaining the utility of the data.

**CHALLENGES INVOLVED IN LOADING AND PREPROCESSING A PUBLIC TRANSPORTATION DATASET**

Loading and preprocessing public transportation datasets can be a complex task with several challenges to overcome. Here are some of the key challenges you might encounter:

* **Data Volume:** Public transportation datasets can be vast, containing millions of records, which can strain memory and processing resources. Loading and handling such large datasets require careful resource management and optimized code.
* **Data Variety:** Public transportation data often comes from various sources, each with its own data format, schema, and quality. Integrating and harmonizing these diverse data sources can be a significant challenge.
* **Missing Data:** Public transportation datasets frequently have missing values, which can affect the quality of analysis. Deciding how to handle missing data (e.g., imputation, deletion, or interpolation) is a critical preprocessing step.
* **Data Inconsistency:** Inconsistent data formats, units, or naming conventions can make it difficult to combine data from different sources. Preprocessing may involve standardizing and cleaning the data for consistency.
* **Outliers and Anomalies:** Datasets may contain outliers or anomalies that can significantly impact analysis results. Detecting and handling these outliers appropriately is essential for accurate insights.
* **Real-Time Data Challenges:** In public transportation, real-time data streams can introduce issues related to data timing, synchronization, and data latency. These challenges must be addressed when dealing with live data.
* **Temporal Data:** Public transportation data often includes temporal information, such as schedules and timetables. Handling and processing this time-dependent data can be complex, especially when dealing with delays and disruptions.
* **Geospatial Data:** Many public transportation datasets involve geospatial data, including route information, stops, and GPS coordinates. Analyzing and processing geospatial data require specialized tools and knowledge.

**HOW TO OVERCOME THE CHALLENGES IN LOADING AND PREPROCESSING A PUBLIC TRANSPORTATION DATASET**

Overcoming the challenges in loading and preprocessing a public transportation dataset requires a combination of technical skills, domain knowledge, and careful planning. Here are some strategies to help you tackle these challenges effectively:

**Data Understanding and Domain Knowledge:**

Start by gaining a deep understanding of the public transportation domain. Familiarize yourself with relevant terminology, schedules, and operational practices.Collaborate with domain experts who can provide insights into the data and its context.

**Data Source Selection:**

Choose data sources carefully. Prioritize sources with well-documented and standardized data formats, as they can reduce data integration challenges.

**Data Cleaning and Quality Assurance:**

Implement data cleaning procedures to address missing values, data inconsistencies, and outliers. This includes techniques like imputation, data validation, and outlier detection.Utilize data profiling tools to automatically identify data quality issues.

**Data Integration:**

Develop a data integration strategy to merge data from various sources. Use data mapping, schema transformation, and data consolidation techniques to harmonize the data.

**Data Preprocessing Tools and Libraries:**

Leverage data preprocessing libraries and tools, such as Python's pandas, R's dplyr, or SQL, which offer a wide range of functions for data manipulation.

**Automated ETL (Extract, Transform, Load):**

Consider building an automated ETL pipeline to streamline data extraction, transformation, and loading. This can ensure consistency and repeatability in data processing.

**Real-Time Data Handling:**

Implement real-time data processing pipelines if working with live data streams. Technologies like Apache Kafka and Apache Spark Streaming can be beneficial.

**PROGRAM:**

import pandas as pd

import matplotlib.pyplot as plt

dataset=pd.read\_csv('https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV')

print("First few rows of the dataset:")

print(dataset.head())

print("\nMissing values:")

print(dataset.isnull().sum())

dataset = dataset.dropna()

dataset['date'] = pd.to\_datetime(dataset['date'])

plt.figure(figsize=(12, 6))

plt.plot(dataset['date'], dataset['riders'])

plt.title("Ridership Over Time")

plt.xlabel("Date")

plt.ylabel("Number of Riders")

plt.grid(True)

plt.show()

dataset.to\_csv(': https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV', index=False)



This code demonstrates the loading and preprocessing of a public transportation dataset, specifically handling missing values and converting a date column to a datetime object. Additionally, it includes a simple line plot to visualize ridership data over time.

