🚖 Uber Trip Analysis

Internship Project Report

📌 Section 1: Project Overview

Project Title:

Uber Trip Analysis

Project Type:

Data Analysis & Machine Learning

Technologies Used:

✅ Python (Data Processing, Machine Learning, Web App)

✅ Flask (Web Application Development)

✅ Pandas & NumPy (Data Manipulation)

✅ Scikit-Learn (Machine Learning)

✅ Matplotlib, Seaborn (Data Visualization)

✅ Joblib (Model Saving & Deployment)

Project Difficulty Level:

✅ Advanced

📌 **Section 2: Project Objective & Need**

🎯 Objective

The aim of this project is to predict Uber trip demand based on historical ride-hailing data using machine learning models. This system helps:

🚖 Estimate demand for Uber trips at different times of the day

📊 Analyze ride-hailing trends using advanced data visualizations

📈 Assist businesses & drivers in making data-driven decisions on fleet distribution

🔍 Why is This Project Needed?

🚕 The demand for ride-hailing services fluctuates throughout the day.

📊 Accurate trip demand prediction helps Uber optimize its fleet for better efficiency.

📉 Prevents surge pricing & availability issues by forecasting peak ride demand.

🛣️ Helps city planners analyze urban mobility patterns for better infrastructure.

📌 Section 3: Dataset Information

📜 Dataset Source:

The dataset is obtained from Uber FOIL (January-February) trip records.

🗂️ Dataset Features (Columns Used for Prediction):

Feature Name	Description
Active Vehicles	Number of Uber vehicles available at a given time.
Hour	The hour of the day (0-23).
Day of the Week	The weekday (Monday-Sunday).
Month	The month of the year.
Weekend Flag	Indicates if the day is a weekend (1) or weekday (0).

📌 Target Variable (What We Predict):

✅ Number of Uber Trips

📌 Section 4: Project Execution - Steps & Implementation

Step 1: Data Collection

✓ Load the dataset using Pandas.

✓ Check for missing values, duplicates, and data types.

Step 2: Data Preprocessing

✓ Convert date column to datetime format.

✓ Extract hour, day of the week, and month from timestamps.

✓ Create a weekend flag (1 for weekends, 0 for weekdays).

✓ Handle missing values and normalize numerical features using StandardScaler.

Step 3: Exploratory Data Analysis (EDA)

✓ Trips per hour of the day (Bar Graph)

✓ Trips per day of the week (Grouped Bar Chart)

✓ Trips vs Active Vehicles (Scatter Plot)

✓ Hourly Trip Demand Heatmap

Step 4: Feature Engineering

✓ Select relevant features:

Active Vehicles

Hour

Day of the Week

Month

Weekend Flag

✓ Scale numerical features using StandardScaler for better ML performance.

Step 5: Model Building

✓ Train a Linear Regression model for trip prediction.

✓ Train a Random Forest model for better accuracy.

✓ Use GridSearchCV for hyperparameter tuning.

Step 6: Model Evaluation

✓ Use Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score to measure accuracy.

✓ Compare actual vs. predicted values using a scatter plot.

Step 7: Data Visualization

✓ Pie Chart → Distribution of trips vs active vehicles.

✓ Bar Graph → Comparison of predicted trips & active vehicles.

✓ Line Chart → Trend analysis of trips over 24 hours.

✓ Scatter Plot → Relationship between active vehicles & trip demand.

✓ Heatmap → Visualizing hourly trip demand patterns.

Step 8: Flask Web Application

✓ Built a Flask web app for users to input date, time, and active vehicles.

✓ Predicts trip demand dynamically based on user input.

✓ Displays related visualizations (bar charts, pie charts, line charts, heatmaps).

✓ Fully responsive & modern UI (Dark Theme with Gold Accents).

📌 Section 5: Results & Conclusion

✅ Key Achievements

🔹 Successfully built a machine learning model to predict Uber trip demand.

🔹 Integrated real-time interactive visualizations (pie charts, bar graphs, line charts).

🔹 Developed a Flask-powered web app for seamless user interaction.

🔹 Created a modern, responsive UI/UX with dark mode & gold accents.

🔍 Future Enhancements

✔ Improve accuracy using XGBoost & Deep Learning models.

✔ Add real-time Uber API integration for dynamic trip predictions.

✔ Deploy on Cloud (AWS, Heroku, or Streamlit) for public access.