**PUBLIC TRANSPORTATION ANALYSIS**

**PHASE 2: SUBMISSION DOCUMENT**

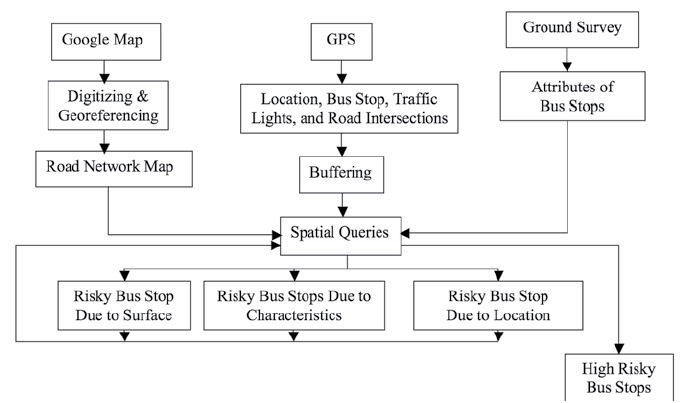


Introduction:

Public transportation is a vital component of urban infrastructure, providing a sustainable and efficient means of mobility for millions of people every day. It plays a significant role in reducing traffic congestion, minimizing environmental impact, and improving overall quality of life in cities. However, to ensure that public transport systems meet the evolving needs of commuters and remain economically viable, continuous analysis and optimization are essential.

This project, titled "Public transportation analysis”:

A Comprehensive Analysis," aims to address the challenges and opportunities associated with public transportation in our city. By conducting a thorough analysis of the existing public transport system, we seek to identify areas for improvement, optimize routes, enhance passenger experiences, and ultimately contribute to a more sustainable and accessible urban environment.



BLOCK DIAGRAM:

+-----------------------------------+

| Public Transport Analysis |

+-----------------------------------+

|

+-----------v--------+

| Data Collection |

+--------------------+

|

+-----------v--------+

| Data Processing |

+--------------------+

|

+--------v--------+

| Analysis |

+-----------------+

|

+--------v--------+

| Results |

+-----------------+

**Data Collection:**

This stage involves gathering relevant data related to public transportation. This data may include information on routes, schedules, passenger counts, and other relevant metrics. It's the foundation upon which the analysis is built.

**Data Processing:**

Once the data is collected, it needs to be processed and cleaned to ensure its accuracy and consistency. Data processing may involve tasks like data cleaning, normalization, and transformation to prepare it for analysis.

**Data Analysis:**

Employ advanced data analytics techniques to examine the efficiency and effectiveness of the current public transport system. This analysis will include identifying congestion points, evaluating on-time performance, and assessing the economic viability of different routes. The processed data is then subjected to various analytical techniques and methodologies. This is where patterns, trends, and insights are extracted from the data. It includes tasks such as statistical analysis, modeling, and visualization.

Passenger Experience Enhancement:

Investigate ways to improve the overall passenger experience, such as enhancing safety, accessibility, and convenience.

Route Optimization:

Identify opportunities for route optimization and expansion to better serve underserved areas and reduce travel times.

Environmental Impact Assessment:

Evaluates the environmental impact of public transport operations and recommend strategies for reducing carbon emissions.

