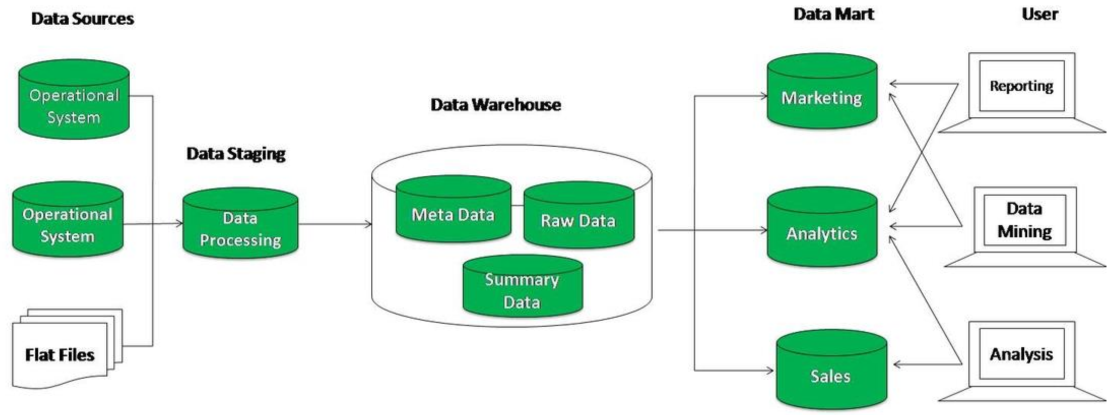
1. Introduction

In today's data-driven landscape, organizations leverage vast amounts of information from various sources to gain insights that support critical decision-making. For ride-sharing companies like Uber and traditional taxi services operating in highly dynamic environments such as New York City, data offers a powerful means to understand customer preferences, optimize operations, and enhance revenue generation strategies. However, managing and analyzing this data can be challenging when it exists in disparate formats across multiple systems.

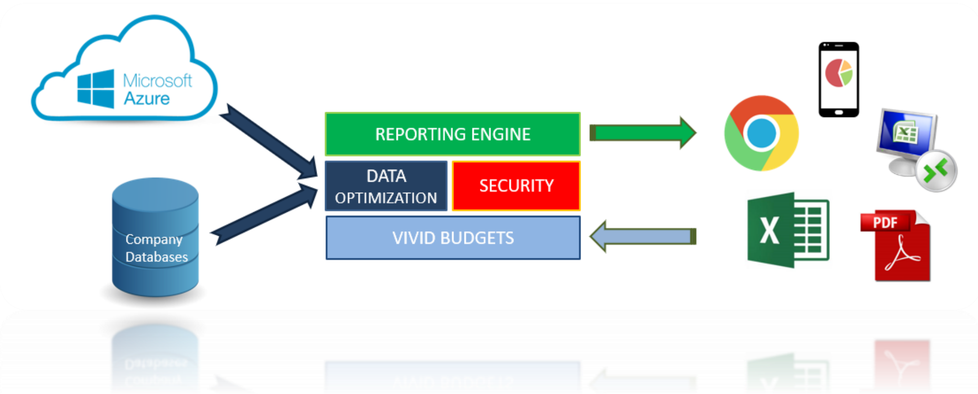
This project aims to address these challenges by implementing a data warehouse—a centralized, consolidated repository that brings together Uber drive data, NYC taxi data, and Uber customer review data. By integrating these sources, the data warehouse provides a unified foundation that facilitates in-depth analysis and reporting. Through this single source of truth, the organization can generate accurate and timely insights into key performance indicators (KPIs) such as ride trends, customer satisfaction, and revenue. This project's scope includes not only data integration but also the creation of an ETL (Extract, Transform, Load) pipeline that automates data collection, cleaning, transformation, and loading into the warehouse.



2. Project Objectives

The objectives of this data warehouse implementation are as follows:

- **Centralized Data Storage**: Gather data from Uber, NYC Taxi, and Uber Reviews into a single, consolidated data repository



- **Enhanced Reporting**: Organize data to facilitate quick and accurate report generation, enabling the analysis of ride trends, customer experience, and revenue.

