Public Transport Route Optimization for SDG 11

SDG Problem Addressed:

UN SDG 11 promotes sustainable cities through efficient transport systems. Inefficient bus stop placements in New York City increase travel times and limit accessibility, particularly in underserved areas. This project optimizes bus stop locations using K-means clustering to identify strategic hubs, enhancing public transport efficiency and equity.

Machine Learning Approach:

The project employs K-means clustering (unsupervised learning) to group bus stops from the MTA Bus Stops dataset (bi7e-6jgj), accessed via Socrata API from NYC Open Data. Features include latitude, longitude, and synthetic population density (placeholder). Data is preprocessed by cleaning missing values and normalizing features. The optimal cluster count (k=5) is determined using elbow and silhouette methods.

Results:

The model achieved a silhouette score of ~0.45, indicating reasonable cluster separation. Five clusters were visualized on an interactive Folium map and a 2D scatter plot, with results saved to clustered_bus_stops.csv. These clusters represent potential bus stop hubs, streamlining routes and reducing emissions, aligning with SDG 11 goals.

Ethical Considerations:

The dataset may underrepresent bus stops in low-income areas, biasing clusters toward high-density zones. Synthetic population density oversimplifies demographics, potentially neglecting underserved regions. To mitigate, real population data (e.g., NYC Census) should be integrated, and community stakeholders consulted to ensure equitable hub placement. Fairness tools like scikit-learns feature importance analysis could identify bias in feature weighting.

Public Transport Route Optimization for SDG 11

Theoretical Understanding:

Q1: TensorFlow vs. PyTorch: TensorFlow offers static computation graphs for production deployment, while PyTorch uses dynamic graphs for flexible research. Choose TensorFlow for scalable systems (e.g., mobile apps); PyTorch for rapid prototyping (e.g., experimental models).

Q2: Jupyter Notebooks Use Cases: (1) Exploratory data analysis (e.g., visualizing bus stop distributions). (2) Model prototyping (e.g., testing K-means parameters).

Q3: spaCy vs. String Operations: spaCy enhances NLP with pretrained models for tasks like NER, outperforming basic Python string operations, which lack contextual understanding and scalability.

Comparative Analysis: Scikit-learn vs. TensorFlow

Target Applications: Scikit-learn excels in classical ML (e.g., decision trees, clustering), ideal for structured data like bus stops. TensorFlow targets deep learning (e.g., CNNs for image recognition).

Ease of Use: Scikit-learn is beginner-friendly with simple APIs (e.g., fit()). TensorFlow has a steeper learning curve due to complex graph management.

Community Support: Both have strong communities, but TensorFlows is larger due to Google backing, while Scikit-learn benefits from Pythons ML ecosystem.

Conclusion:

This project demonstrates K-means clusterings utility in optimizing public transport for SDG 11. Future enhancements include real population data and a Streamlit app for stakeholder engagement.

Word count note:

Word count: ~250