Project 10: Public Transportation Analysis

(TechnologyName_Phase2)

Phase 2 : Innovation

Public transportation analysis innovation refers to the development and application of new technologies and methods to improve the efficiency, accessibility, and sustainability of public transportation systems.

ALGORITHM OR STEPS TO PERFORM THE PUBLIC TRANSPORTATION ANALYSIS ON A GIVEN DATASET USING MACHINE LEARNING ALGORITHM:

Performing Public Transportation analysis using machine learning typically involves a series of steps :

1. **Data Collection and Preprocessing:**
    - ✞ Gather the dataset containing relevant information about public transport, such as schedules, routes, ridership data, weather data, and any other pertinent information.
    - ✞ Preprocess the data by cleaning, formatting, and transforming it into a suitable format for analysis.
    - ✞ This may involve handling missing data, outliers, and encoding categorical variables

2. **Feature Selection/Extraction:**
    - ✞ Identify the relevant features (attributes) from the dataset that can be used as inputs for your machine learning model.
    - ✞ Create new features if needed, such as time-based features, geographical features, or derived features that might help improve model performance.

3. **Data Splitting:**
    - ✞ Split the dataset into training, validation, and test sets. Common splits might be 70-80% for training, 10-15% for validation, and 10-15% for testing.

4. **Select Machine Learning Algorithm:**
    - ✞ Choose a suitable machine learning algorithm(s) based on the nature of your analysis.
    - ✞ Common choices include regression, time series forecasting, classification, and clustering algorithms.

5. **Model Building:**
    - ✞ Train your selected machine learning model(s) on the training data.
    - ✞ Fine-tune hyperparameters to optimize model performance, using techniques like cross-validation.

6. **Model Evaluation:**
    - Evaluate your model's performance on the validation dataset using appropriate evaluation metrics. Common metrics for public transport analysis might include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or others depending on your specific goals.
7. **Model Deployment:**
    - Once satisfied with the model's performance, deploy it in a real-world environment if necessary. This might involve integrating it into a software system or a web application.

8. **Monitoring and Maintenance:**
    - Continuously monitor the model's performance in the production environment and retrain it periodically with new data to keep it up-to-date.

9. **Interpretability and Visualization:**
    - Make your results interpretable and actionable by stakeholders. Visualization tools can help in presenting the analysis findings in an understandable way.

10. **Feedback Loop:**
    - Collect feedback from users, operators, or other stakeholders to improve the model and the analysis process continually.

11. **Documentation:**
    - Document the entire analysis process, including data sources, preprocessing steps, model selection, and evaluation metrics. This documentation is crucial for reproducibility and knowledge transfer.

PERFORMING PUBLIC TRANSPORTATION ANALYSIS USING MACHINE LEARNING REQUIRES SPECIFIC HARDWARE AND SOFTWARE RESOURCES:

Hardware Requirements:

1. **Computing Power:**
    - Machine learning models can be computationally intensive, especially deep learning models. You may need access to high-performance CPUs or GPUs to train and run complex models efficiently.

2. **Storage :**
    - You need sufficient storage space to store your dataset, model checkpoints, and any intermediate results. SSDs are preferred for faster data access.

3. Memory (RAM):

- The amount of RAM required depends on the size of your dataset and the complexity of your models. Large datasets and deep learning models may require substantial RAM.

4. Parallel Processing:

- If you are dealing with large datasets, distributed computing resources or multi-core processors can speed up data preprocessing and model training.

Software Requirements:

1. Operating System:

- A compatible operating system such as Windows, macOS, or Linux. The choice of the operating system often depends on your preference and the specific machine learning libraries you plan to use.

2. Python:

- Python is a popular programming language for machine learning. You'll need Python installed on your system. Consider using Anaconda or Miniconda to manage Python environments and libraries.

3. Integrated Development Environment (IDE):

- Choose an IDE or code editor that you're comfortable with, such as Jupyter Notebook, Visual Studio Code, or PyCharm, for writing and running your machine learning code.