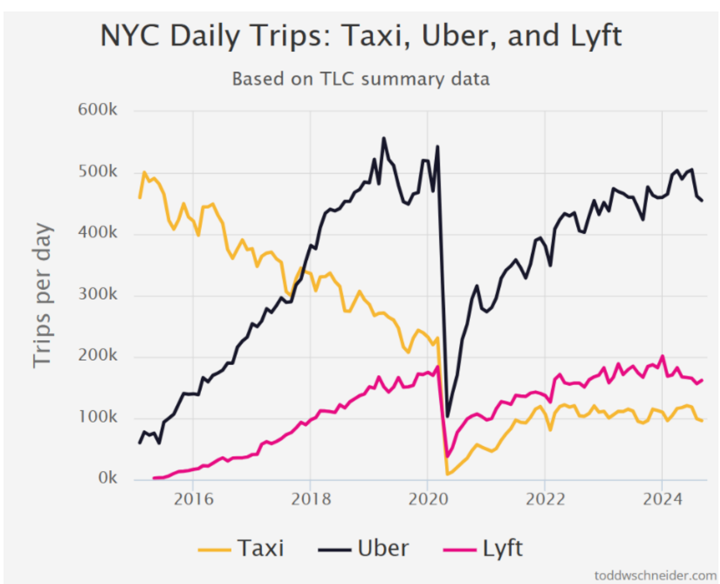
- **Data Accuracy**: Ensure the integration of clean, consistent data across all sources to support reliable business insights.
- **Business Insights and Decision-Making**: Utilize tools such as Power BI or Tableau to provide visual analytics, assisting stakeholders in data-driven decision-making.

3. Data Sources

The following data sources were integrated:

- **Uber Drive Data**: Contains detailed information about each Uber ride, including start and end times, ride category, distance, and purpose.

- **NYC Taxi Data**: Covers essential taxi ride information such as pickup and drop-off times, trip distance, fare amount, and pickup/drop-off location IDs.



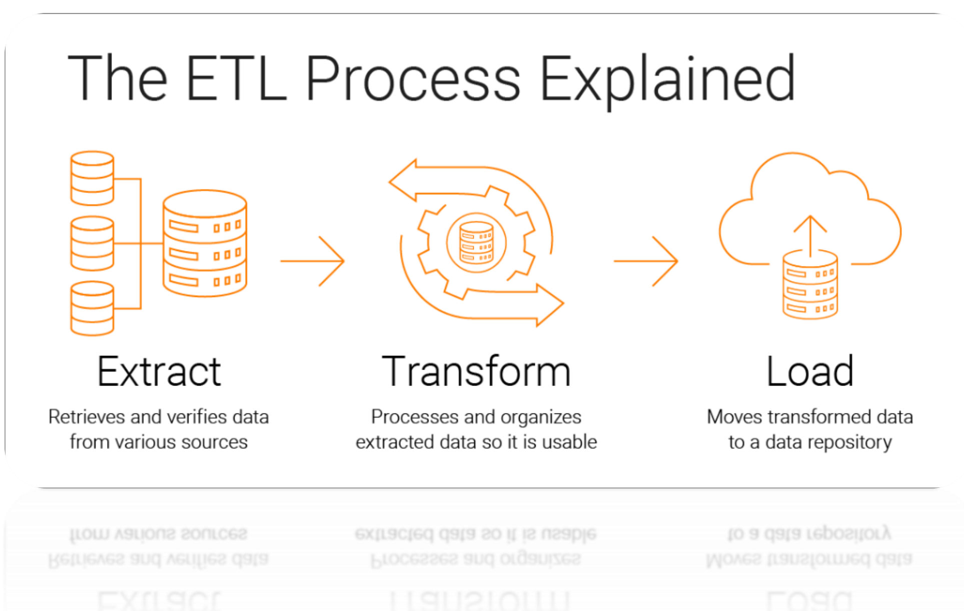
- **Uber Review Data**: Comprises customer feedback, including ratings and comments,

