Data Science : CSE558

SmartFleet:

Predictive Booking Analysis for Ride Optimization

From Data to Insights: Uncovering Trends and Predictions

GitHub Repo

Dataset Explanation

Our dataset includes taxi ride records with the following features:

- pickup_datetime: Timestamp of when the ride started
- pickup_longitude: Longitude where the ride started
- pickup_latitude: Latitude where the ride started
- dropoff_longitude: Longitude where the ride ended
- dropoff_latitude: Latitude where the ride ended
- passenger_count: Number of passengers
- fare_amount: Cost of the taxi ride

55 M+
ORIGINAL DATASET

44.79 M+
PREPROCESSED DATASET

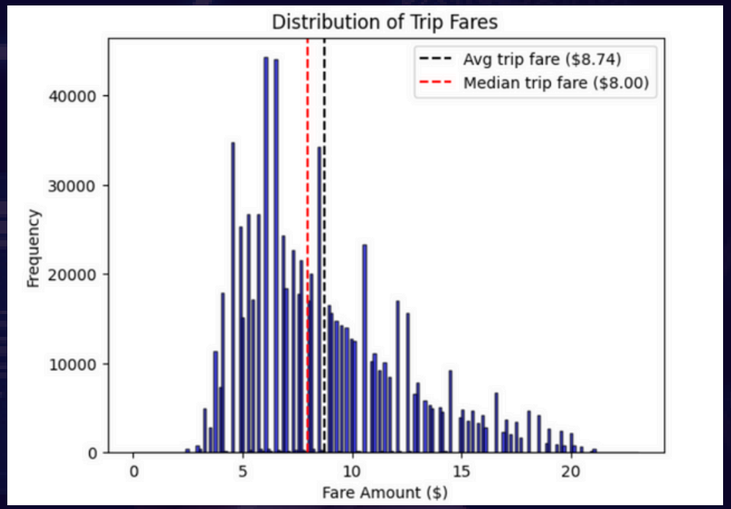
Problem Statement

A leading ride-hailing company is experiencing significant operational inefficiencies, with drivers facing considerable idle time and customers experiencing extended wait times due to suboptimal cab allocation across service regions. The current manual resource allocation system fails to effectively account for dynamic factors such as time-based demand variations and historical booking patterns across different months, impacting both revenue potential and customer satisfaction. The company needs a data-driven solution that can accurately predict demand patterns across different regions using monthly historical booking data and automatically optimize cab allocation to minimize idle time and improve customer satisfaction while maximizing revenue potential. This solution must be scalable across all operational regions, comply with data privacy regulations, and integrate with existing dispatch systems while providing real-time optimization capabilities.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a process in data analysis where analysts examine and summarize datasets to gain insights, understand the structure, and discover patterns, anomalies, or relationships before formal modeling.
On our dataset the following EDA processes were performed -

1. Data Inspection:
   - Basic statistical information like mean and median for numerical columns was calculated.
2. Distribution of Fare Amount:
   - A histogram plotted the distribution of fare_amount with mean and median indicators, aiding in understanding skewness and spread.
3. Date-Time Feature Engineering:
   - The pickup_datetime column was converted to datetime format, and new columns for year, month, day, and hour were created.
   - Counts of pickups were analyzed over time to detect any seasonal or hourly trends.
4. Analysis by Day of the Week:
   - Days of the week were extracted, and passenger_count and fare_amount were aggregated to observe weekday patterns.
5. Scatter Plot of Passenger Count and Fare Amount:
   - Scatter plots were used to visualize potential relationships between the number of passengers and fare amount.

