Project 2: Public Transport Efficiency Analysis

Objective:

Our goal is to incorporate Machine Learning algorithms to improve the accuracy of the predictive model by continuously monitoring and training the ML Model for Public Transport Efficiency Analysis

Project Definition:

The project involves analyzing public transportation data to assess service efficiency, on time performance, and passenger feedback. The objective is to provide insights that support transportation improvement initiatives and enhance the overall public transportation experience. This project includes defining analysis objectives, collecting transportation data, designing relevant visualizations in IBM Cognos, and using code for data analysis.

Data loading:

Data loading for public transport analysis is a crucial initial step in the data analysis pipeline. This process involves the collection, cleaning, integration, transformation, and loading of data for further analysis.

Data collection begins by gathering information from diverse sources, including transit agencies, GPS trackers, ticketing systems, and open data portals. This data encompasses details about routes, stops, schedules, ticket sales, and vehicle tracking.

The subsequent step involves data cleaning, where datasets are scrutinized for duplicates, missing values, and outliers. This cleaning process ensures that the data is consistent and accurate, providing a reliable foundation for analysis.

Data integration may be necessary to merge data from various sources to create a comprehensive dataset. This step is essential for understanding the complete public transport network.

Data transformation is performed to prepare the data for analysis. This often includes aggregating data by time intervals (e.g., hourly or daily), geospatial analysis, and feature engineering to make it suitable for analytical tools and techniques.

Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for vaccine analysis, as the datasets are often complex and noisy. By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

Missing Data

Another common issue that we face in real-world data is the absence of data points. Most machine learning models can’t handle missing values in the data, so you need to intervene and adjust the data to be properly used inside the model.

Scaling the features:

It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

1.Loading the dataset

Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

Identify the dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

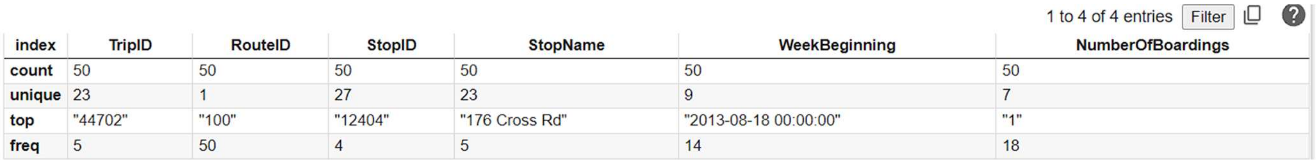
Load the Dataset:

Load your dataset into a Pandas DataFrame**.** The quality and reliability of data can significantly impact the outcomes of vaccine analysis, making it imperative to have robust data loading procedures in place.

Program:

```
import pandas as pd

import numpy as np

import csv

import matplotlib.pyplot as plt

route_details=pd.read_csv("/content/20140711.csv",quoting=csv.QUOTE_NONE)

route_details= route_details.sample(n=50,replace=True)

route_details.describe()
```

Output:

