**PUBLIC TRANSPORTATION ANALYSIS**



**Comprehensive Study:**

**Ma**

objectives to integrating code for effective decision-making.

Explore the world of public transportation analysis, from setting analysis

**AnalysisObjective:**

**1.optimizing efficiency:**

**Analyzing data points to identify areas for improvement and increase the efficiency of public transportation systems**

**2.enchancing accessibility:**

**Examining the accessibility of public transportation networks to ensure equal access for all members of the community**

**3Reducing emissions:**

**Utilizing data analysis to develop sustainable strategies that reduce**

**environmental impact and promote cleaner transportation**.

**Data Collection**

**Survey andInterviews:**

Engaging with commuters and transportation officials to gather valuable insights and feedback on the current system.

**Data Mining Techniques**

Exploring large datasets and leveraging advanced algorithms to extract meaningful patterns and trends.

**Sensor and IoT Data**

Capturing real-time data from smart sensors and Internet ofThings (IoT) devices to monitor

**Code Integration**

**Automation and Machine Learning**

Integrating code and algorithms to automate data analysis, predictive modeling, and optimization of public transportation systems.

**Data Visualization Libraries**

Leveraging powerful data visualization libraries to create informative and visually appealing representations of public transportation data.

Analysis of Public Transportation for Efficiency

**Steps to preprocessing the dataset**

**import pandas as pd**

**import scipy**

**import numpy as np**

**from sklearn.preprocessing import MinMaxScaler**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**df=pd.read\_csv('20140171.CSV')**

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 10857234 entries, 0 to 10857233

Data columns (total 6 columns):

# Column Dtype

--- ------ -----

0 TripID int64

1 RouteID object

2 StopID int64

3 StopName object

4 WeekBeginning object

5 NumberOfBoardings int64

dtypes: int64(3), object(3)

memory usage: 497.0+ MB

In [83]:

df.head(5)

Out[83]:

|  | **TripID** | **RouteID** | **StopID** | **StopName** | **WeekBeginning** | **NumberOfBoardings** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 23631 | 100 | 14156 | 181 Cross Rd | 2013-06-30 00:00:00 | 1 |
| **1** | 23631 | 100 | 14144 | 177 Cross Rd | 2013-06-30 00:00:00 | 1 |
| **2** | 23632 | 100 | 14132 | 175 Cross Rd | 2013-06-30 00:00:00 | 1 |
| **3** | 23633 | 100 | 12266 | Zone A Arndale Interchange | 2013-06-30 00:00:00 | 2 |
| **4** | 23633 | 100 | 14147 | 178 Cross Rd | 2013-06-30 00:00:00 | 1 |

In [84]:

df.tail(5)

Out[84]:

|  | **TripID** | **RouteID** | **StopID** | **StopName** | **WeekBeginning** | **NumberOfBoardings** |
| --- | --- | --- | --- | --- | --- | --- |
| **10857229** | 13346 | W91C | 14629 | 21 Cashel St | 2014-07-06 00:00:00 | 1 |
| **10857230** | 13346 | W91C | 14708 | 22 Cashel St | 2014-07-06 00:00:00 | 3 |
| **10857231** | 13346 | W91C | 13709 | 2 Greenhill Rd | 2014-07-06 00:00:00 | 1 |
| **10857232** | 13346 | W91C | 14029 | 10 East Av | 2014-07-06 00:00:00 | 1 |
| **10857233** | 13346 | W91C | 13824 | 6 Leader St | 2014-07-06 00:00:00 | 1 |

In [58]:

df.isnull

<bound method NDFrame.\_add\_numeric\_operations.<locals>.sum of TripID RouteID StopID StopName WeekBeginning NumberOfBoardings

0 False False False False False False

1 False False False False False False

2 False False False False False False

3 False False False False False False

4 False False False False False False

... ... ... ... ... ... ...

10857229 False False False False False False

10857230 False False False False False False

10857231 False False False False False False

10857232 False False False False False False

10857233 False False False False False False

[10857234 rows x 6 columns]>

df.describe()

|  | **TripID** | **StopID** | **NumberOfBoardings** |
| --- | --- | --- | --- |
| **count** | 1.085723e+07 | 1.085723e+07 | 1.085723e+07 |
| **mean** | 2.952100e+04 | 1.366132e+04 | 4.743737e+00 |
| **std** | 1.960938e+04 | 1.971760e+03 | 9.382286e+00 |
| **min** | 7.900000e+01 | 1.000100e+04 | 1.000000e+00 |
| **25%** | 1.191700e+04 | 1.231100e+04 | 1.000000e+00 |
| **50%** | 2.747900e+04 | 1.334600e+04 | 2.000000e+00 |
| **75%** | 4.885800e+04 | 1.491600e+04 | 4.000000e+00 |
| **max** | 6.553500e+04 | 1.871500e+04 | 9.770000e+02 |
| **corr = df.corr()**    **plt.figure(dpi=130)**  **sns.heatmap(df.corr(), annot=True, fmt= '.2f')**  **plt.show()** |  |  |  |