**PREDICTING THE DATA SET:**

import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv('your\_dataset.csv')

print("First few rows of the dataset:")

print(dataset.head())

print("\nMissing values:")

print(dataset.isnull().sum())

plt.figure(figsize=(10, 6))

plt.hist(dataset['your\_column'], bins=20)

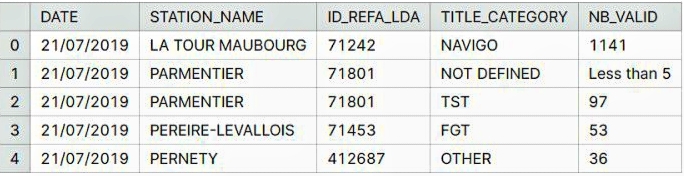
plt.title("Histogram of Your Column")

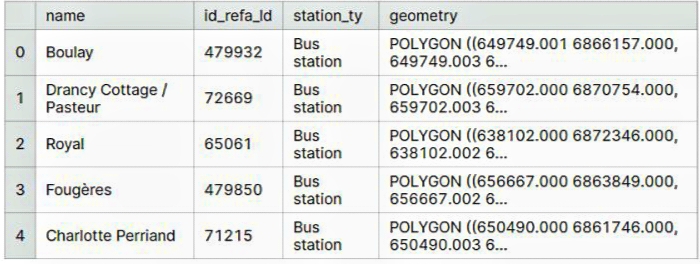
plt.xlabel("Value")

plt.ylabel("Frequency")

plt.grid(True)

plt.show()