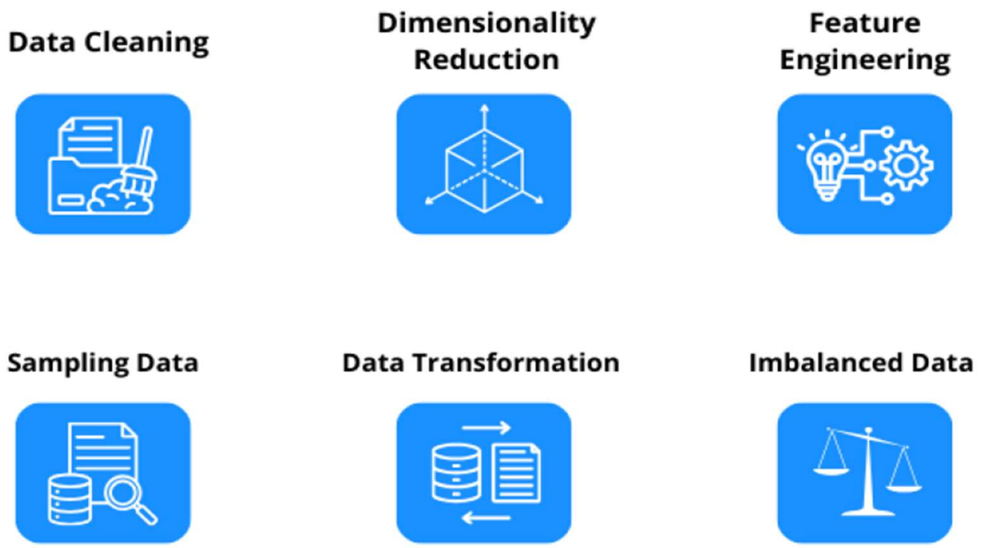


index	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
count	50	50	50	50	50	50
unique	23	1	27	23	9	7
top	"44702"	"100"	"12404"	"176 Cross Rd"	"2013-08-18 00:00:00"	"1"
freq	5	50	4	5	14	18

Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format.

6 techniques for Data Preprocessing



Data cleaning:

This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missing values.

Feature Scaling:

Normalize or standardize numerical features to bring them to a common scale. Common methods include Min-Max scaling (scaling features to a specific range) and z-score normalization (scaling features to have a mean of 0 and a standard deviation of 1).

Feature Engineering:

Create new features or modify existing ones to capture more meaningful information from the data. This may involve mathematical transformations, interaction terms, or aggregations.

Data transformation:

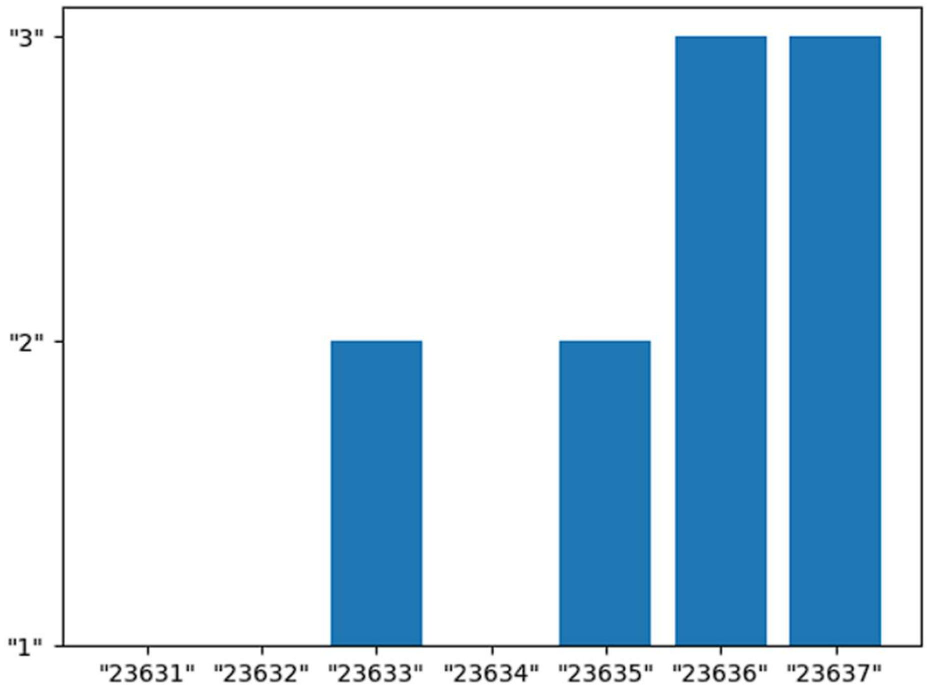
It is a critical aspect of data preprocessing that involves converting and modifying the data to make it more suitable for analysis. It can help improve the performance of machine learning models, enhance the interpretability of the data, and ensure that it aligns with the assumptions of certain statistical techniques.