In [69]:

df.isnull().sum

<bound method NDFrame.\_add\_numeric\_operations.<locals>.sum of TripID RouteID StopID StopName WeekBeginning NumberOfBoardings

0 False False False False False False

1 False False False False False False

2 False False False False False False

3 False False False False False False

4 False False False False False False

... ... ... ... ... ... ...

10857229 False False False False False False

10857230 False False False False False False

10857231 False False False False False False

10857232 False False False False False False

10857233 False False False False False False

[10857234 rows x 6 columns]>

corr['StopID'].sort\_values(ascending **=** **False**)

Out[69]:

StopID 1.000000

TripID 0.105974

NumberOfBoardings 0.038397

Name: StopID, dtype: float64

In [77]:

X **=** df.drop(columns **=**['StopID'])

Y **=** df.StopID

In [68]:

**def** mean\_imputation(data, inplace **=** **False**):

data.fillna(data.mean(), inplace **=** inplace)

​

In [ ]:

scaler **=** StandardScaler()

scaler.fit(X\_train)

X\_train\_standardized **=** scaler.transform(X\_train)

X\_cv\_standardized **=** scaler.transform(X\_cv)

In [49]:

**import** imblearn

**from** imblearn.over\_sampling **import** RandomOverSampler

**from** imblearn.under\_sampling **import** TomekLinks

**from** imblearn.over\_sampling **import** SMOTE

**from** imblearn.under\_sampling **import** NearMiss

**def** sampler\_function(data\_x, data\_y, sampler **=** 0, random\_state **=** 101):

**if** sampler **==** 0:

sampler **=** RandomOverSampler(random\_state **=** random\_state)

**elif** sampler **==** 1:

sampler **=** TomekLinks()

**elif** sampler **==** 2:

sampler **=** SMOTE()

**else**:

sampler **=** NearMiss()

X\_transformed, y\_transformed **=** sampler.fit\_resample(data\_x, data\_y)

print('Original dataset shape:', Counter(data\_y))

print('Resample dataset shape:', Counter(y\_transformed))

**return** X\_transformed, y\_transformed

In [87]:

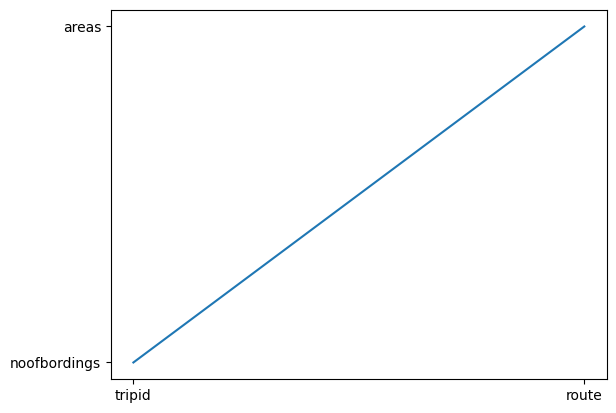
**from** matplotlib **import** pyplot **as** plt

x **=** ['tripid', 'route']

y **=** ['noofbordings', 'areas']

plt.plot(x, y)

plt.show()



In [89]:

from matplotlib import pyplot as plt

x = ['TripID','routeID','StopID']

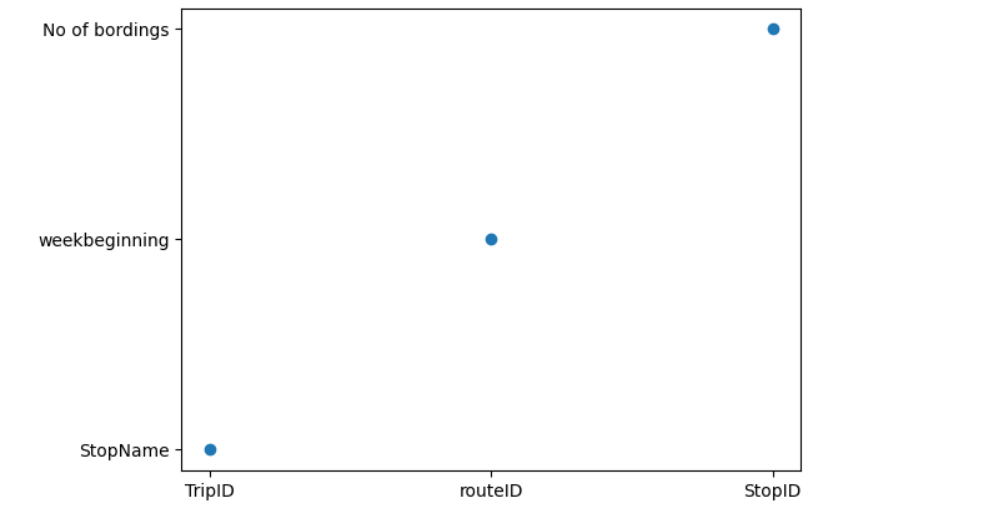
y = ['StopName','weekbeginning','No of bordings']