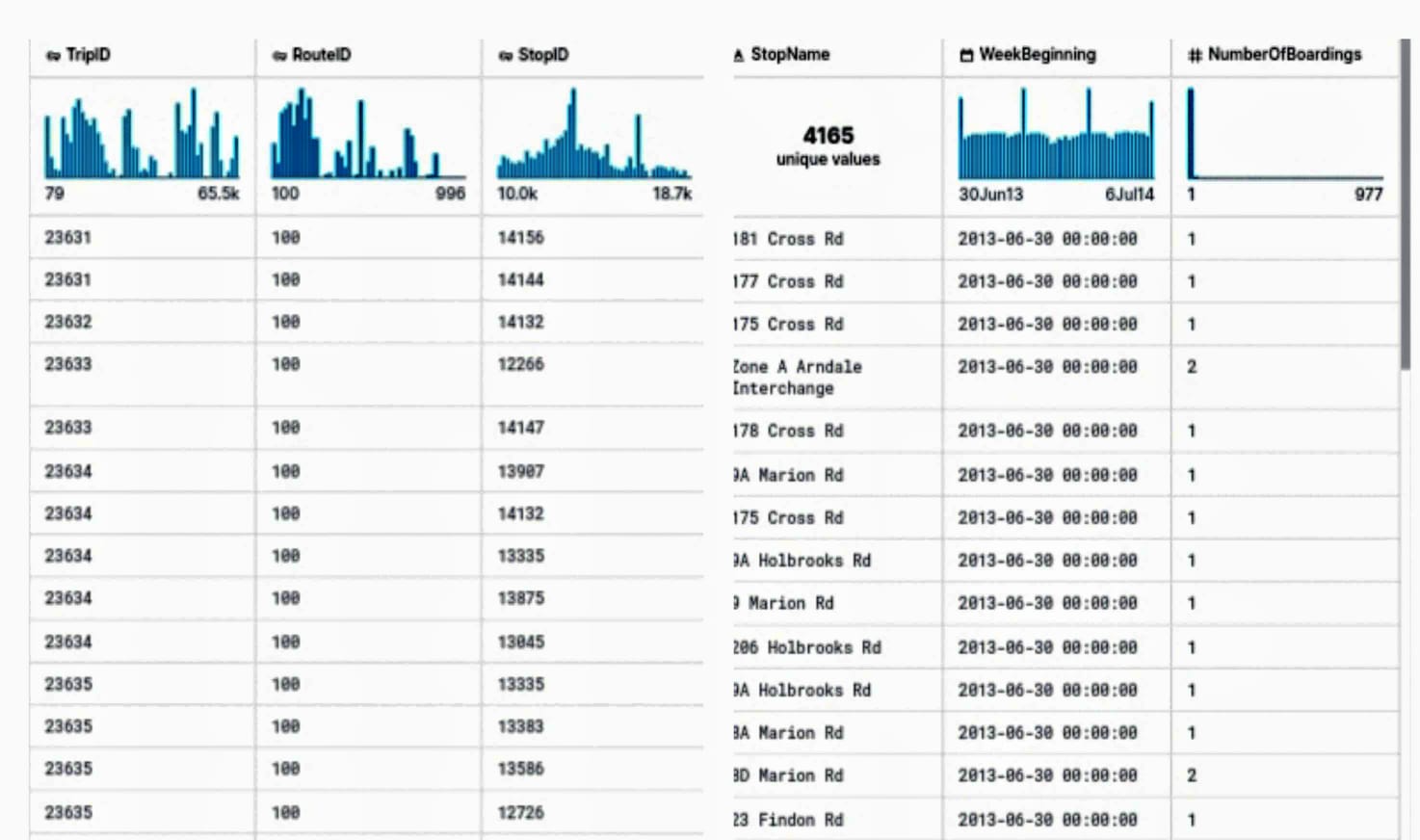


**1.LOADING DATASET :**

Dataset = pd.ready\_csv(“E:/USA\_Public transportation.csv”)

**DATA EXPLORATION:**

**Data Set:**

****

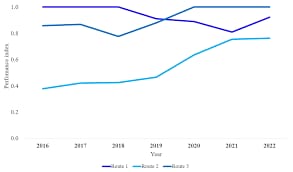
**2.PREPROCESSING THE DATA SET:**

**1.Load the Data:**

Use a library like Pandas in Python to load the CSV file into a DataFrame.

import pandas as pd

data = pd.read\_csv('20140711.CSV')

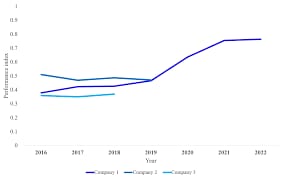


**2.Data Exploration:**

Explore the data to get a better understanding of its structure. Check for the number of rows, columns, data types, and some sample data points.

data.head()

data.info()



**3.Handling Missing Values:**

Check for missing values and decide how to handle them. You can either impute missing data or remove rows/columns with missing values, depending on the dataset and your analysis.

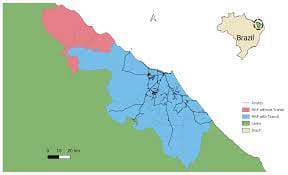
data.isnull().sum()

# To impute missing values:

data.fillna(value, inplace=True)

# To remove rows with missing values:

data.dropna(inplace=True)



**4.Data Cleaning:**

Correct any inconsistencies or errors in the data. This may include dealing with outliers, data type conversions, or renaming columns for clarity.

**5.Feature Engineering:**

Create new features or transform existing ones to extract more meaningful information. For example, you can extract date-related information, one-hot encode categorical variables, or scale numerical features.

**6.Data Encoding:**

Encode categorical variables into numerical format using techniques like one-hot encoding or label encoding.

data = pd.get\_dummies(data, columns=['categorical\_column'])

**7.Split Data:**

If you're planning to train a machine learning model, split the data into training and testing sets.

