Swiggy Delivery Time Prediction App

Project Overview

This project is a **machine learning–based web application** that predicts the **estimated delivery time (ETA)** for Swiggy food orders.
Users provide order and delivery details such as distance, pickup time, and order timing, and the app predicts how long the delivery will take.

The application is built using **Python**, **Machine Learning**, and **Streamlit**

🛠️ Applications / Technologies Used

1. Python

- Main programming language used for:
  - Data processing
  - Model loading
  - Prediction logic

Streamlet

- Used to create the **web-based user interface**
- Provides:
  - Input fields (distance, pickup time, festival, weekend, etc.)
  - Predict button
  - Display of predicted ETA

Pandas

- Used for:
    - Creating and managing input data as a Data Frame
    - Formatting data before passing it to the ML model

Joblib

- Used to:
    - Load the pre-trained machine learning model (`swiggy_eta_model.pkl`)
    - Efficiently handle serialized ML models

Machine Learning Model

- A trained regression model saved as:
    - `swiggy_eta_model.pkl`
- Predicts delivery time based on:
    - Distance
    - Multiple deliveries
    - Peak hours
    - Festival & weekend demand
    - Rider and order features

Datetime Module

Used to:

- Automatically get current date, day, and month
- Generate time-based features for prediction

Project Files

File Name	Description
`swiggy app.py`	Main Streamlit application file
`swiggy_eta_model.pkl`	Trained machine learning model
`data.ipynb`	Jupyter notebook used for data analysis and model training
`README.md`	Project documentation

- ## How the Application Works

- User enters delivery details in the Streamlit UI
- Features are processed and engineered (peak hour, delivery load, demand)
- Data is converted into a Pandas DataFrame
- Pre-trained ML model predicts delivery time
- ETA is displayed in minutes

How to Run the Application

Install Required Libraries

pip install streamlit pandas joblib

Run the App:

streamlit run "swiggy app.py"

Sample Features Used for Prediction

- Distance (km)
- Pickup time (minutes)
- Multiple deliveries
- Peak hour indicator
- Festival & weekend flags
- Rider and vehicle details

- Traffic and weather conditions

Output

- Displays **Estimated Delivery Time (in minutes)** on button click

Conclusion

This project demonstrates how **machine learning** can be integrated with a **web application** to solve real-world delivery prediction problems efficiently.