Applied Artificial Intelligence | AI-4007 | Project

APRIL 28, 2023

Abdullah Malik : 20i-0930
Daniyal Khan : 20i-1847

Project: Del-01 | Applied AI

Machine learning

Contents

Title: Uber Demand and Supply Prediction System:

Introduction:

The given code is designed to predict the gap between the demand and supply of Uber rides, utilizing various machine learning models. The system processes a dataset consisting of order, weather, and region data, performs feature engineering, trains several machine learning models, and makes predictions using the trained models.

Dataset:

The dataset includes three types of data: order data, weather data, and region data. Order data consists of order_id, driver_id, passenger_id, start_region_hash, dest_region_hash, price, and time. Weather data comprises time, weather, temperature, and PM2.5 (air quality index). Lastly, region data maps region hashes to region ids.

Preprocessing:

The code performs the following preprocessing steps:

1. Extracting Files: The code extracts the dataset files from compressed archives such as RAR and ZIP files.

2. Reading and Concatenating Data: The code reads and concatenates data from multiple CSV files for order and weather data. It also preprocesses the data by extracting relevant date and time information.

3. Merging Data: The code merges order and weather data based on the "Time" column and removes duplicate rows.

4. Reading Region Data: The code reads region data from the "cluster_map" file and maps region hashes to region ids.

Feature Engineering:

The code performs the following feature engineering steps:

1. Supply and Demand Calculation: The code calculates supply (number of drivers) and demand (number of passengers) based on the start_region_hash and time.
2. Merging Features: The code merges supply and demand data with weather data to create the final dataset.
3. Time Feature Extraction: The code processes the final dataset by adding new columns for year, month, day, hour, and minute.
4. Gap Calculation: The code calculates the gap between demand and supply.

Model Training:

The code trains multiple machine learning models on the preprocessed dataset. The models used are:

1. XGBRegressor: An optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable.
2. RandomForestRegressor: A flexible, easy to use machine learning algorithm that produces, even without hyperparameter tuning, a great predictive result most of the time.
3. GradientBoostingRegressor: A machine learning technique for regression problems that produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

4. AdaBoostRegressor: A boosting algorithm that fits a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but adjusts the weights of instances incorrectly predicted by the previous regressor.

5. BaggingRegressor: An ensemble meta-estimator that fits base regressors on random subsets of the original dataset and aggregates their individual predictions to form a final prediction.

6. LinearRegression: A basic linear approach for modeling the relationship between a dependent variable and one or more independent variables.

7. MLPRegressor: A multi-layer perceptron regressor that optimizes the mean squared error using the backpropagation algorithm.

Model Evaluation:

The code evaluates the performance of the trained models using Mean Absolute Error (MAE) on the testing data. The lower the MAE value, the better the model's performance.

Prediction:

The code provides a function to make predictions using the trained models. This function accepts input data and iterates over each model to make predictions.

Conclusion:

The Uber Demand and Supply Prediction System offers an efficient way to analyze the relationship between various factors like region, time, and weather on Uber ride demands and supplies. By training multiple machine learning models and selecting the best-performing one, the system can make accurate predictions and contribute to better resource allocation in the Uber ecosystem.