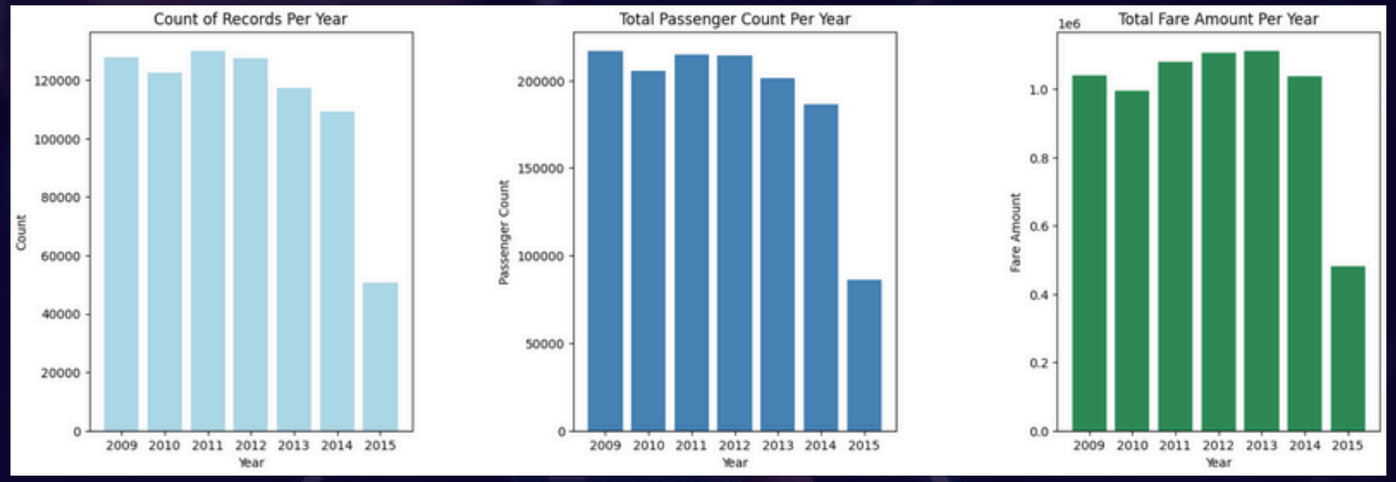


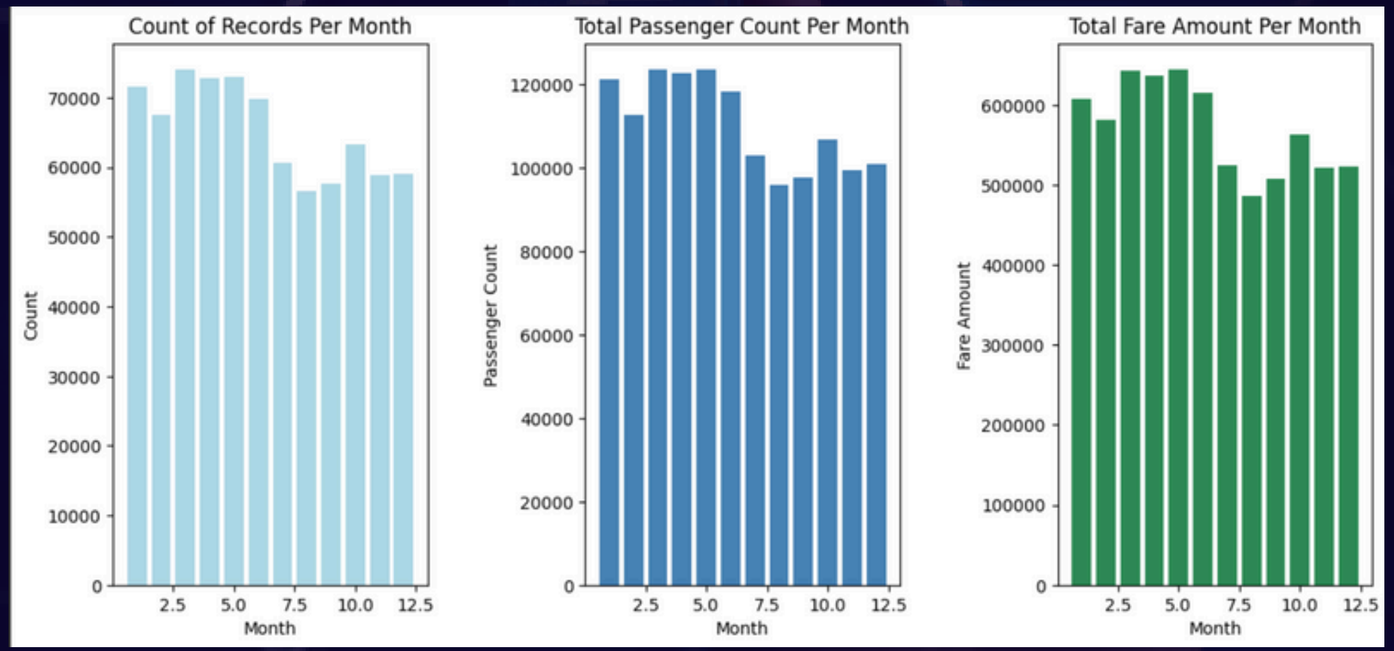
Yearly aggregated data:

	Year	passenger_count	fare_amount
0	2009.0	125120.0	856547.20
1	2010.0	120214.0	824584.57
2	2011.0	125870.0	899651.57
3	2012.0	121858.0	914933.68
4	2013.0	112334.0	915719.98
5	2014.0	105208.0	858376.95
6	2015.0	49278.0	400864.23

1. Geospatial Analysis:
    - Scatter plots were generated for drop-off and pickup locations within city boundaries to understand the geographic distribution of trips.

These analyses help uncover data distributions, patterns, and relationships that are essential for making decisions on data cleaning, transformation, and feature selection for modeling.





SHORT CONCLUSIONS FROM EDA

- Yearly Records: Steady from 2009 to 2014; drop in 2015 indicates fewer trips or incomplete data.
- Passenger Count: Stable over years, similar decline in 2015.
- Total Fare Amount: Peaked around 2012–2013; decreased in 2015.
- Geographical Analysis: Central urban areas were major hotspots for pickups/drop-offs.
- Outliers: High fares/passenger counts suggest rare cases or data issues.
- Temporal Patterns: Peaks during rush hours/weekends; seasonal trends may show holiday effects.
- Correlations: Expected patterns confirmed, like fare increasing with distance.

Preprocessing Tasks

Preprocessing Task	Why?	Effect
Trip Distance Calculation	Created a new column, trip_distance_km, by calculating distance between pickup and dropoff points using the Haversine formula.	Adds an essential feature based on geographic coordinates, crucial for fare prediction. Reduces reliance on raw coordinate values by deriving a more interpretable metric.
Fare per Kilometer Calculation	Added a new column, fare_per_km, by dividing fare_amount by trip_distance_km to calculate fare efficiency for each trip.	Provides insight into cost efficiency across different trip lengths, adding a feature to help the model distinguish costly trips.
Outlier Handling in Fare Amount	Cleaned the dataset by filtering out extreme values in fare_amount based on a reasonable threshold.	Eliminates unusually high or low fares, likely due to errors or rare cases, improving the focus on realistic fare values.
Outlier Handling in Trip Distance	Removed outliers in trip_distance_km by setting a maximum threshold for reasonable travel within NYC (e.g., trips above the 99th percentile).	Removes trips with unrealistic distances, which could distort the model and introduce unnecessary noise.
Outlier Handling in Pickup/Dropoff Locations	Removed trips where pickup or dropoff locations were outside realistic geographic boundaries of NYC.	Ensures trips are within NYC limits, preventing model confusion from outliers located far from the city's core service area.
One-Hot Encoding	Apply one-hot encoding to pickup_dayofweek, pickup_month, and pickup_hour derived from pickup_datetime.	Transforms categorical time-based features into binary format, enabling analysis of patterns by day, month, and hour without implying ordinal relationships.
Feature Extraction from Datetime	Extract pickup_year, pickup_month, pickup_day, pickup_dayofweek, pickup_hour, and is_weekend from pickup_datetime.	Adds temporal context for seasonal and daily ride pattern analysis and creates a is_weekend flag for differentiating weekend and weekday trends.
Resampling (Time-Series)	Aggregates fare_amount to monthly averages, enabling a time-based analysis of fare trends.	Reveals seasonality or trend changes in fare data, useful for forecasting models. Normalizes fluctuations by focusing on long-term patterns instead of daily variances.