4. Machine Learning Libraries:

- Scikit-Learn for traditional machine learning algorithms.
- TensorFlow or PyTorch for deep learning.
- Pandas for data manipulation.
- Matplotlib or Seaborn for data visualization.

5. Data Visualization Tools:

- Data visualization tools like Tableau or Power BI is used for exploring and visualizing the dataset before and after analysis.

7. Additional Tools:

- Depending on specific analysis requirements, we may need specialized tools for GIS (Geographic Information System) data analysis or other domain-specific software.

8. Web Frameworks:

- To deploy our public transportation analysis model as a web application, we need web development tools and frameworks like Flask or Django.

9. Version Control:

✞ Consider using version control tools like Git for managing your codebase, especially if you're collaborating with others or want to track changes.

10. Documenting and Reporting Tools:

✞ Tools like Jupyter Notebook, LaTeX, or Markdown editors will be used for documenting the analysis process and creating reports.

Hardware specifications:

✞ Operating system: Windows 10 or 11

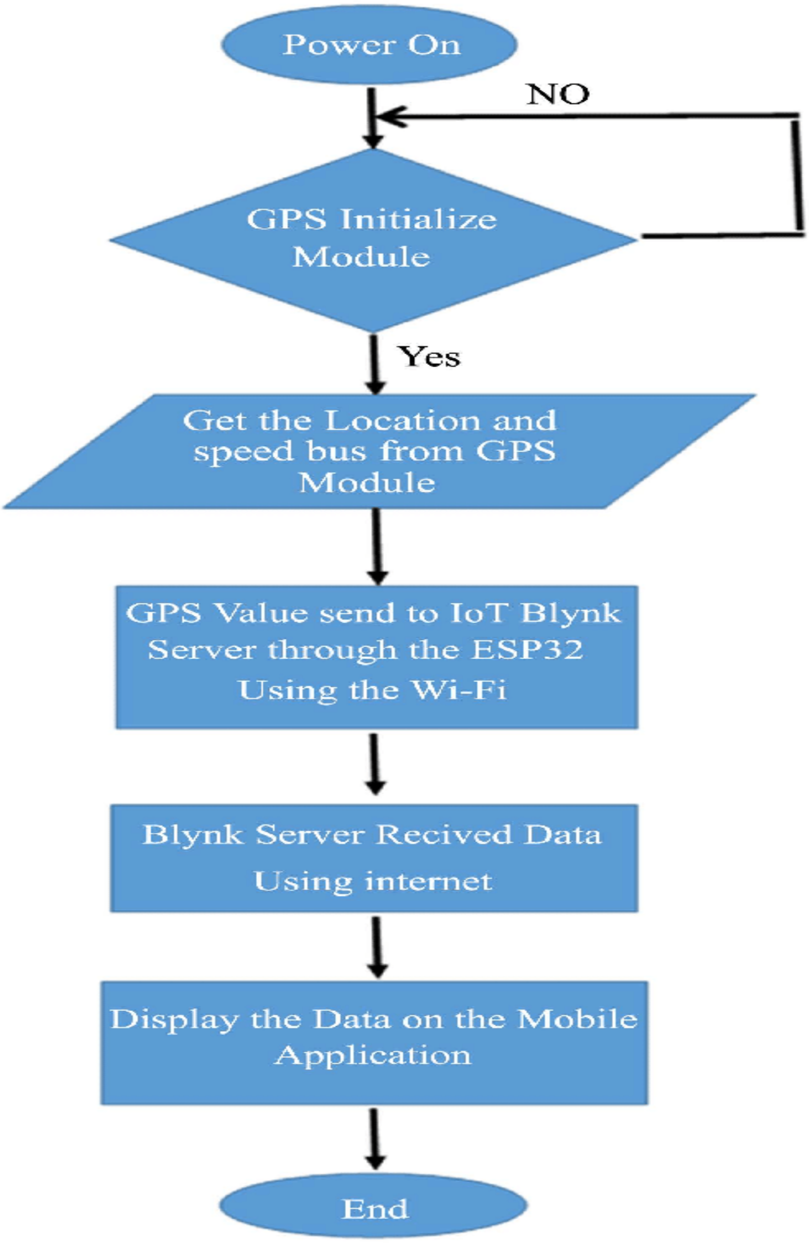
✞ RAM : 64GB

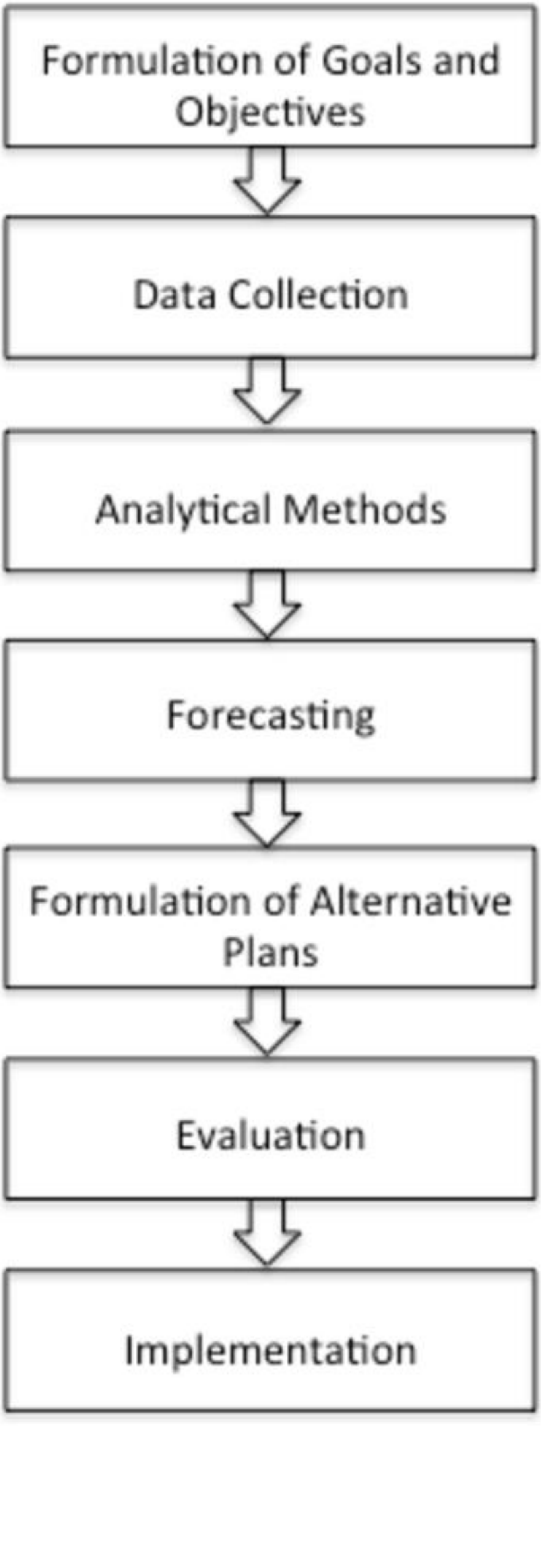
✞ Hard disc or SSD: More than 500 GB **Software specifications:**

✞ Processor : Intel 3rd generation or high or Ryzen with 64 GB Ram ✞ Software's : Python 3.6 or high version ✞ IDE: PyCharm.

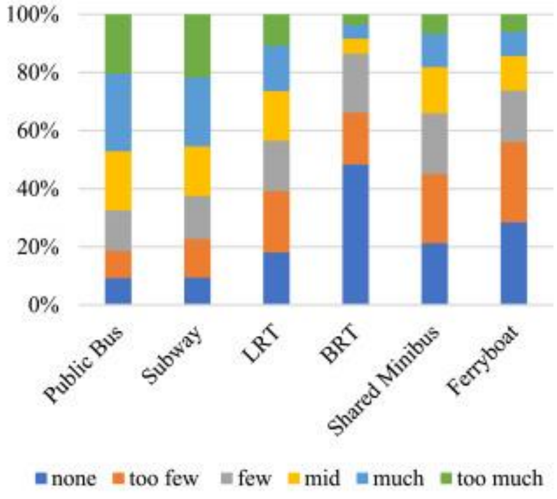
✞ Framework: Flask.

