Uber Data Analysis Using Python ,SQL Server and PowerBI

# Problem Statement

Understand and analyze Uber trip data to uncover insights on trip distances, fare distributions, vendor performance, and customer behavior. Build an end-to-end ETL pipeline that automates data ingestion and transformation, then visualize key metrics to inform operational and strategic decisions.

# Solution

An automated pipeline extracts raw trip data from CSV, cleans and enriches it using Python (Pandas/NumPy), and loads the results into a SQL Server data warehouse via bulk inserts. SQL queries compute core metrics, and an interactive Power BI dashboard presents revenue trends, vendor comparisons, and usage patterns for stakeholders.

# About Dataset

- \*\*Uber\_Data.csv\*\*: Contains timestamped records of trips, including pickup/dropoff times and locations, trip distance, fare amount, tip, payment type, and vendor ID.  
- \*\*Data Warehouse Schema\*\*: A T-SQL script defines fact tables for trips and dimension tables for dates, vendors, and payment types to support performant analytics.

# Data Cleaning & Transformation

1. \*\*Null Handling\*\*: Drop or impute missing values in fare, tip, and distance columns.
2. \*\*Type Casting\*\*: Convert strings to datetime for pickup/dropoff; numeric fields to appropriate numeric types.
3. \*\*Feature Engineering\*\*:  
   - \*\*Trip Duration\*\* = `dropoff\_datetime – pickup\_datetime` (in minutes)  
   - \*\*Revenue per Mile\*\* = `(fare\_amount + tip\_amount) / trip\_distance`  
   - \*\*Distance Band\*\*: Categorize trips (e.g., Short: ≤ 2 mi; Medium: 2–10 mi; Long: > 10 mi)
4. \*\*Data Warehouse Loading\*\*: Use SQL Server’s BULK INSERT to load the cleaned CSV into staging, then INSERT INTO fact/dimension tables.

# Calculated Functions & Queries

* \*\*Total Trips\*\* = COUNT(\*) on the trips fact table.
* \*\*Total Revenue\*\* = SUM(fare\_amount + tip\_amount)
* \*\*Average Fare\*\* = AVG(fare\_amount)
* \*\*Total Tips per Vendor\*\* = SUM(tip\_amount) grouped by vendor\_id
* \*\*Trips by Payment Type\*\* = COUNT(\*) grouped by payment\_type

# Key Performance Indicators (KPIs)

|  |  |
| --- | --- |
| KPI | Definition / SQL Expression |
| Total Trips | COUNT(\*) |
| Total Revenue | SUM(fare\_amount + tip\_amount) |
| Average Trip Distance | AVG(trip\_distance) |
| Total Tips per Vendor | SUM(tip\_amount) OVER (PARTITION BY vendor\_id) |
| Revenue by Distance Band | SUM(fare\_amount + tip\_amount) GROUP BY distance\_band |
| Trips by Hour of Day | COUNT(\*) GROUP BY DATEPART(hour, pickup\_datetime) |

# Visualization & Results

The Dashboards/ folder contains Power BI .pbix files that connect directly to the SQL Server warehouse. Key visuals include:  
- Revenue vs. Distance: Column chart showing total revenue for each distance band.  
- Vendor Tip Comparison: Bar chart comparing total tips by vendor.  
- Payment Type Breakdown: Donut chart of trip counts by payment method.  
- Hourly Trip Patterns: Line chart of trip volumes across 24 hours.  
- Fare & Distance Distribution: Histogram overlays for fare amounts and distances.

# Conclusion

This project delivers a scalable ETL pipeline that transforms raw Uber trip logs into structured data, enabling deep analysis via SQL and compelling visual storytelling in Power BI. Stakeholders gain actionable insights on revenue drivers, vendor efficiency, and customer behavior.

# Future Work

* Real-Time Streaming: Integrate Kafka or Azure Event Hubs to process live trip events.
* Predictive Modeling: Build ML models to forecast demand by region and time.
* Geospatial Analytics: Incorporate GIS maps for pickup/dropoff heatmaps and route optimization.
* Cost Optimization: Analyze cost per mile and recommend dynamic pricing strategies.