SmartFleet

Preprocessing Tasks

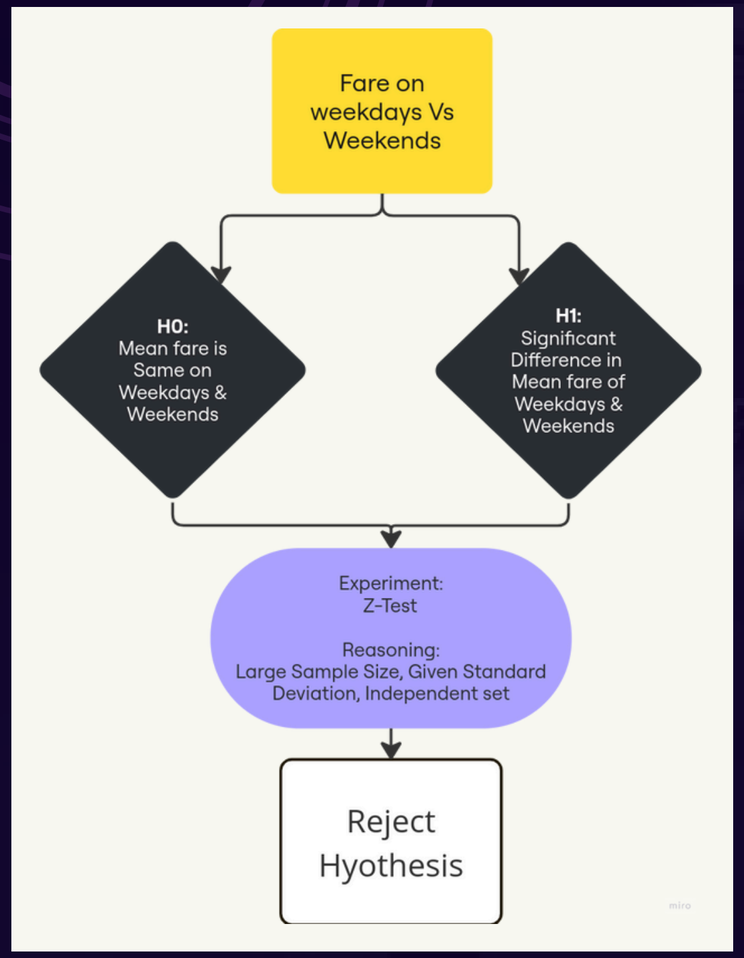
In the final dataset, several columns were updated and new ones were introduced to enrich the data for analysis. The pickup_datetime column was broken down into pickup_year, pickup_month, pickup_day, pickup_dayofweek, and pickup_hour to allow for detailed time-based analysis. The is_weekend column, indicating whether the trip occurred on a weekend, was added to capture potential fare differences on weekends. Additionally, the fare_per_km column was created to reflect fare efficiency by dividing fare_amount by trip_distance_km. To further categorize trips by distance, the distance_band column classifies trips into bands such as 'short', 'medium', 'long' , based on trip_distance_km, which aids in segmenting trip distances for analysis. These updates provide better context for analyzing fare patterns, trip timing, and geographical factors.

Column Name	Description	Data Type	Classification
key	Unique identifier for each trip.	Identifier/String	Discrete
fare_amount	Monetary cost of the trip in USD.	Float/Decimal	Continuous
pickup_datetime	Timestamp of when the trip started, combining date and time.	Datetime	Categorical/Temporal
pickup_longitude	Longitude of the pickup location.	Float/Decimal	Continuous
pickup_latitude	Latitude of the pickup location.	Float/Decimal	Continuous
dropoff_longitude	Longitude of the dropoff location.	Float/Decimal	Continuous
dropoff_latitude	Latitude of the dropoff location.	Float/Decimal	Continuous
passenger_count	Number of passengers in the trip.	Integer	Discrete
trip_distance_km	Distance of the trip in kilometers.	Float/Decimal	Continuous
pickup_year	Year when the trip occurred, extracted from pickup_datetime.	Integer	Discrete
pickup_month	Month when the trip occurred (1-12), extracted from pickup_datetime.	Integer	Discrete/Categorical
pickup_day	Day of the month when the trip occurred (1-31), extracted from pickup_datetime.	Integer	Discrete
pickup_dayofweek	Day of the week when the trip occurred (e.g., Monday, Tuesday), extracted from pickup_datetime.	String/Categorical	Discrete/Categorical
pickup_hour	Hour of the day when the trip started (0-23), extracted from pickup_datetime.	Integer	Discrete/Categorical
is_weekend	Binary indicator if the trip occurred on a weekend (1) or weekday (0).	Binary (0/1)	Discrete/Binary
fare_per_km	Fare charged per kilometer traveled, calculated as fare_amount / trip_distance_km.	Float/Decimal	Continuous
distance_band	Categorical label classifying trip distances into 'short', 'medium', or 'long'.	String/Categorical	Discrete/Categorical

