Rideshare-uber dataset description

**📄 Feature Descriptions**

**Datetime & Temporal Features**

* **hour** *(int)* → Hour of the day (0–23). Helps capture daily patterns in ride demand and pricing.
* **day** *(int)* → Day of the month (1–31). Useful for monthly seasonality trends.
* **month** *(int)* → Month of the year (1–12). Useful for long-term seasonality (e.g., December holidays).
* **datetime** *(datetime)* → Full timestamp of the ride request. Original raw feature.
* **day\_of\_week** *(int)* → Encoded as 0=Monday, 6=Sunday. Captures weekly seasonality.
* **is\_weekend** *(binary int)* → 1 if the ride is on Saturday/Sunday, else 0. Often higher prices on weekends.
* **time\_of\_day** *(categorical)* → Categorical bucket (Morning, Afternoon, Evening, Night). Derived from hour.

**Location Features**

* **source** *(object)* → Pickup location (e.g., “Haymarket Square”). Important for spatial demand patterns.
* **destination** *(object)* → Drop-off location (e.g., “North Station”). Paired with source to analyze routes.
* **route** *(object)* → Concatenation of source + destination. Helps capture demand between popular routes.

**Ride & Business Features**

* **cab\_type** *(object)* → Type of ride provider (e.g., Uber, Lyft).
* **name** *(object)* → Specific ride type (e.g., UberX, UberPool, Lyft Line). Different categories have different pricing.
* **price** *(float)* → Ride fare (target variable for prediction).
* **distance** *(float)* → Distance of the trip in miles/km. Strong predictor of price.
* **log\_distance** *(float)* → Log-transformed distance to reduce skewness and normalize feature distribution.
* **surge\_multiplier** *(float)* → Surge pricing factor (e.g., 1.0 = no surge, 2.0 = double price). Directly influences ride cost.

**Weather Features**

* **temperature** *(float)* → Measured temperature at ride time (°C or °F).
* **apparentTemperature** *(float)* → “Feels like” temperature considering humidity and wind.
* **feels\_like\_diff** *(float)* → Difference between temperature and apparentTemperature. Captures discomfort due to weather.
* **precipIntensity** *(float)* → Rainfall intensity (mm/hr or equivalent). Higher values indicate heavy rain.
* **precipProbability** *(float)* → Probability of precipitation (0–1).
* **humidity** *(float)* → Relative humidity (0–1).
* **windSpeed** *(float)* → Wind speed at ride time.
* **windGust** *(float)* → Sudden peak wind speed.
* **visibility** *(float)* → Distance of visibility (miles/km). Low visibility can impact demand and safety.
* **cloudCover** *(float)* → Fraction of sky covered by clouds (0–1).
* **short\_summary** *(object)* → Brief textual weather condition (e.g., “Mostly Cloudy”, “Rain”).
* **bad\_weather** *(binary int)* → Engineered feature; 1 if bad weather conditions (rain >50%, low visibility, or high cloud cover), else 0.

**🏷️ Target Variable**

* **price** *(float)* → Ride fare amount in currency. This is the **target variable** we aim to predict using the above features.

👉 This document can go directly into your **project report / GitHub README** so recruiters clearly see:

* What features you used
* Why they’re important
* How you engineered additional features