Project Title	**🚗 TripFare : Predicting Urban Taxi Fare with Machine Learning**
Skills Take Away From This Project	• Exploratory Data Analysis (EDA) • Data cleaning and preprocessing • Data Visualization with Matplotlib & Seaborn • Feature Engineering • Regression Model Building • Model Evaluation & Comparison • Hyperparameter Tuning • Streamlit
Domain	Urban Transportation Analytics & Predictive Modeling

📣 Problem Statement

As a Data Analyst at an urban mobility analytics firm, your mission is to unlock insights from real-world taxi trip data to enhance fare estimation systems and promote pricing transparency for passengers. This project focuses on analyzing historical taxi trip records collected from a metropolitan transportation network.

The goal is to build a predictive model that accurately estimates the total taxi fare amount based on various ride-related features. Learners will preprocess the raw data,

engineer meaningful features, handle data quality issues, train and evaluate multiple regression models, and finally deploy the best-performing model using Streamlit.

🎯 Real-World Use Cases:

1. **Ride-Hailing Services** – Fare estimate before ride booking.
2. **Driver Incentive Systems** – Suggest optimal locations and times for higher earnings.
3. **Urban Mobility Analytics** – Fare trends by time, location, and trip type.
4. **Travel Budget Planners** – Predict estimated trip fare for tourists.
5. **Taxi Sharing Apps** – Dynamic pricing for shared rides.

🧠 Problem Type:

Supervised Machine Learning – Regression

Target Variable: `total_amount`

🔧 Tasks & Workflow:

1️⃣ Data Collection

- Dataset - Link
- Download and load dataset using Pandas

2️⃣ Data Understanding

- Explore and understand the dataset using basic pandas functions
- Check for Shape, Datatypes, duplicates , missing values etc

---

3️⃣ Feature Engineering:

Perform feature engineering by creating new columns to identify patterns and trends in the dataset.

📌 Derived Columns(ideas):

- **trip_distance:** Use Haversine formula (from pickup & dropoff coordinates)
- **pickup_day**: Extract weekday/weekend
- **am/pm**: Extract am/pm
- **is_night**: Binary flag for late-night/early-morning trips
- Convert pickup_datetime from **UTC** to **EDT**.

These derived features will help in better analysis. Create indicators to enable the analysis of whether taxi fares fluctuate based on weekends, rush hours or late-night rides.

Additional relevant columns can also be created as needed to enhance the overall analysis and improve model interpretability.

4️⃣ Exploratory Data Analysis (EDA)

Perform both univariate and bivariate analysis to understand the distribution and relationships within the dataset. Some recommended analyses include:

- **Fare vs. Distance:** Examine how fare amounts vary with trip distance.
- **Fare vs. Passenger Count:** Analyze the relationship between fare and the number of passengers.
- **Outlier Detection:** Identify and handle outliers in variables such as fare amount, trip distance, and trip duration.

In addition to these, perform further EDA to uncover meaningful patterns and trends in the data. This could involve:

- Analyzing **fare variations across different times of the day, weekdays vs. weekends, and months.**
- Studying the distribution of **trip distances, trip durations, and pickup hours.**
- Exploring how **fare per mile and fare per minute** behave across different time periods or trip lengths.
- Visualizing **trip counts by pickup hour and pickup day** to identify peak demand periods.
- Investigating the impact of **night rides and weekend trips on fare amounts**.

5️⃣ Data Transformation:

- Handle **outliers** using Z-score or IQR
- Fix **skewness** in continuous variables using different transformations
- **Encode** categorical variables

6️⃣🔍 Feature Selection:

Apply various feature selection techniques such as correlation analysis, Chi-Square test (for categorical variables), etc and feature importance from models like Random Forest to identify the most relevant features for building accurate and efficient regression models.

7️⃣ Model Building

Regression:

- Build at least **5 models** (e.g. Linear Regression, Ridge, Lasso, RandomForest, GradientBoosting)
- Compare using:
    - R^2
    - MSE,RMSE, MAE

🛠️ Hyperparameter Tuning :

Use `GridSearchCV` or `RandomizedSearchCV` to optimize the best-performing model if required.

8️⃣ Finalize Best Models

- Choose best models based on performance metrics
- Save the best performing model(pickle format)

9️⃣📌 Final Task: Build a Streamlit UI

After training and evaluating your regression models:

- Create a Streamlit UI where users can input relevant trip details such as pickup and dropoff locations, passenger count, time of travel, and other trip-related features.

- On submitting the inputs:
  Display the **predicted total fare amount** using your best regression model.

🗒️ Column Descriptions

Column Name	Description
`VendorID`	ID of the taxi provider

`tpep_pickup_datetime`	Date and time when the trip started
`tpep_dropoff_datetime`	Date and time when the trip ended
`passenger_count`	Number of passengers in the taxi
`pickup_longitude`	Longitude where the passenger was picked up
`pickup_latitude`	Latitude where the passenger was picked up
`RatecodeID`	Type of rate (e.g., standard rate, JFK, Newark, negotiated fare)
`store_and_fwd_flag`	Whether the trip data was stored and forwarded
`dropoff_longitude`	Longitude where the passenger was dropped off
`dropoff_latitude`	Latitude where the passenger was dropped off
`payment_type`	Payment method used
`fare_amount`	Base fare amount charged
`extra`	Extra charges (e.g., for peak time, night surcharge, etc.)
`mta_tax`	MTA (Metropolitan Transportation Authority) tax
`tip_amount`	Tip amount paid by the passenger
`tolls_amount`	Toll charges (e.g., bridge/tunnel tolls)
`improvement_surcharge`	Flat fee surcharge (usually $0.30)

`total_amount`	Total trip amount including all fees, tips, and surcharges (Target)

🛠️ Technical Tags

`Pandas,EDA,DataPreprocessing,MachineLearning,RegressionModel,ModelEvaluation,StreamlitApp,Python,ScikitLearn,TrainTestSplit,UserInputInterface,haversine formula, taxi fare prediction, data transformation, model optimization`

📌 Project Deliverables:

- Python Notebook with:
  - Clean code and comments
  - Visualizations
  - Model evaluations
- Streamlit UI

⏳ Timeline

The project should be completed and submitted **within 10 days** from the date it is assigned.

References

Streamlit recording (English)	Special session for STREAMLIT(11/08/2024)
Streamlit Reference doc	Streamlit API reference
Project Live Evaluation	Project Live Evaluation
Capstone Explanation Guideline	Capstone Explanation Guideline
GitHub Reference	How to Use GitHub.pptx
Machine Learning(Eng) Classification and Regression	Project Excellence Series: Guided Lear…
Machine Learning(Tam) Classification and Regression	Project Excellence Series: Guided Lear…
Project Orientation	Project Orientation Session : TripFare : …

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.

Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday-Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

Note: This form will Open only on Saturday (after 2 PM) and Sunday on Every Week

Timing: Monday-Saturday (05:30PM to 07:00PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA

Created By	Verified By	Approved By
Nilofer Mubeen	Shadiya.P.P	Nehlath Harmain