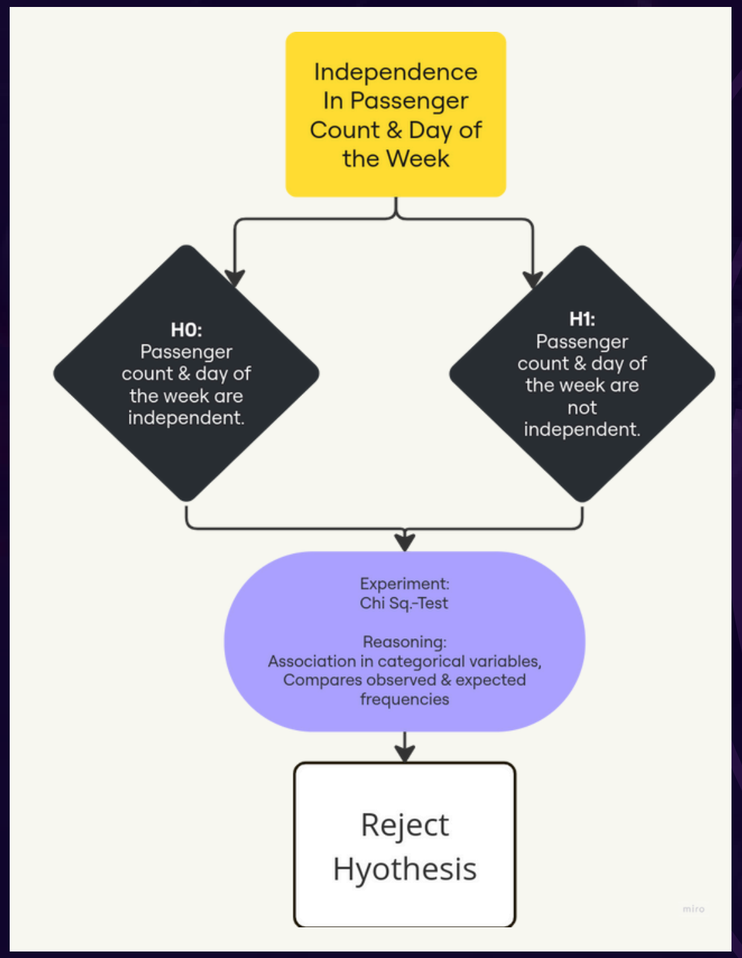
SmartFleet

Hypothesis Tests

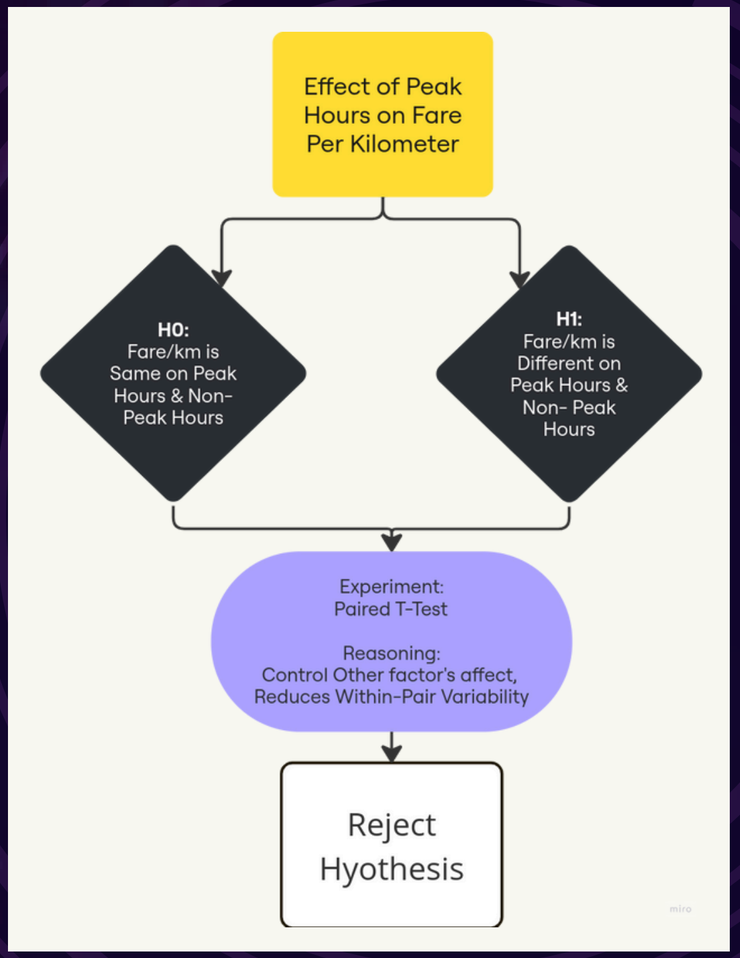
08



- **Pattern of cab booking across the week**