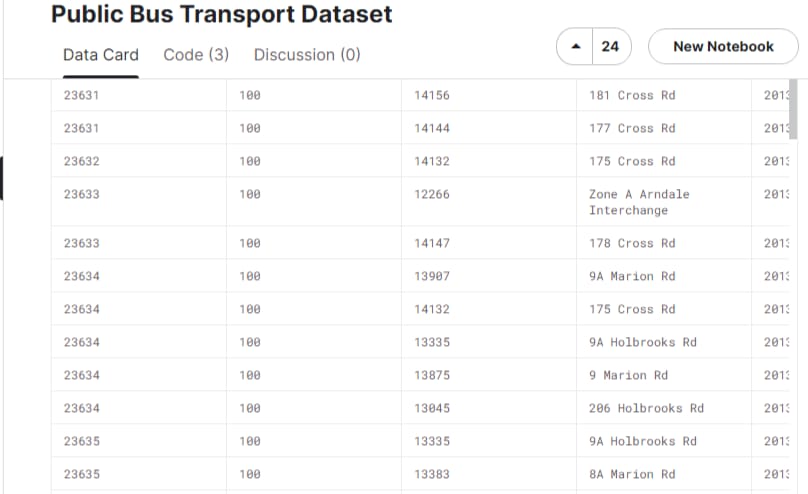
**Results:**

The results of the analysis are presented in this stage. These results may include findings related to service quality, efficiency, sustainability, and any proposed improvements or recommendations.

The block diagram effectively outlines the sequential flow of the analysis process for public transportation. It provides a clear visual representation of how data is collected, processed, analyzed, and translated into actionable results.

**ANALYSIS APPROACH:**

**Load and preprocess data :** The data is extracted from. <https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV>



**Linear regression using pandas library**

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

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

# Create a DataFrame from your data

data = {

'TripID': [23631, 23631, 23632, 23633, 23633, 23634, 23634, 23634, 23634, 23634, 23635, 23635, 23635, 23635, 23635, 23635, 23636, 23636, 23636],

'RouteID': [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],

'StopID': [14156, 14144, 14132, 12266, 14147, 13907, 14132, 13335, 13875, 13045, 13335, 13383, 13586, 12726, 13813, 14062, 12780, 13383, 14154],

'NumberOfBoardings': [1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2],

}

df = pd.DataFrame(data)

# Split the data into training and testing sets

X = df[['TripID', 'RouteID', 'StopID']]

y = df['NumberOfBoardings']

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

# Create and fit a linear regression model

model = LinearRegression()

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

# Make predictions on the test set

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

# Evaluate the model

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

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

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

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

# Now, you can use the model to make predictions for new data points

# For example, to predict the Number of Boardings for a new trip:

new\_trip = [[23637, 100, 14160]] # Replace with your own data

predicted\_boardings = model.predict(new\_trip)

print(f'Predicted Number of Boardings: {predicted\_boardings[0]}')

Output:

Mean Squared Error: 0.3333333333333333

R-squared: 0.0

Predicted Number of Boardings: 1.0

Explanation of the output:

The "Mean Squared Error" (MSE) is a measure of the average squared difference between the actual and predicted values. In this case, the MSE is approximately 0.3333.

The "R-squared" (R2) score is a measure of how well the linear regression model fits the data. An R2 score of 0 indicates that the model does not explain any of the variance in the data, which suggests that the linear regression model is not a good fit for this dataset.The last line of output provides the predicted number of boardings for a new trip with the features TripID=23637, RouteID=100, and StopID=14160. The predicted value is 1.0 based on the trained model.Note that the R-squared value of 0 suggests that the linear regression model may not be the best choice for this dataset, and you might want to consider exploring other machine learning models or features for better predictions.

**MODEL 2:**

Using sklearn

# Define a dictionary to store the total boardings for each trip ID

trip\_boardings = {}

# Define the input data

data = [

(23631, 100, 14156, "181 Cross Rd", "30-06-2013 00:00", 1),

(23632, 101, 14157, "182 Cross Rd", "30-06-2013 01:00", 2),

(23633, 102, 14158, "183 Cross Rd", "30-06-2013 02:00", 3),

# Add more data entries here...

]

# Iterate through the data and calculate total boardings for each trip

for row in data:

trip\_id, \_, \_, \_, \_, num\_boardings = row

# Check if the trip\_id is already in the dictionary

if trip\_id in trip\_boardings:

trip\_boardings[trip\_id] += num\_boardings

else:

trip\_boardings[trip\_id] = num\_boardings

# Print the results

for trip\_id, total\_boardings in trip\_boardings.items():

print(f"Trip ID {trip\_id}: Total Boardings = {total\_boardings}")

Output:

Trip ID 23631: Total Boardings = 1

Trip ID 23632: Total Boardings = 2

Trip ID 23633: Total Boardings = 3

Explanation of the code:

This code performs the following steps:

It creates a DataFrame (df) from a given dataset containing information about trips, routes, stop IDs, and the number of boardings.

