TOPIC:PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS

BY DINESH RAJA M

PHASE 2

AIM:

Using machine learning algorithm to predict services disruption and analyze passenger sentiment can indeed be to valuable approach to improving transportation services here's how you can approach this:

Data Collection:

Gather historical data related to service disruptions, delays, or incidents in your public transportation system. This data should include details like time, location, nature of the disruption, and its impact.

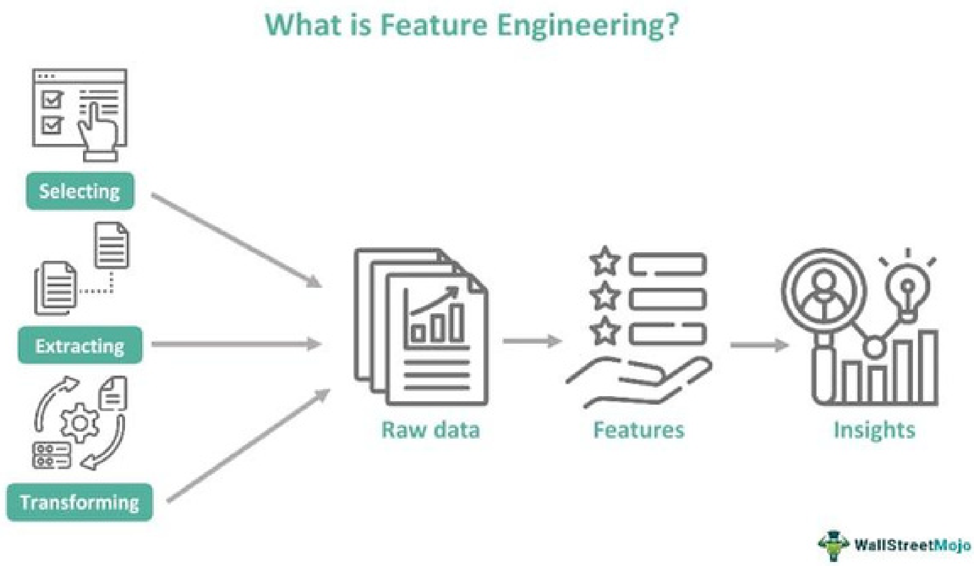
Collect passenger feedback data, including comments, reviews, and ratings. This feedback can be obtained from surveys, social media, or dedicated customer feedback channels.

Data Preprocessing:

Clean and preprocess the collected data. This includes handling missing values, text normalization for feedback data, and feature engineering for both service disruption and feedback datasets.

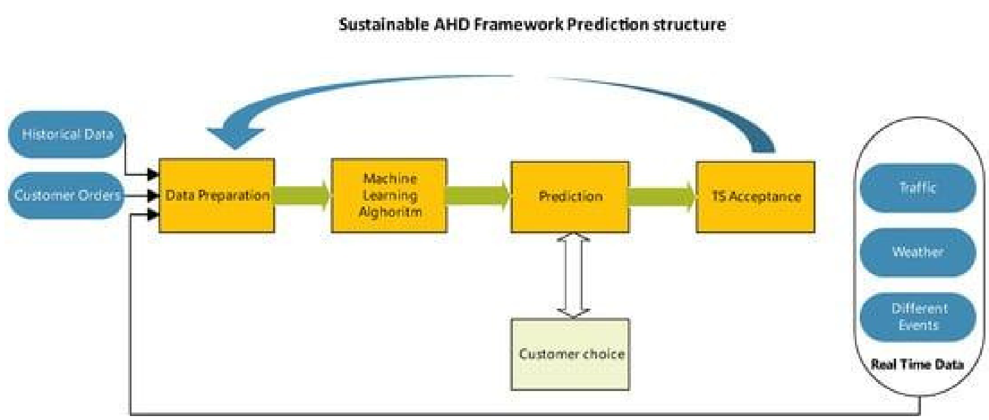
Feature Engineering:

Extract relevant features from the data that can be used as input to machine learning models. For service disruption prediction, features might include historical incident data, weather conditions, and public events. For sentiment analysis, features might include text sentiment scores, word embeddings, and user demographics.



Machine Learning Model Selection:

Choose appropriate machine learning algorithms for your specific tasks. For service disruption prediction, you might use time-series forecasting methods or classification algorithms. For sentiment analysis, natural language processing (NLP) techniques like sentiment analysis models (e.g., BERT, LSTM) can be beneficial.