Exploring the data:

Program:

```
chart1=pd.read_csv("/content/20140711.csv",quoting=csv.QUOTE_NONE,nrows=25)
xaxix = chart1['TripID']
yaxix = chart1['NumberOfBoardings']
plt.bar(x=xaxix,height=yaxix)
plt.show()
```

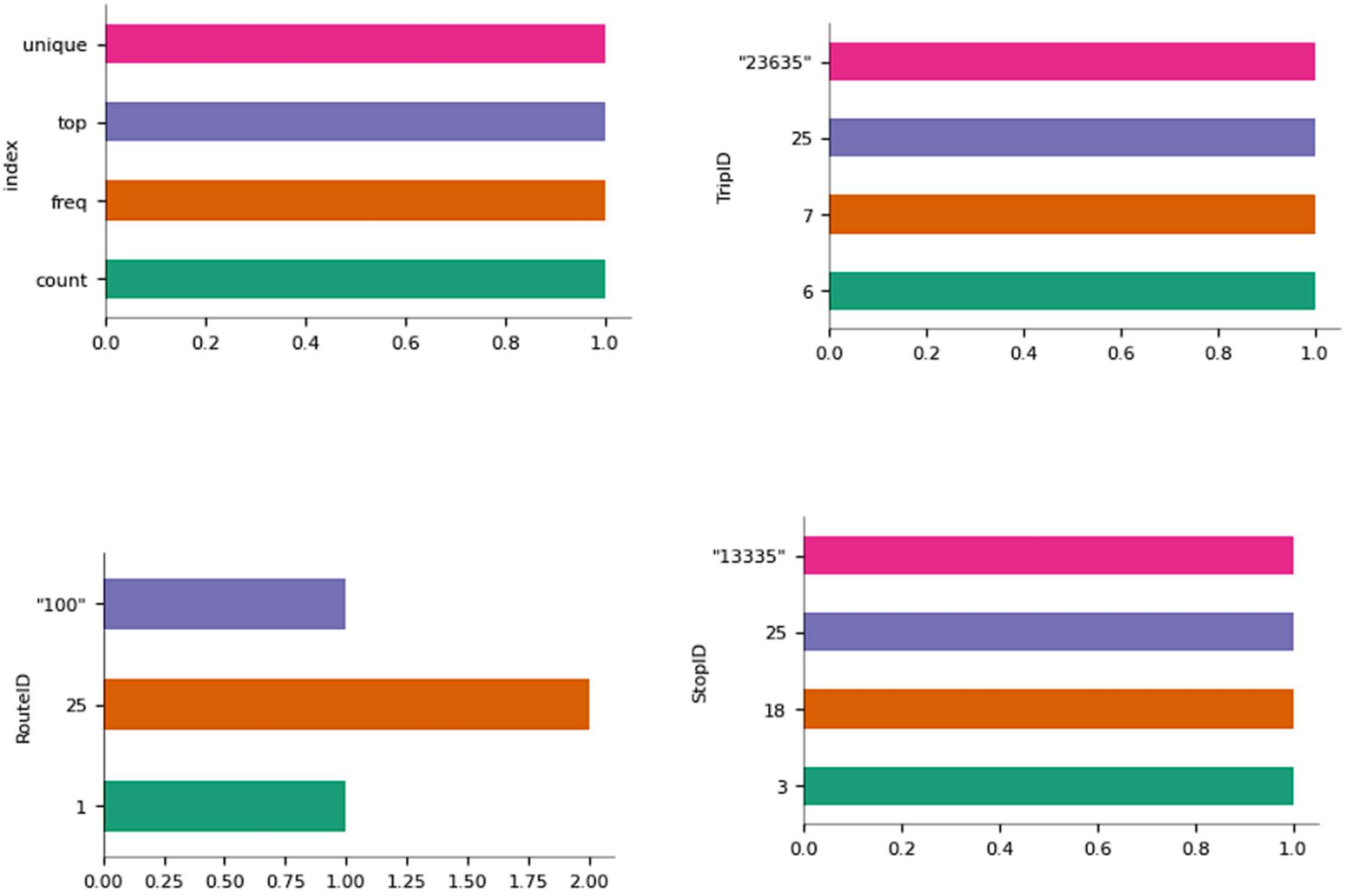
Output:



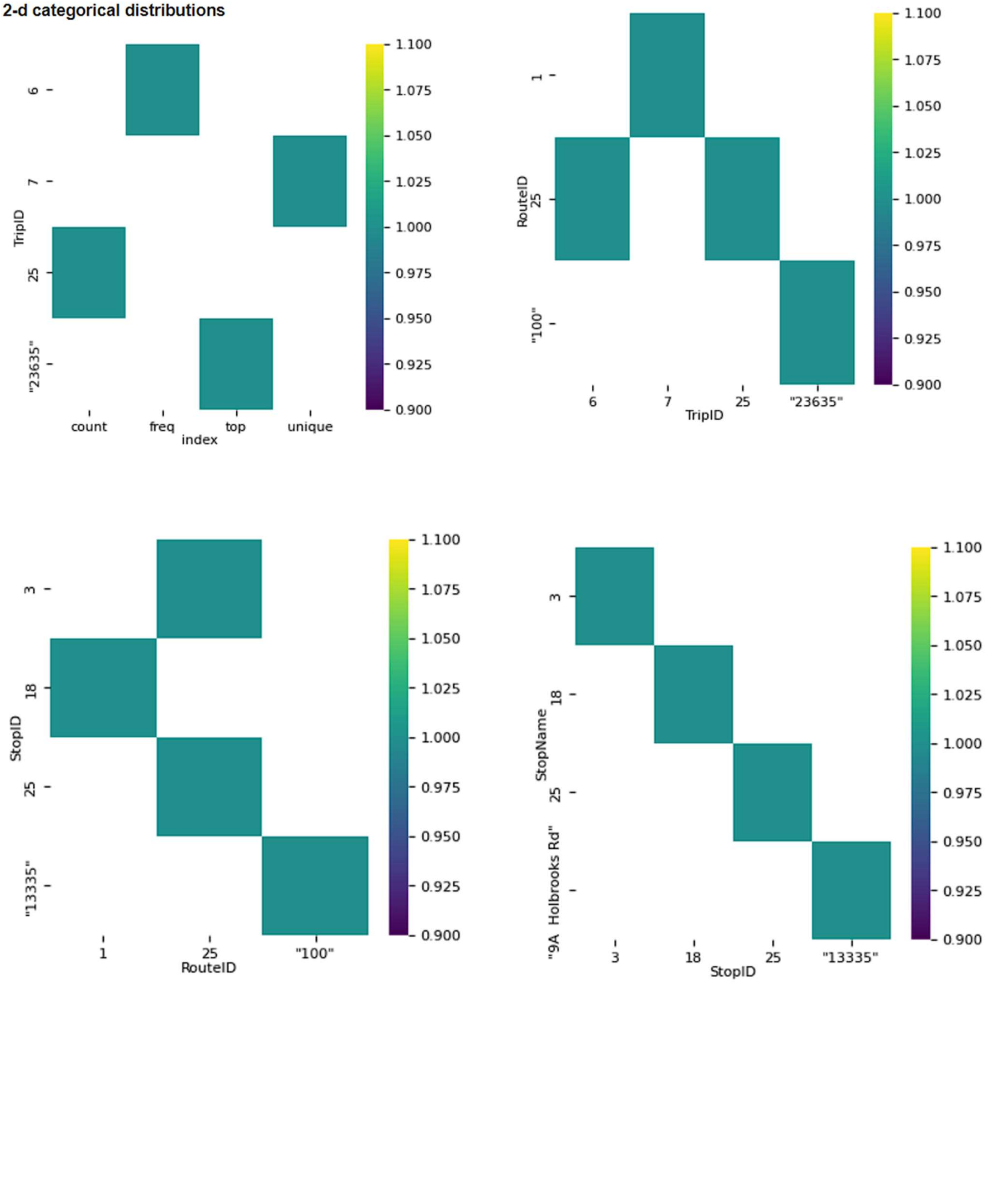
Program:

chart1.describe()

Output:



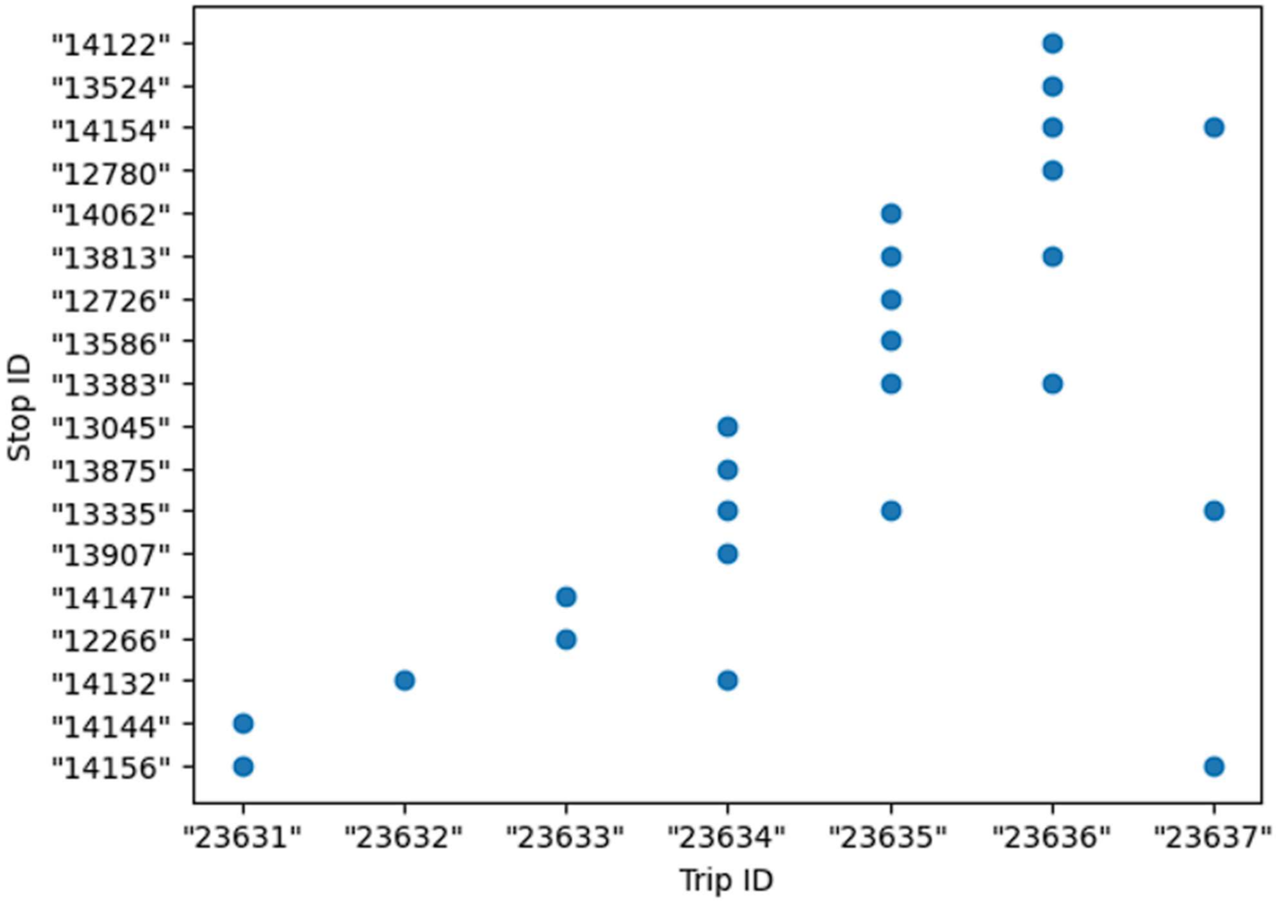
2-d categorical distributions



Program:

```
xaxix = chart1['TripID']

yaxix = chart1['StopID']

plt.scatter(xaxix,yaxix)

plt.xlabel("Trip ID")

plt.ylabel("Stop ID")

plt.show()
```