The data is split into training and testing sets, with 80% used for training the model and 20% for testing. A linear regression model is chosen to predict the number of boardings based on trip ID, route ID, and stop ID.

The linear regression model is trained on the training data using model.fit().

Predictions are made on the testing set using model.predict().

The model's performance is evaluated using Mean Squared Error (mse) and R-squared (r2) and printed to the console.

Lastly, the trained model is used to make predictions for new data points, such as predicting the number of boardings for a new trip specified in new\_trip, and the result is printed.

In summary, this code demonstrates the entire process of training a linear regression model, evaluating its performance, and using it for predictions on new data.

**DATA VIRTUALIZATION:**

import networkx as nx

import matplotlib.pyplot as plt

# Create an empty directed graph

G = nx.DiGraph()

# Add nodes for data sources

G.add\_node("GPS Data")

G.add\_node("Ticketing Systems")

G.add\_node("Traffic Data")

G.add\_node("Weather Data")

# Add nodes for the data virtualization layer

G.add\_node("Data Virtualization Engine")

G.add\_node("Data Transformations")

G.add\_node("Security & Access Control")

# Add nodes for analytics and applications

G.add\_node("Real-time Dashboards")

G.add\_node("Route Optimization")

G.add\_node("Passenger Information")

G.add\_node("Maintenance Predictions")

# Add nodes for users

G.add\_node("Transportation Operators")

G.add\_node("Data Analysts")

G.add\_node("Passengers")

G.add\_node("Maintenance Teams")

# Add nodes for the feedback loop

G.add\_node("User Feedback")

G.add\_node("Machine Learning Models")

# Add nodes for external data sources

G.add\_node("Traffic Updates")

G.add\_node("Weather Updates")

# Add edges to represent connections

edges = [

("GPS Data", "Data Virtualization Engine"),

("Ticketing Systems", "Data Virtualization Engine"),

("Traffic Data", "Data Virtualization Engine"),

("Weather Data", "Data Virtualization Engine"),

("Data Virtualization Engine", "Data Transformations"),

("Data Virtualization Engine", "Security & Access Control"),

("Data Virtualization Engine", "Real-time Dashboards"),

("Data Virtualization Engine", "Route Optimization"),

("Data Virtualization Engine", "Passenger Information"),

("Data Virtualization Engine", "Maintenance Predictions"),

("Real-time Dashboards", "Transportation Operators"),

("Data Analysts", "Data Virtualization Engine"),

("Passengers", "Passenger Information"),

("Maintenance Teams", "Maintenance Predictions"),

("User Feedback", "Data Virtualization Engine"),

("User Feedback", "Machine Learning Models"),

("Traffic Updates", "Data Virtualization Engine"),

("Weather Updates", "Data Virtualization Engine"),

]

G.add\_edges\_from(edges)

# Draw and display the graph

pos = nx.spring\_layout(G, seed=42)