- **Revenue Distribution**



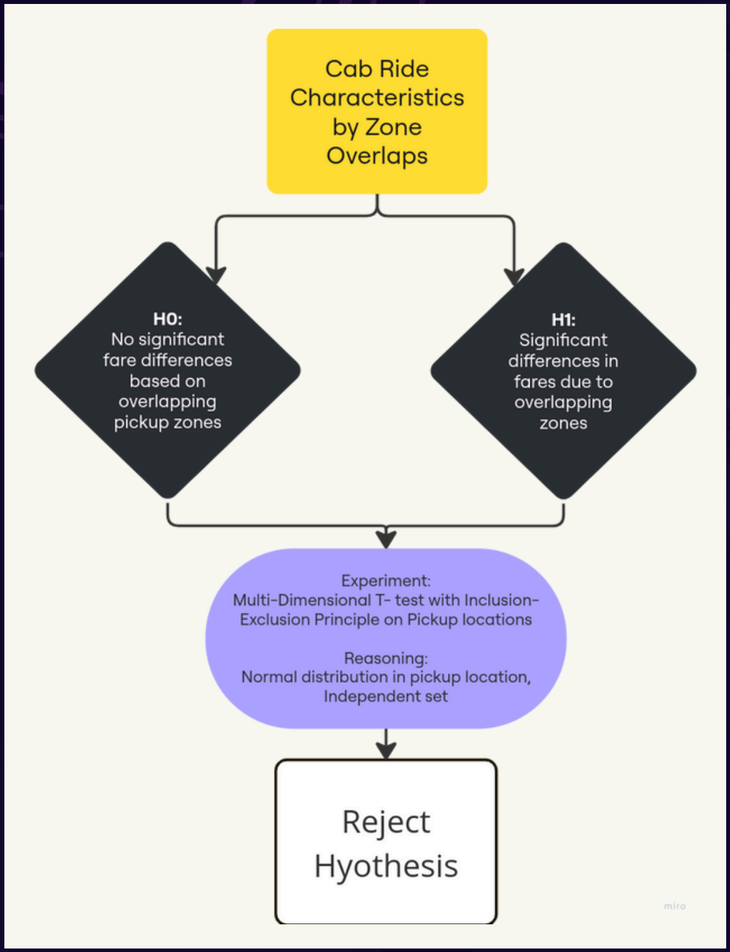
- **The pattern of passengers(customers) with different days**