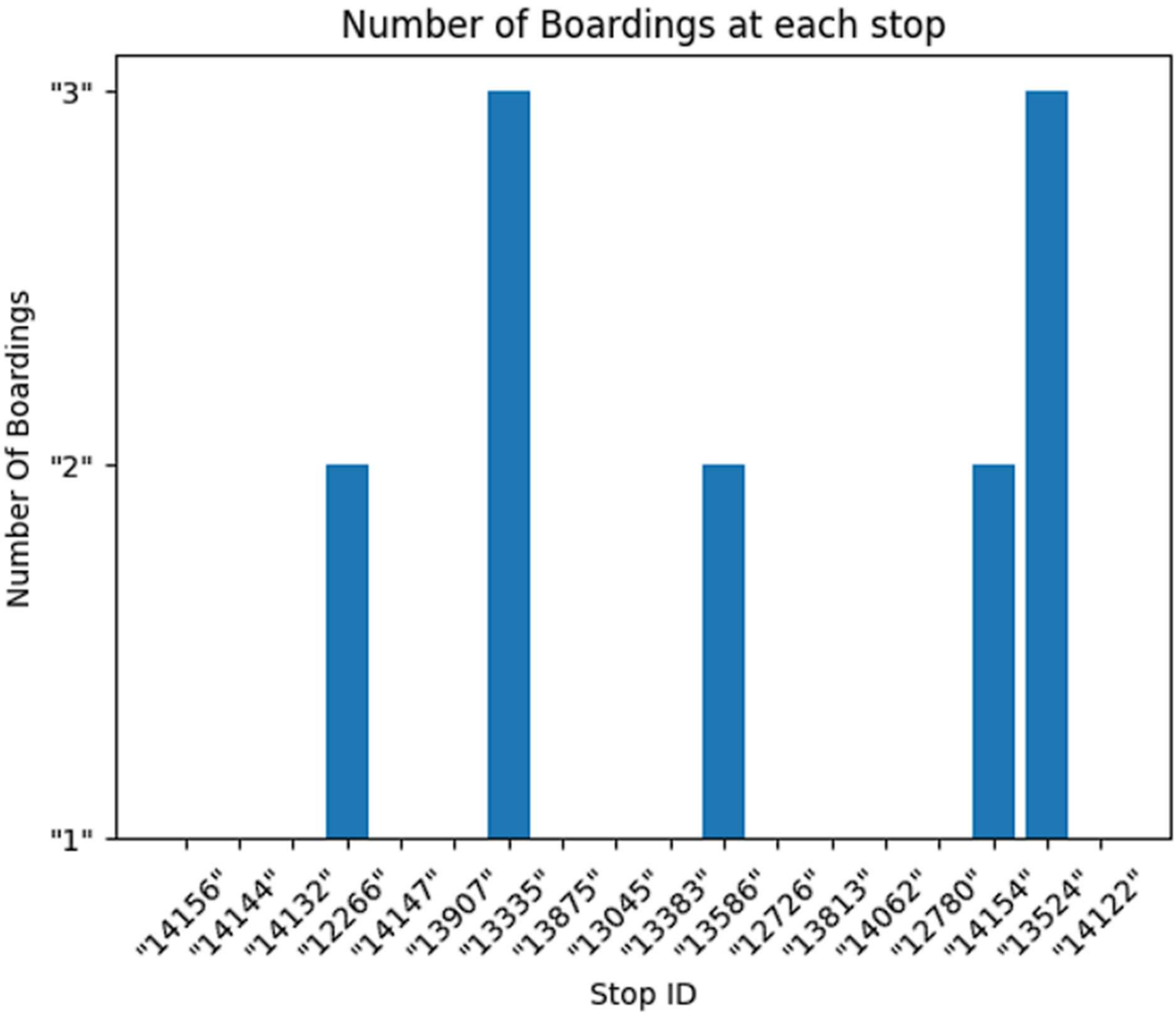
Output:



Program:

```
xaxix = chart1['StopID']
yaxix = chart1['NumberOfBoardings']
plt.bar(xaxix,yaxix)
plt.title("Number of Boardings at each stop")
plt.xlabel("Stop ID")
plt.ylabel("Number Of Boardings")
plt.xticks(rotation=45)
plt.show()
```

