**PHASE 2:**

INNOVATION

**Topic: Public Transport Analysis**

**1. Data Collection** :

Gather historical service disruption data and passenger feedback. This data should include information about when disruptions occurred, the type of disruption, and passenger sentiment.

**2.Data pre processing:**

- Preprocess the passenger feedback, including text cleaning, tokenization, and removing stopwords.

- Label the sentiment in the feedback data (e.g., positive, negative, neutral).

- Prepare the data for training and testing, ensuring it's in a format suitable for machine learning models.

**3. Feature Engineering:**

- Extract relevant features from passenger feedback, such as sentiment scores, text vectorization (e.g., TF-IDF or word embeddings), and any other useful features.

- Combine these features with the service disruption data, creating a feature matrix.

**4. Machine Learning Model:**

- Choose an appropriate machine learning algorithm for sentiment analysis, such as a text classification algorithm (e.g., Naive Bayes, Logistic Regression, or a neural network).

- Train the model on the labeled passenger feedback data.

**5. Prediction:**

- Use the trained model to predict sentiment for incoming passenger feedback.

- Monitor feedback in real-time, and when a new piece of feedback is received, pass it through the model to predict its sentiment.

**6. Service Disruption Prediction**:

- For predicting service disruptions, you can use time series analysis, anomaly detection, or other relevant algorithms to identify patterns in historical disruption data.

- When a potential disruption is detected, consider using the passenger sentiment as an additional input to assess the impact of the disruption on passengers.

**7. Alert and Response:**

- Set up alerting mechanisms to notify relevant personnel when potential service disruptions or negative passenger sentiments are detected.

- Implement a response strategy to address disruptions and manage passenger sentiment effectively.

**Here's a simplified Python code snippet for the sentiment analysis part:**

Python code:

# Sample code for sentiment analysis using NLTK

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Initialize the sentiment analyzer

sia = SentimentIntensityAnalyzer()

def analyze\_sentiment(text):

sentiment\_scores = sia.polarity\_scores(text)

# Classify sentiment as positive, negative, or neutral based on scores

if sentiment\_scores['compound'] >= 0.05:

return 'Positive'

elif sentiment\_scores['compound'] <= -0.05:

return 'Negative’

else:

return 'Neutral'





