Amazon Delivery Time Prediction – Detailed Project Report

1. Problem Statement

Timely deliveries are the backbone of customer satisfaction in e-commerce. Companies like Amazon face challenges due to unpredictable factors such as traffic, weather, distance, and agent delays. The goal of this project is to build a machine learning model that can **predict delivery times more accurately**, helping optimize planning, reduce inefficiencies, and improve customer trust.

2. Data Understanding

- Dataset size: **43,739 records, 16 columns**
- Key features:
    - **Agent details** – Age, Rating
    - **Geographical info** – Store & Drop coordinates
    - **Time info** – Order date/time, Pickup time
    - **External factors** – Weather, Traffic
    - **Operational factors** – Vehicle type, Area, Product category
    - **Target variable** – Delivery Time (hours)

3. Data Preprocessing

- **Handled missing values**:
    - Agent_Rating → filled with median
    - Weather → filled with mode
- **Data type conversions**:
    - Converted dates & times to datetime
    - Converted categorical columns to category
- **Feature Engineering**:
    - Distance_km → calculated from store & drop coordinates
    - Pickup_Delay_min → difference between order and pickup

- Order_DayOfWeek → day extracted from date
- Order_Hour → extracted from order time

4. Exploratory Data Analysis (EDA)

- **Delivery patterns by weekday** – Some days show longer delays
- **Impact of traffic** – High traffic significantly increases delivery times
- **Weather influence** – Stormy and foggy weather correlated with longer delays
- **Agent rating** – Higher ratings often linked with faster deliveries
- **Distance factor** – Strong positive correlation between distance and delivery time

5. Model Building

Trained multiple models to compare performance:

1. **Linear Regression** – Baseline, poor fit
2. **Random Forest Regressor** – Strong performance, handled non-linearity well
3. **Gradient Boosting Regressor** – Good, but slightly weaker than Random Forest
4. **XGBoost Regressor** – Best performing model

6. Model Evaluation

Model	**RMSE**	**MAE**	**R^2**
Linear Regression	32.44	25.70	0.6124
Random Forest	22.98	17.59	0.8054
Gradient Boosting	24.20	18.98	0.7842
XGBoost	22.92	17.87	0.8065

Interpretation:

- Linear Regression was too simple.
- Random Forest and Gradient Boosting captured non-linearity well.

- **XGBoost emerged as the best**, explaining ~81% of the variance in delivery times.

7. Deployment (Streamlit App)

- A **user-friendly app** was built in Streamlit.
- Users can input agent details, distance, traffic, weather, etc.
- The app predicts delivery time instantly using the **XGBoost model**.
- Designed for business teams to use predictions without technical knowledge.

8. Conclusion

- Predictive modelling can significantly improve **last-mile delivery planning**.
- **XGBoost** provided the most accurate results.
- The app demonstrates an end-to-end pipeline: from raw data → cleaned dataset → feature engineering → model training → deployment.
- Businesses can use such models to reduce delays, optimize resources, and improve customer satisfaction.

9. Future Scope

- Incorporate **real-time traffic/weather APIs** for live predictions
- Explore **deep learning models** for further accuracy
- Extend the system to **recommend optimal routes** along with delivery time prediction