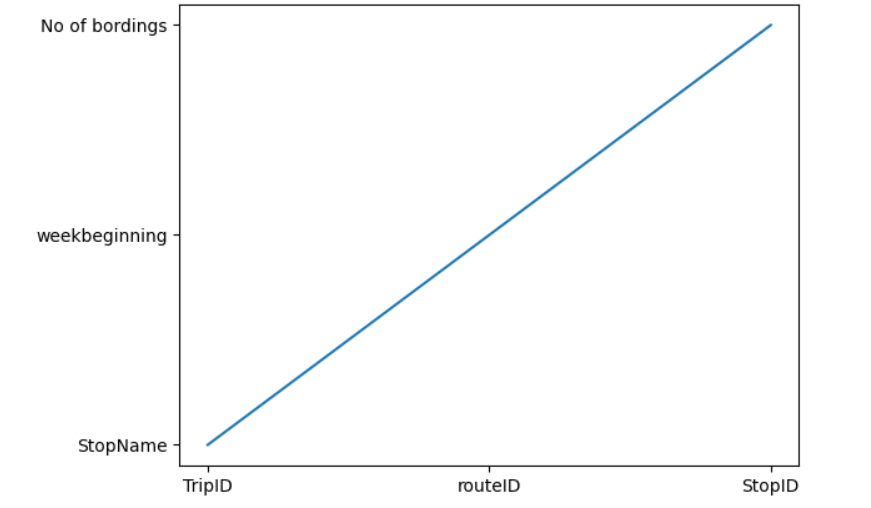
plt.scatter(x, y)

plt.show()

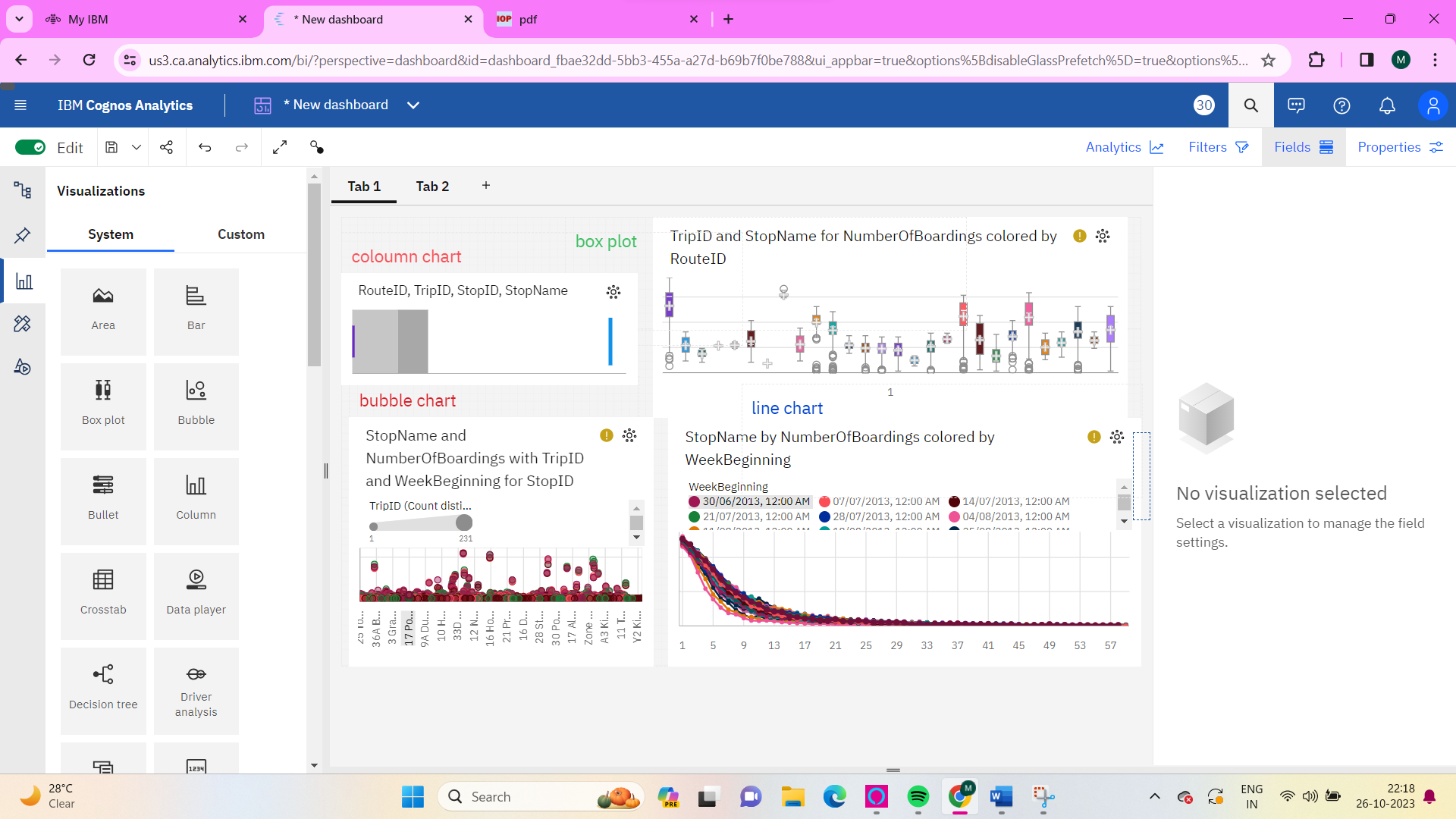
plt.plot(x, y)

