Uber Ride Bookings Feature Engineering Workflow Report

1. Objective

The goal of this ETL workflow is to prepare clean, structured, and feature-engineered data from raw Uber ride booking records stored in Amazon S3.
This processed dataset is optimized for downstream predictive analytics and machine learning models, specifically focused on the target variable: booking value (ride fare).

2. Data Source

- **Raw Input**: CSV file (ncr_ride_bookings.csv) stored in Amazon S3.
- **Schema (original)** included ride timestamps, vehicle type, pickup/drop locations, ride metrics (VTAT, CTAT, distance), cancellation flags, ratings, and payment method.

3. Workflow Architecture

1. Ingestion (AWS Glue)

- Used a Glue crawler to catalogue raw data stored in S3.
- Created a Glue ETL job to read the CSV into a DynamicFrame.

2. Schema Normalization

- Converted date to date, time to timestamp datatypes.
- Converted ride statistics and monetary values to numeric types (float, double).
- Cancellation and incomplete ride flags were cleaned into integers (0/1) to be converted to binary later.

3. Feature Engineering (PySpark)

- **Target Variable**: booking_value (ride fare).
- All transformations were designed to ensure predictors are numeric and consistent:
    - **Categorical encodings**:
        - vehicle_type → label encoding (e.g., ebike=1, sedan=2, …).
        - payment_method → label encoding (UPI=1, Cash=3, Credit Card=5, missing=0).
    - **Null handling**: cancellation/incomplete flags converted to 0 where missing.

        - **Numeric features** (avg_vtat, avg_ctat, ride_distance, ratings) preserved.

4. Data Quality Checks

- Added Glue data quality rules to validate schema integrity (non-empty, valid column counts).

5. Output Storage

- Final processed dataset written to S3 in:
    - **Parquet** (for ML pipelines & Athena queries).
    - **CSV** (for inspection & lightweight analytics).

4. Feature Engineering Decisions

Retained (predictors for booking_value)

Temporal features:

- ride_date (could be used for seasonality, weekday/weekend fare variation).
- ride_time (could be used for hourly surge pricing).

Ride characteristics:

- vehicle_type_encoded (vehicle category strongly influences fare).
- ride_distance (primary driver of fare).
- avg_vtat, avg_ctat (time-related metrics linked to cost).

Ride outcome flags:

- cancelled_rides_by_customer, cancelled_rides_by_driver, incomplete_rides → included since cancellations affect revenue (booking value may be zero or reduced).

Service quality:

- driver_ratings, customer_rating (possible influence on fare adjustments or premium services).

Payment method:

- payment_method_encoded (affects revenue distribution and potential fare rounding).

Dropped

- **Pickup and drop location**:
    - High cardinality categorical strings → not immediately useful for ML without geospatial feature engineering.

    - Could be reintroduced later as distance buckets, regions, or clustering features.

- **Raw string versions of date/time/cancellations**:
    - Dropped in favor of cleaned numeric/timestamp forms. Dropped reasons of cancellation as its string and adds no value
    - Also dropped CustomerID and BookingID as it didn't add any value for the target value.

5. Reasoning for Feature Engineering

- **Target-driven preparation**: All features selected for potential correlation with fare amount (booking_value).
- **Numeric encodings**: Ensures ML models (linear regression, tree models, neural nets) can use categorical data.
- **Cancellation handling**: Important since canceled rides often reduce booking value to 0.
- **Distance + vehicle type synergy**: Strong predictors for Uber pricing models.
- **Time-of-day**: Surge pricing & traffic patterns influence booking_value.

6. Final Feature Set (post-ETL)

Feature	**Type**	**Role**
ride_date	date	temporal predictor
ride_time	timestamp	temporal predictor
vehicle_type_encoded	int	categorical predictor
avg_vtat	float	numeric predictor
avg_ctat	float	numeric predictor
cancelled_rides_by_customer	int	target modifier
cancelled_rides_by_driver	int	target modifier
incomplete_rides	int	target modifier
ride_distance	double	key predictor of booking_value
driver_ratings	double	quality indicator
customer_rating	double	quality indicator
payment_method_encoded	int	categorical predictor

Feature	**Type**	**Role**
booking_value	float	**Target Variable**

7. Conclusion

This ETL pipeline delivers a feature-engineered Uber dataset focused on predicting ride booking value (fare amount).
The final dataset is:

- **Cleaned**: nulls replaced, types normalized.
- **Feature-rich**: categorical encodings, temporal features, ride metrics.
- **Target-ready**: explicitly centered on booking_value.
- **Accessible**: stored in **S3 (Parquet + CSV)** for both ML and BI workloads.

This makes the dataset suitable for predictive modeling tasks like:

- **Fare prediction** (based on ride characteristics).
- **Revenue forecasting**.
- **Dynamic pricing models** (impact of time, location, payment method).