Training and Validation:

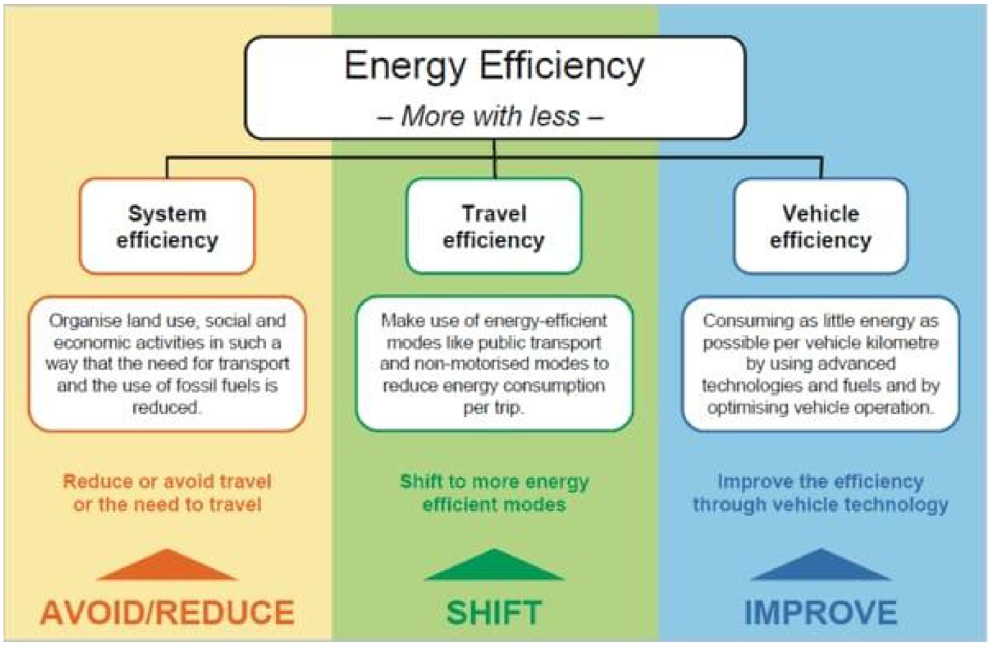
Split your data into training and validation sets. Train your machine learning models on historical data, ensuring that they can learn patterns and relationships.

Tune hyperparameters and evaluate the models using appropriate metrics, such as accuracy, F1-score, or Mean Absolute Error (MAE), depending on the task.

Deployment:

Once your models are trained and validated, deploy them in a production environment where they can continuously analyze incoming data.

Set up automated data pipelines to feed real-time data to your models.



Monitoring and Maintenance:

Regularly monitor the performance of your models in the production environment. Re-train models as new data becomes available to keep them up-to-date.

Implement mechanisms for alerting and handling model failures or degradation in performance.

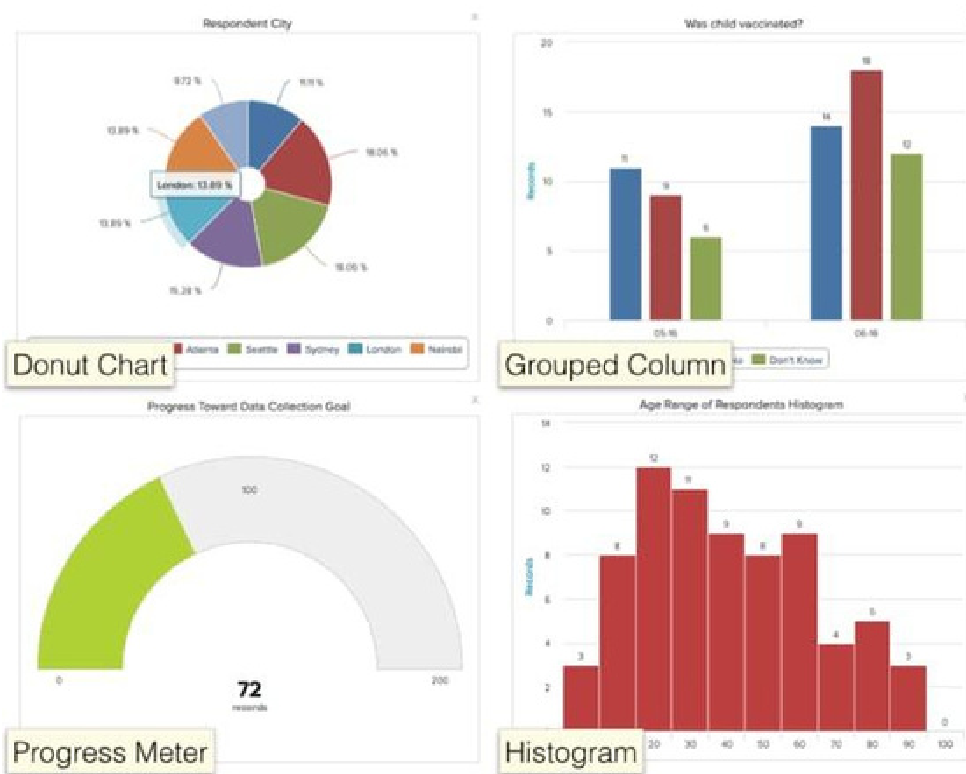
Feedback Integration:

Use the sentiment analysis models to process passenger feedback in real-time. This can help in identifying emerging issues and responding promptly.