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

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

**8.Scaling/Normalization:**

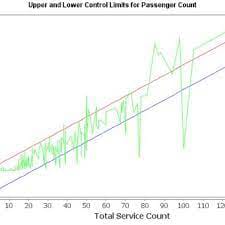
If you're using algorithms sensitive to feature scales (e.g., SVM, K-Means), consider scaling or normalizing your data.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)



**9.Data Saving:**

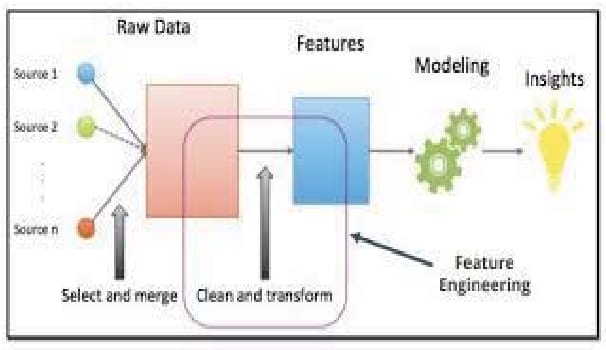
Save the preprocessed data for future use, so you don't have to repeat these steps.

data.to\_csv('preprocessed\_data.csv', index=False)

These are general preprocessing steps, and the specific steps you need to perform may vary depending on your dataset and analysis goals. You should also consider domain-specific knowledge and any particular requirements of your analysis or modeling task.

**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

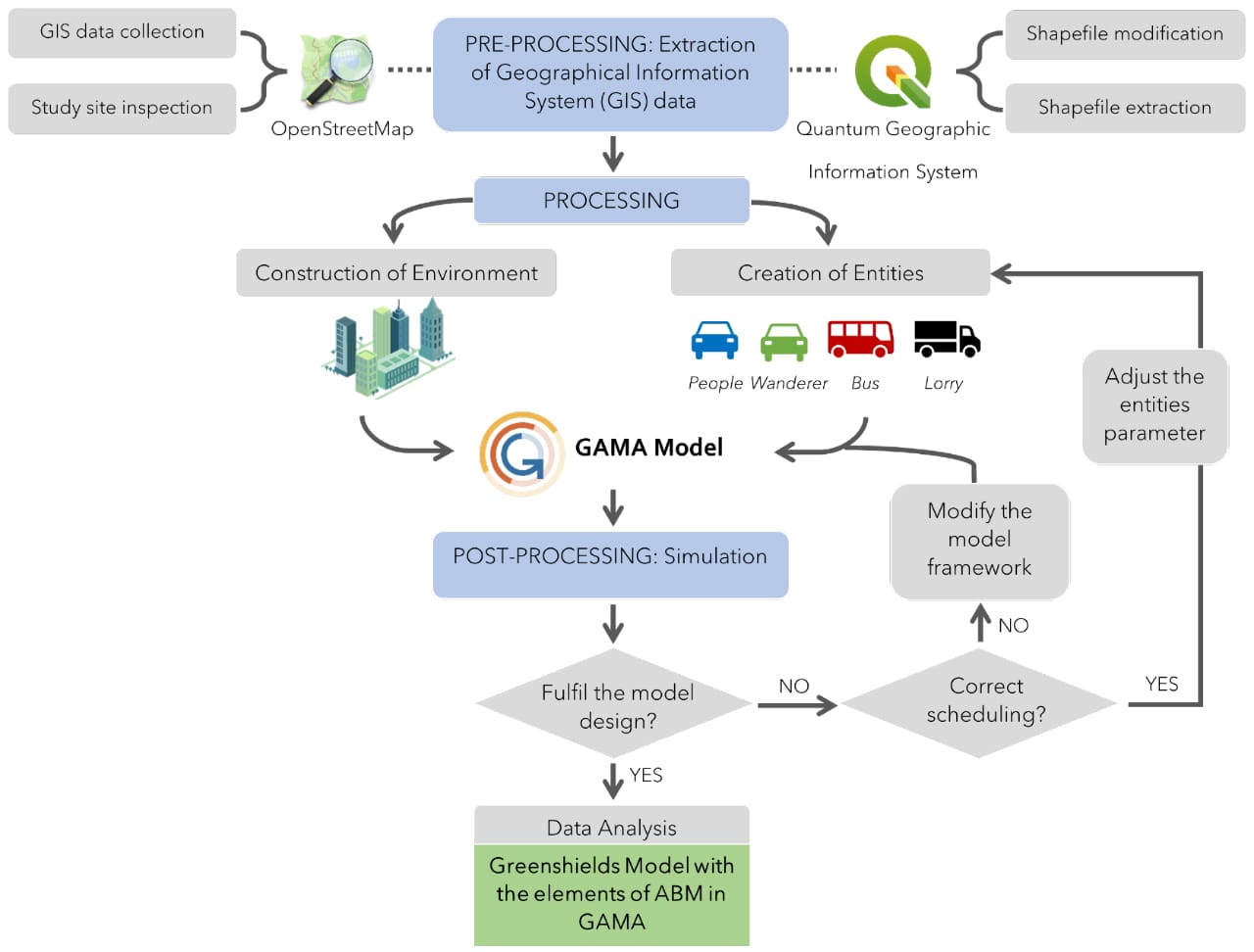
print(f'Mean Absolute Error: {mae}')

