IOT_PHASE 2

PUBLIC TRANSPORT OPTIMIZATION

TEAM MEMBERS

D.M.BINDUSRI (2021105007)

S.DIVYADARSHNE (2021105010)

R.HARINI (2021105015)

S.MEENA (2021105028)

M.PREETHIKA (2021105036)

Improving arrival time prediction accuracy using machine learning algorithms is a valuable application in transportation and logistics.

Here's a general outline of the steps involved:

Data Collection:

- Gather historical data on arrival times for the specific transportation route or system you're interested in.
- Collect real-time or historical traffic data, weather data, and any other relevant information that may affect travel times.

Data Preprocessing:

- Clean and preprocess the data, handling missing values and outliers.
- Convert categorical data (e.g., day of the week, time of day) into numerical format.
- Create features such as day of the week, time of day, holidays, and special events.

Feature Engineering:

- Extract relevant features that may impact arrival times, such as traffic congestion, road conditions, and weather conditions.
- Consider using feature scaling or normalization if needed.

Data Splitting:

- Split the dataset into training, validation, and test sets. This helps assess the model's performance accurately.

Selecting Machine Learning Algorithms:

- Choose suitable machine learning algorithms for regression tasks. Common choices include Linear Regression, Decision Trees, Random Forest, Gradient Boosting, and Neural Networks.

Model Training:

- Train the selected models on the training dataset.
- Experiment with various hyperparameters to optimize model performance.

Model Evaluation:

- Use the validation dataset to assess the models' performance.
- Evaluate using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or R-squared.

Model Tuning:

- Adjust model parameters and features based on the evaluation results to improve accuracy.

Real-time Data Integration:

- Implement a system for collecting and integrating real-time data, such as live traffic information and weather conditions, into the prediction process.

Deployment:

- Deploy the trained model to make real-time predictions. This can be in the form of an API, a web application, or an embedded system depending on your use case.

Continuous Learning:

- Implement mechanisms to continuously update your model with new data to adapt to changing traffic conditions and patterns.

Monitoring and Maintenance:

- Regularly monitor the model's performance in a production environment and retrain or update it as needed to maintain accuracy.

Feedback Loop:

- Collect user feedback and performance metrics to further improve the system.