FLOWCHARTS :

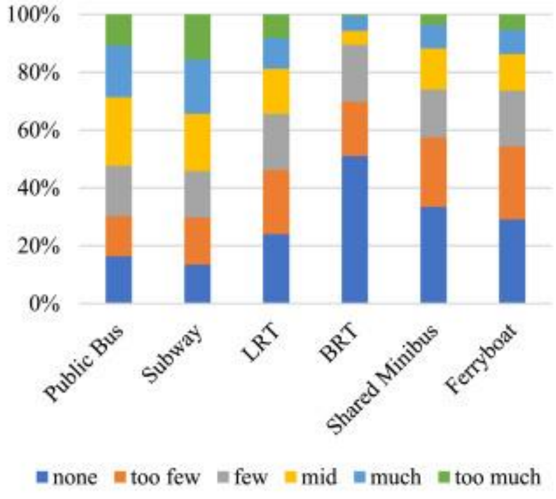

Power On
NO
GPS Initialize Module
Yes
Get the Location and speed bus from GPS Module
GPS Value send to IoT Blynk Server through the ESP32 Using the Wi-Fi
Blynk Server Recived Data Using internet
Display the Data on the Mobile Application
End


Formulation of Goals and Objectives
Data Collection
Analytical Methods
Forecasting
Formulation of Alternative Plans
Evaluation
Implementation

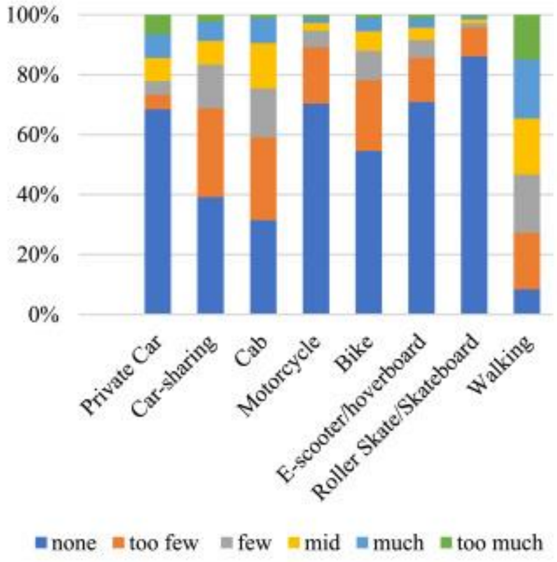
Expected Output :



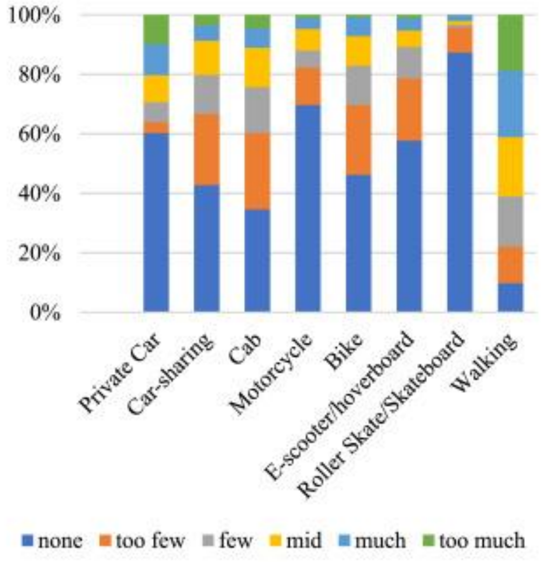
(a)



(b)



(a)



(b)