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 efficience
* 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))

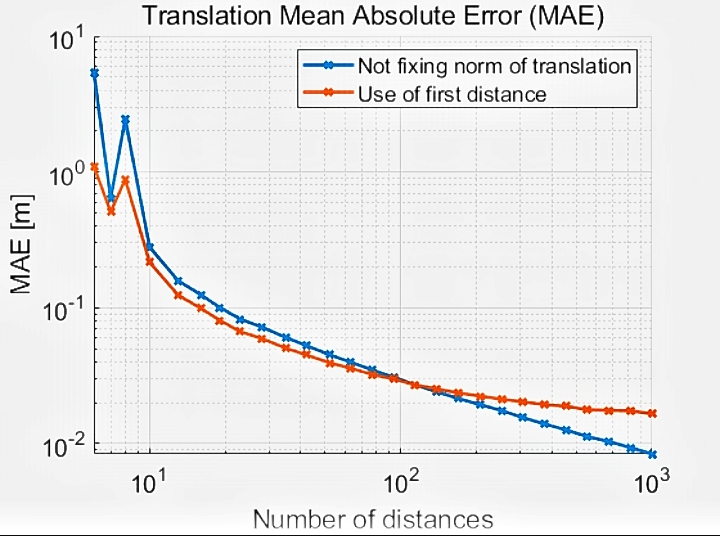
plt.scatter(y\_test, y\_pred, alpha=0.5)

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

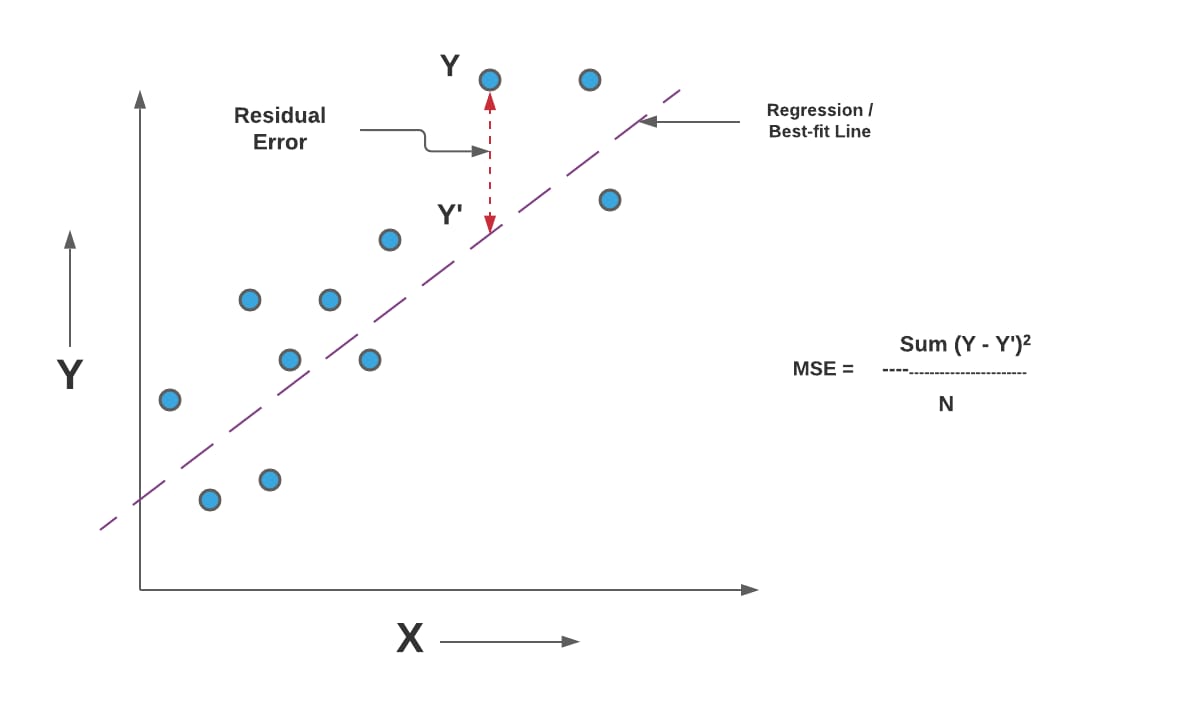
actual = [10, 20, 30, 40, 50]

