Amazon Delivery Time Prediction - Project Report

Objective

To build a predictive model that estimates the delivery time for Amazon orders based on geospatial and operational features, and to develop an interactive Streamlit web app for real-time predictions using various regression models.

Problem Statement

Efficient last-mile delivery is crucial for e-commerce companies. Predicting delivery time accurately helps reduce cost, improve logistics, and enhance customer satisfaction. The goal is to create a machine learning model that accurately estimates delivery times based on historical delivery data.

Dataset Description

Features:

- Store_Latitude: Latitude of the store
- Store_Longitude: Longitude of the store
- Drop_Latitude: Latitude of the drop-off location
- Drop_Longitude: Longitude of the drop-off location
- Agent_Rating: Rating of the delivery agent
- Order_Time: Timestamp of order
- Delivery_Time: Actual time taken for delivery (target)

Technologies Used

- Python
- pandas, numpy - Data manipulation
- scikit-learn - Machine learning models
- joblib - Model serialization
- geopy - Distance calculation

- streamlit - Frontend web app

- mlflow - Experiment tracking

Methodology

Data Preprocessing:

- Calculated geodesic distance using geopy

- Extracted order hour from timestamp

- Cleaned and normalized dataset

Feature Selection:

- Distance (km)

- Agent Rating

- Order Hour

Model Selection:

- Linear Regression

- Random Forest Regressor

- Gradient Boosting Regressor

- XGBoost Regressor

Evaluation Metrics

Model Performance:

- Linear Regression: RMSE = 47.73, MAE = 37.04, R^2 = 0.13

- Random Forest: RMSE = 51.34, MAE = 39.35, R^2 = -0.00

- Gradient Boosting: RMSE = 44.35, MAE = 33.75, R^2 = 0.25

- XGBoost: RMSE = 45.24, MAE = 34.45, R^2 = 0.22

Streamlit App Features

- Input store/drop coordinates, agent rating, order hour

- Model selection

- Real-time prediction

- URL: http://localhost:8501/

MLflow Tracking

- MLflow used for experiment tracking and model logging

- Experiment name: Amazon Delivery Time Prediction

- Logged parameters, metrics (RMSE, MAE, R^2), and model artifacts

Challenges Faced

- Modeling spatial distance

- Feature transformations

- Variability in model generalization

- Building modular prediction pipeline

Results

- Gradient Boosting had best accuracy with $R^2 = 0.25$

- Streamlit interface deployed for real-time use

- Useful business insights generated

Future Improvements

- Add traffic/weather data

- Incorporate live tracking

- Serve through cloud APIs

- Expand feature engineering