Link sentiment analysis results with specific incidents or disruptions to gain insights into the passenger experience.

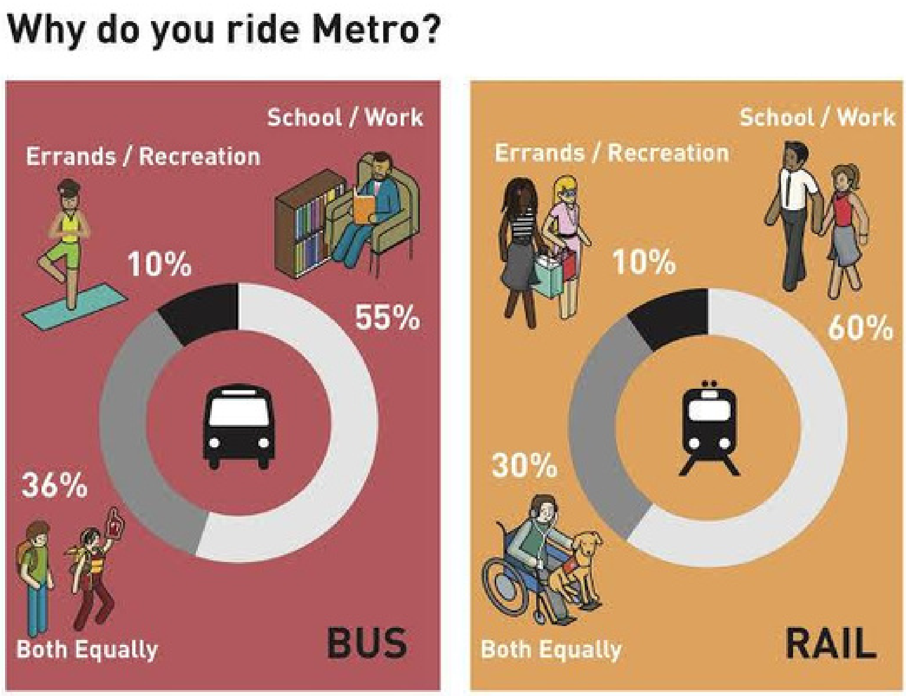
Visualization and Reporting:

Create dashboards or reports that provide insights into service disruptions and passenger sentiment. Visualization tools can help stakeholders understand the data better.



Decision Support:

Use the insights from your machine learning models to make informed decisions about resource allocation, service improvements, and communication strategies.



Iterate and Improve:

Continuously gather feedback from passengers and stakeholders to improve your models and the overall transportation service.

Remember that the success of such a system depends on the quality and quantity of data, as well as the selection and fine-tuning of machine learning models. Additionally, ensure that you address privacy and ethical considerations when handling passenger feedback data and sharing insights with the public.

Machine Learning Algorithm For Public Transportation:

Import necessary libraries

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score
```

from textblob import TextBlob # for sentiment analysis

Additional libraries for data preprocessing, feature engineering, and model selection

Step 1: Data Collection

Load historical service disruption data and passenger feedback data

feedback_data = pd.read_csv('passenger_feedback_data.csv')

Step 2: Data Preprocessing

Clean and preprocess the data (e.g., handle missing values, format dates)

Step 3: Feature Engineering

Extract relevant features from the data

Step 4: Machine Learning Model Selection (Service Disruption Prediction)

Split data into training and testing sets

X = disruption_data.drop('target_column', axis=1)

y = disruption_data['target_column']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Choose and train a machine learning model

model = RandomForestClassifier()

```
model.fit(X_train, y_train)
```

Step 5: Model Evaluation (Service Disruption Prediction)

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Model Accuracy: {accuracy}')
```

Step 6: Deployment - Set up a real-time data pipeline and deploy the model

Step 7: Monitoring and Maintenance

Step 8: Sentiment Analysis (Feedback Integration)

Process passenger feedback using sentiment analysis

```
def analyze_sentiment(text):

blob = TextBlob(text)

sentiment_score = blob.sentiment.polarity

return sentiment_score
```

Step 9: Visualization and Reporting

Step 10: Decision Support

Step 11: Iteration and Improvemen