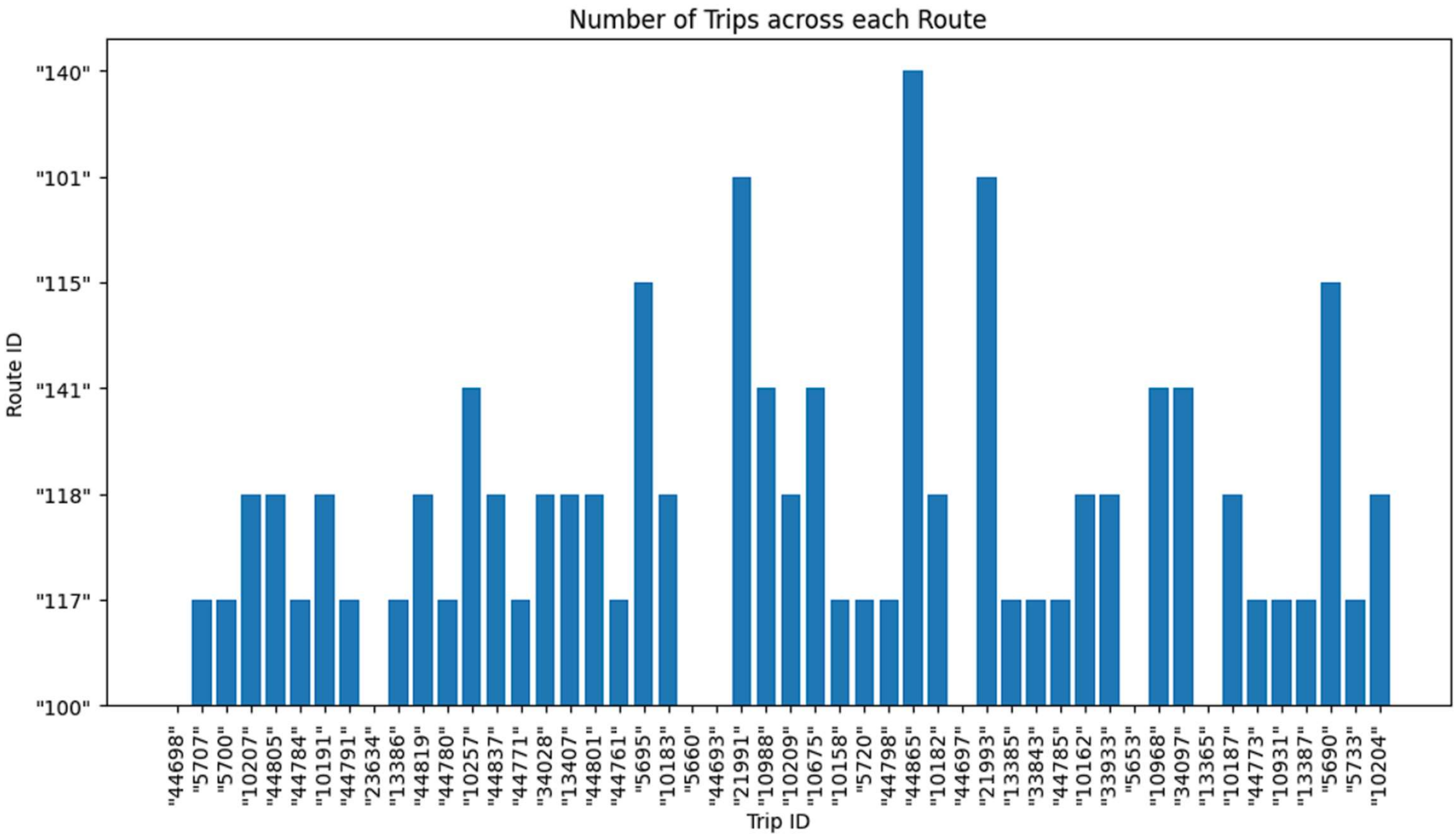
Output:



Program:

```
yaxix = route_details['RouteID']

xaxix = route_details['TripID']

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

plt.bar(xaxix,yaxix)

plt.title("Number of Trips across each Route")

plt.xlabel("Trip ID")

plt.ylabel("Route ID")

plt.xticks(rotation=90)

plt.show()
```