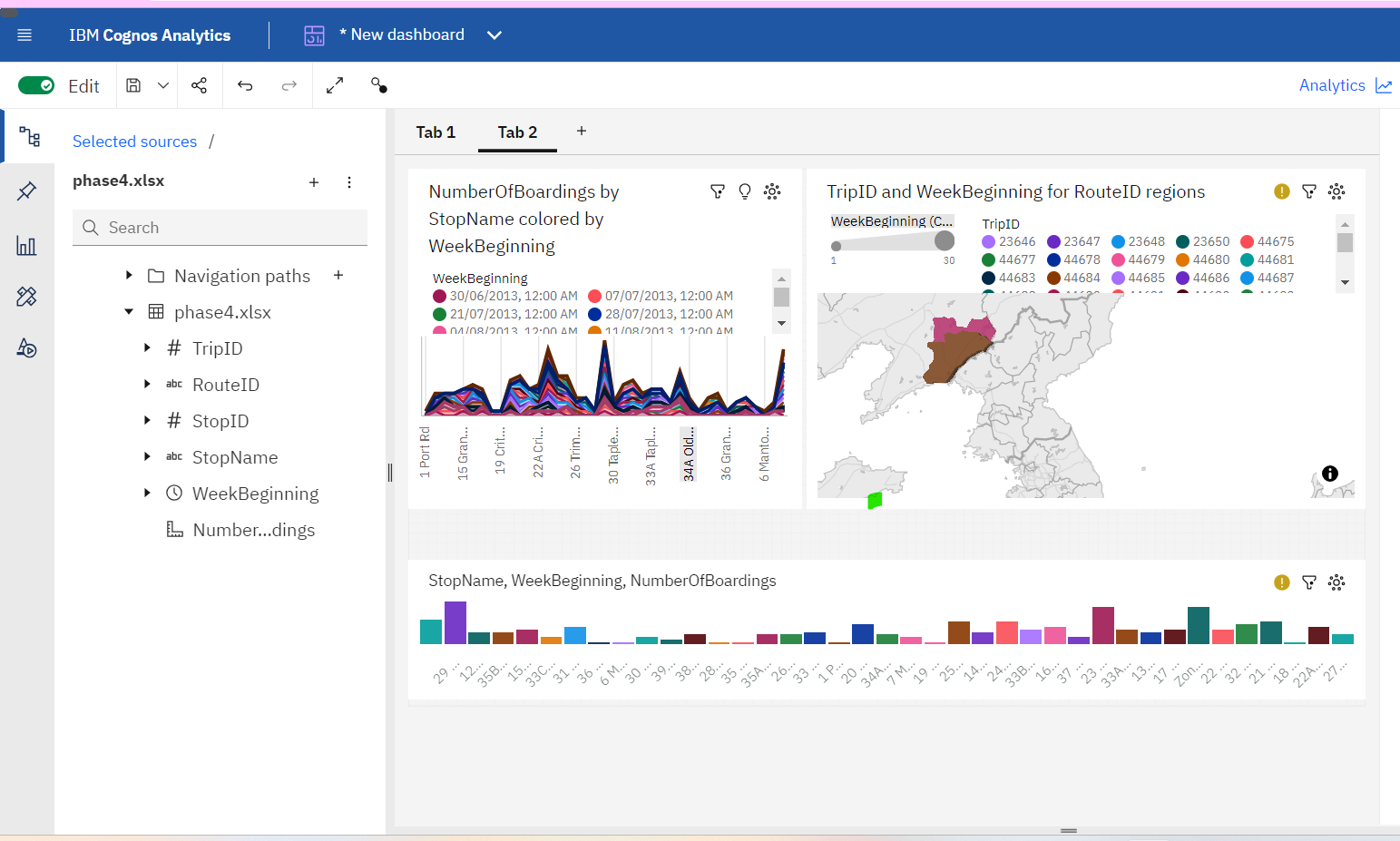
plt.show()