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]

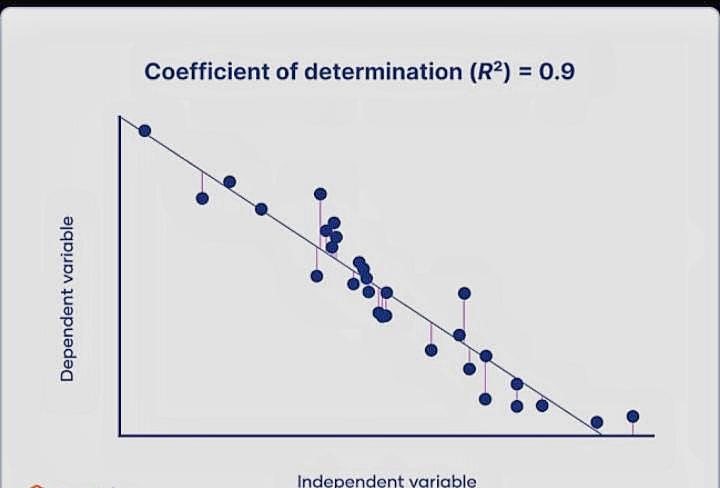
y\_predicted = [12, 18, 35, 42, 48]

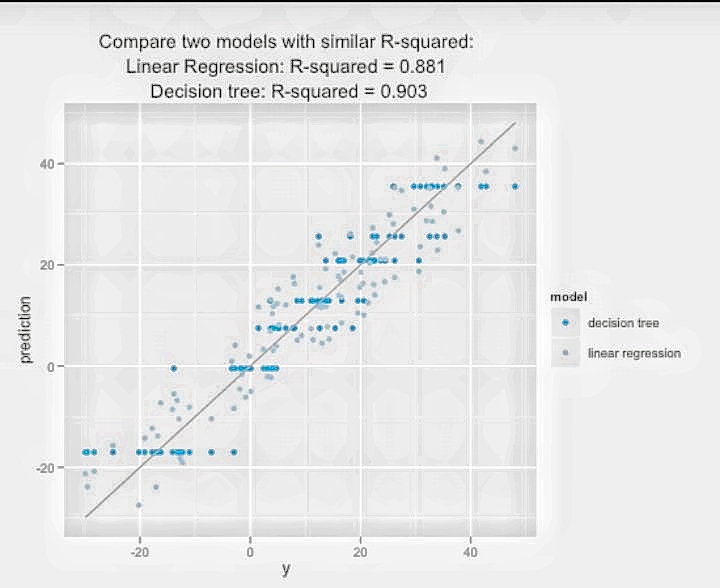
# Calculate R-squared

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

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

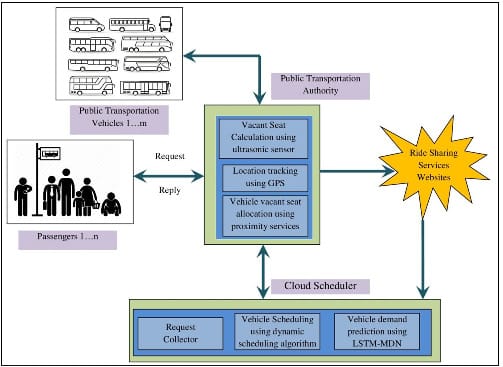
This code will calculate and print the R-squared value for your public transportation efficiency analysis. R² values range between 0 and 1, with higher values indicating a better fit of model to the data.





**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)

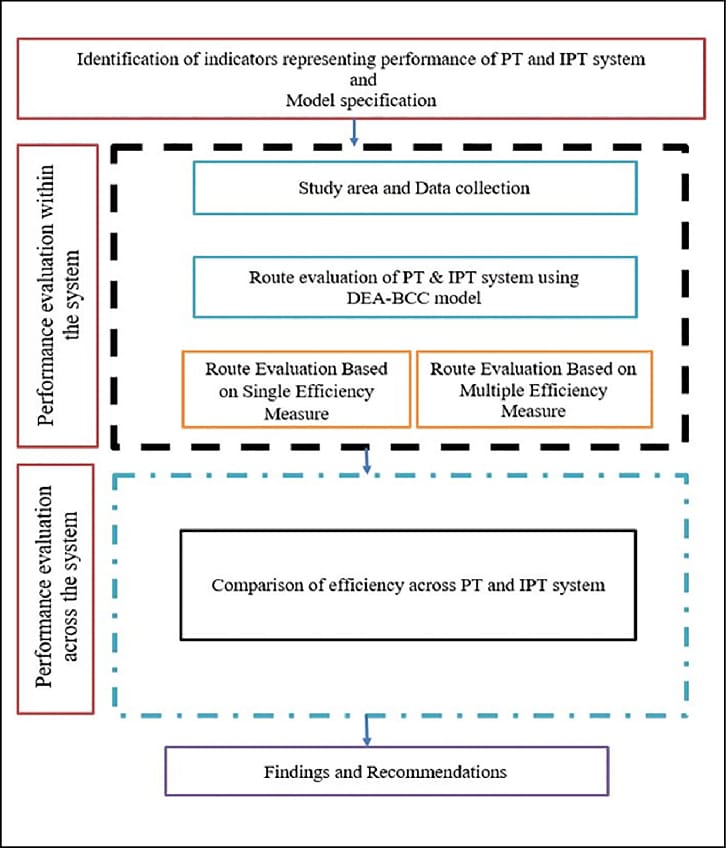
print("R-squared (R²):", r\_squared)

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, our comprehensive analysis of public transport efficiency has shed light on the critical role that well-organized and sustainable transportation systems play in our communities. Throughout this project, we have delved into various aspects of public transport, examining factors such as accessibility, affordability, environmental impact, and overall effectiveness. Our findings have highlighted the importance of investing in public transport infrastructure and optimizing its operations to create a more efficient and eco-friendly mode of transportation.
* Through our research, we have not only identified areas for improvement but also celebrated the successes and innovations that many cities and regions have achieved in enhancing their public transport systems. We have learned that a well-planned, integrated, and accessible public transport network can significantly reduce traffic congestion, lower emissions, and improve the overall quality of life for residents.
* In the face of growing urbanization and environmental concerns, our analysis underscores the necessity of continuing to invest in and innovate within the public transport sector. It is our hope that the data and insights presented in this project will serve as a valuable resource for policymakers, urban planners, and transportation authorities in their efforts to create more efficient and sustainable public transport systems.
* As we move forward, we encourage ongoing collaboration and research in this field, with the ultimate goal of making public transport the preferred choice for commuters worldwide. By prioritizing public transport efficiency, we can not only alleviate congestion and reduce our carbon footprint but also enhance the overall well-being of our communities.