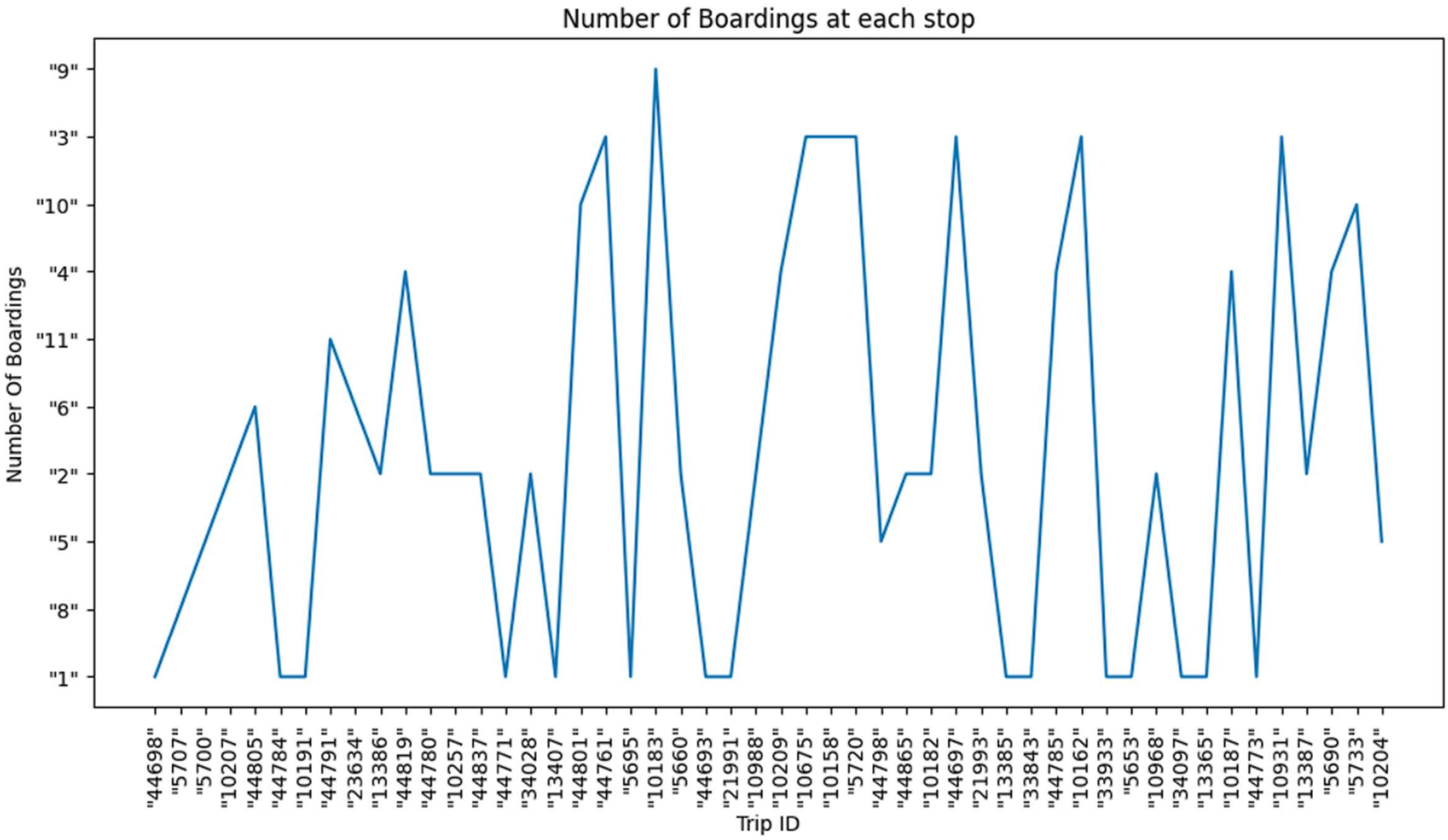
Output:



Program:

```
xaxix = route_details['TripID']

yaxix = route_details['NumberOfBoardings']

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

plt.plot(xaxix,yaxix)

plt.title("Number of Boardings at each stop")

plt.xlabel("Trip ID")

plt.ylabel("Number Of Boardings")

plt.xticks(rotation=90)

plt.show()
```

Output:



Conclusion:

Data loading and preprocessing for public transport efficiency analysis serve as the critical initial steps that empower researchers and government traffic department to make informed decisions about traffic situation, daily routes, rush routes, and safety. The quality, accuracy, and suitability of the data at this stage are pivotal in determining the success of subsequent analyses. Through diligent and systematic data handling, we can harness the power of data-driven insights to address public traffic challenges and contribute to the betterment of public safety.