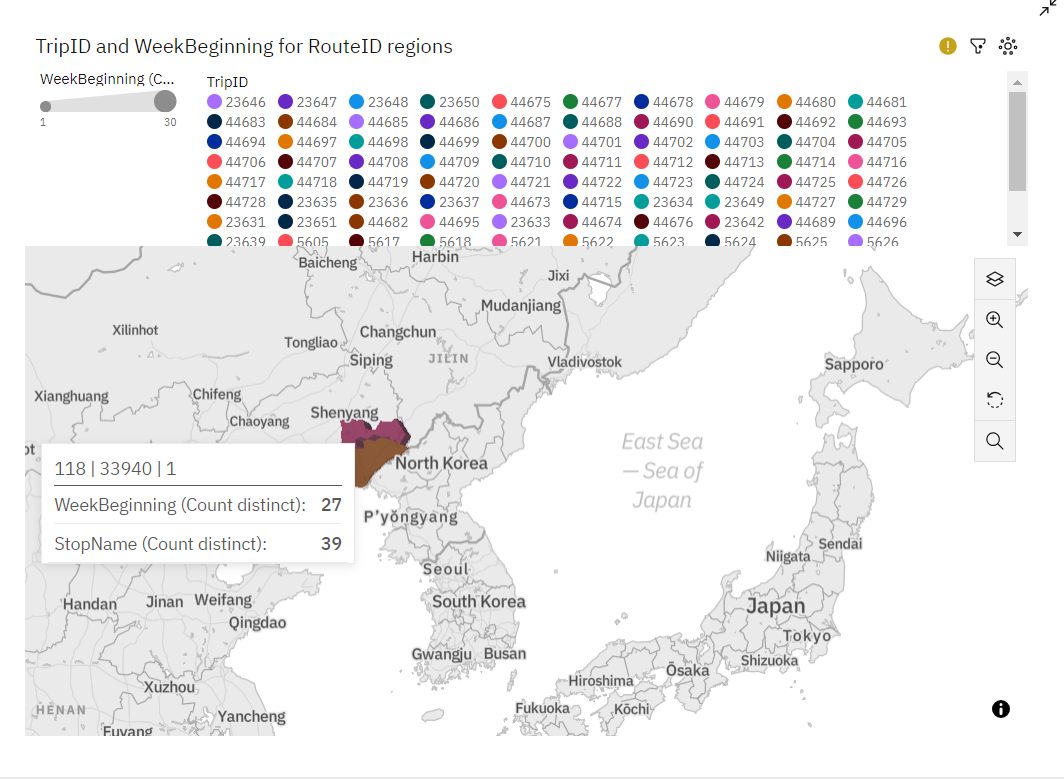


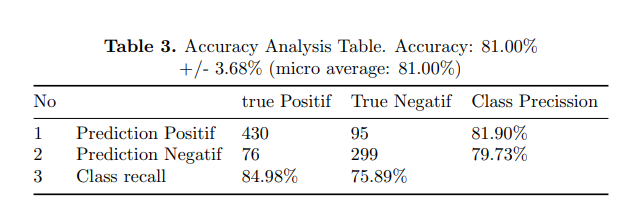


Sentiment analysis on passengers feed back & creating Dashboard









**Conclusion**

Transportation analysis plays a crucial role in optimizing efficiency,

accessibility, and sustainability of public transportation systems.

Data collection techniques and visualization strategies provide valuable

insights for decision-making.

Integrating code and smart city initiatives can lead to transformativeimprovements in urban transportation.