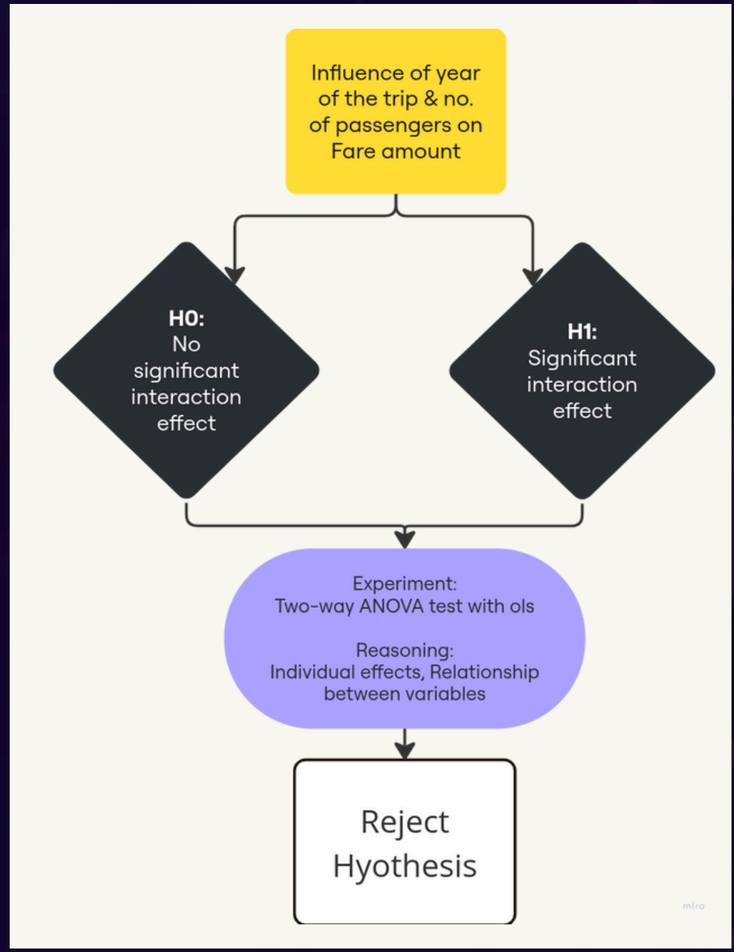
- **Revenue dynamics and demand patterns in a day**

SmartFleet

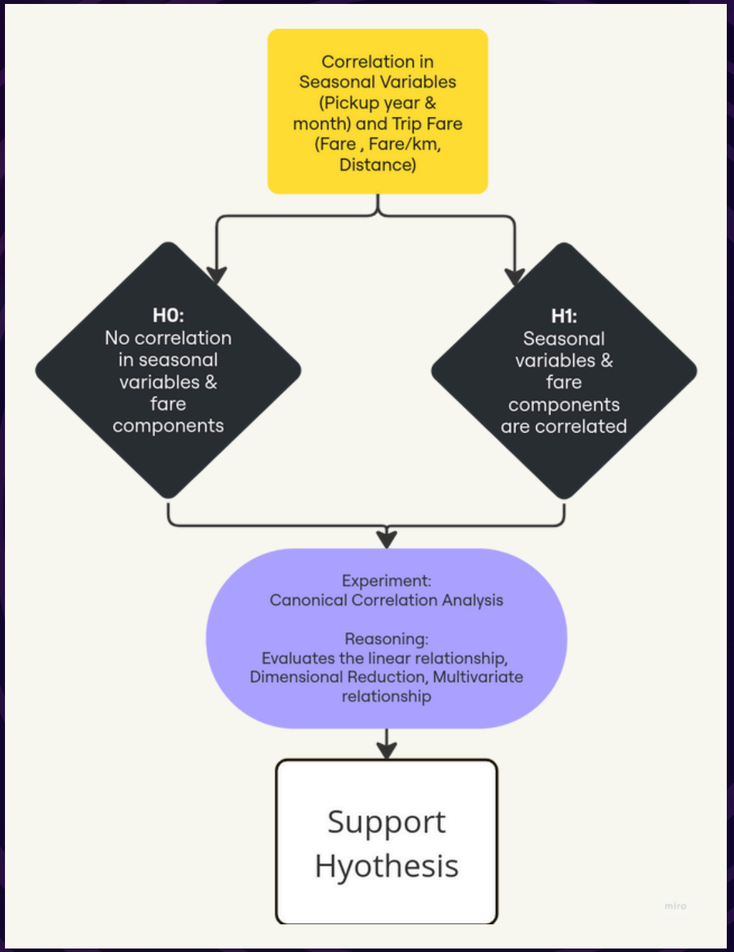
Hypothesis Tests



- **The geographic overlap zones affect fare.**
- **Identification of high-demand regions**



- **Segmentation of Fare Adjustments with year and no of passengers**
- **The affect of financial year on revenue**



- **The seasonal variables determine Predictive Patterns.**

Thank You!

Aditya Bindlish : 2021004
Himani : 2021053
Ankush Gupta : 2021232
Priya Yadav : 2021272