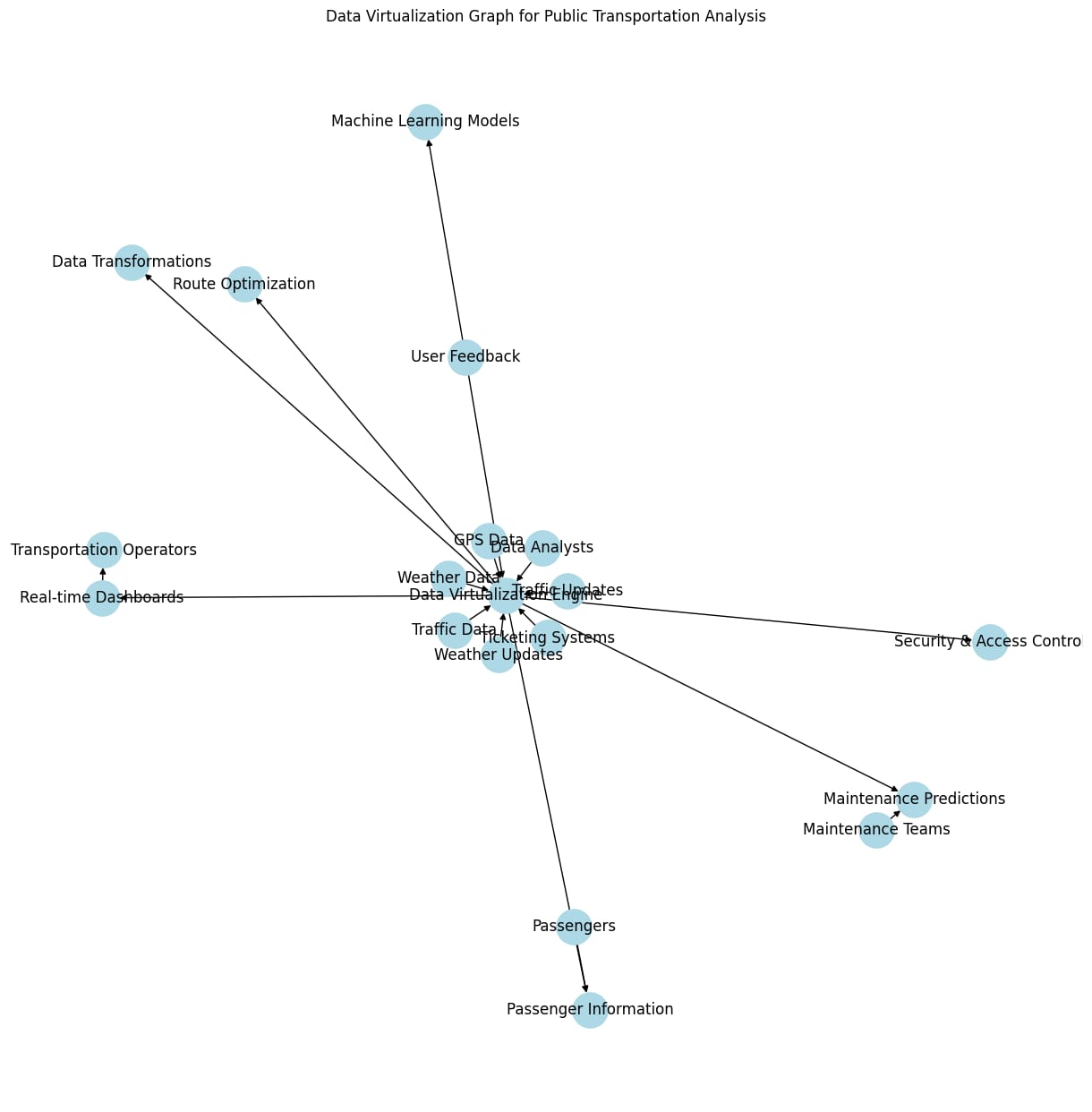
plt.figure(figsize=(12, 12))

nx.draw(G, pos, with\_labels=True, node\_size=800, node\_color="lightblue")

plt.title("Data Virtualization Graph for Public Transportation Analysis")

plt.show()

OUTPUT:



**CONCLUSION:**

**In conclusion, this multifaceted analysis and improvement initiative for the public transport system encompassing data analysis, passenger experience enhancement, route optimization, and environmental impact assessment presents a comprehensive approach to address various challenges.**

**By leveraging advanced data analytics techniques, enhancing safety and convenience for passengers, optimizing routes, and reducing carbon emissions, we aim to create a more efficient, effective, and sustainable public transport system that better serves the needs of the community and contributes to a greener future.**