	START_DA	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	payment_t	fare_amou	tip_amour	total_amo	passenger_	Rating	Comment	TIME
2	########	########	business	Fort Pierce	Fort Pierce	5.1	meal/ente	2	56.2	0	61.95	2	1	I had an ac	Night
3	########	########	business	Fort Pierce	Fort Pierce	5	no purpose	2	12.8	0	19.3	1	1	I have had	Night
4	########	########	business	Fort Pierce	Fort Pierce	4.8	errand/sup	1	7.9	0	14.4	1	1	I requeste	Evening
5	########	########	business	Fort Pierce	Fort Pierce	4.7	meeting	1	20.5	5.4	32.4	1	1	I've been c	Evening
6	########	########	business	Fort Pierce	West Palm	63.7	customer v	1	16.3	5.06	32.11	1	1	Uber is ov	Afternoon
7	########	########	business	West Palm	West Palm	4.3	meal/ente	1	7.2	1.3	15	1	1	I had an ai	Evening
8	########	########	business	West Palm	Palm Beac	7.1	meeting	1	15.6	4.42	26.52	1	1	I worked f	Evening
9	########	########	business	Cary	Cary	0.8	meeting	2	5.1	0	11.6	1	1	In July of t	Afternoon
10	########	########	business	Cary	Morrisville	8.3	meeting	1	34.5	13.23	67.92	1	1	My driver,	Morning
11	########	########	business	Jamaica	New York	16.5	customer v	2	19.1	0	24.85	1	1	I had sever	Afternoon
12	########	########	business	New York	Queens	10.8	meeting	1	9.3	3.16	18.96	1	1	Our driver	Afternoon
13	########	########	business	Elmhurst	New York	7.5	meeting	2	13.5	0	20	2	1	When the	Evening
14	########	########	business	Midtown	East Harle	6.2	meeting	1	18.5	0	19	1	1	Being a res	Evening
15	########	########	business	East Harle	NoMad	6.4	temporary	1	22.6	0.02	29.12	1	1	They didn'	Morning
16	########	########	business	Flatiron Di	Midtown	1.6	errand/sup	1	41.5	11.99	71.93	2	1	I have bee	Morning
17	########	########	business	Midtown	Midtown E	1.7	meal/ente	3	-67.1	0	-70.6	1	1	Don't use i	Afternoon
18	########	########	business	Midtown E	Midtown	1.9	meal/ente	3	67.1	0	70.6	1	1	Upfront pr	Afternoon
19	########	########	business	Midtown	Hudson Sq	1.9	meal/ente	1	14.9	0.01	21.41	2	1	I don't beli	Afternoon
20	########	########	business	Hudson Sq	Lower Ma	4	meal/ente	1	13.5	1	21	1	1	Drive with	Afternoon
21	########	########	business	Lower Ma	Hudson Sq	1.8	errand/sup	1	86.94	17.59	105.53	2	1	Uber quot	Afternoon
22	########	########	business	Hudson Sq	Hell's Kitch	2.4	customer v	1	25.4	5.88	35.28	1	1	Driver was	Afternoon
23	########	########	business	Hell's Kitch	Midtown	2	errand/sup	1	47.8	10.36	63.91	3	1	I called ub	Afternoon
24	########	########	business	New York	Queens Co	15.1	meeting	1	70	0	85.94	2	1	This is the	Afternoon
25	########	########	business	Downtown	Gulfton	11.2	meeting	4	-25.1	0	-30.35	1	1	uber app like to steal your money then claim they refunded you this why I prefer Lyft	
26	uber app li	Afternoon													
27	########	########	business	Gulfton	Downtown	11.8	meeting	4	25.1	0	30.35	1	1	Cherry picl	Afternoon
28	########	########	business	Houston	Houston	21.9	customer v	1	35.9	7.98	47.88	1	1	Uber in the	Afternoon
29	########	########	business	Eagan Park	Jamestown	3.9	errand/sup	1	12.1	4.65	23.25	2	1	I ordered r	Night
30	########	########	business	Morrisville	Cary	8	errand/sup	1	7.9	4.32	18.72	1	1	Uber drive	Night

combined_data

4. ETL Process

The ETL (Extract, Transform, Load) process was executed in several stages to ensure clean, organized data:



Step 1: Data Extraction

Data was extracted from the three sources using Python's panda library, reading the datasets into Data Frames for further processing. This initial extraction laid the foundation for cleaning, transforming, and ultimately merging the data into a unified structure.

```
#collected from kaggle datasets
import pandas as pd
first_source_df = pd.read_csv('sample_data/Uber Drives.csv')
first_source_df.head()

#collected from kaggle datasets
import pandas as pd
first_source_df = pd.read_csv('sample_data/Uber Drives.csv')
first_source_df.head()


#scraped from consumer affairs website
third_source_df= pd.read_csv('sample_data/Uber Review Data.csv')
third_source_df.head()
```

Step 2: Data Transformation

- **Remove Asterisks from Column Names**: To standardize column naming conventions, all asterisks were removed.

```python
# Remove asterisks (*) from column names
first_source_df.columns = first_source_df.columns.str.replace('*', '', regex=False)
```

- **Column Selection**: Only relevant columns were retained for each dataset, such as ride times, distances, categories, fare amounts, ratings, and comments.

```python
first_source_df = first_source_df[['START_DATE', 'END_DATE', 'CATEGORY', 'START', 'STOP', 'MILES',
'PURPOSE']]
second_source_df = second_source_df[['tpep_pickup_datetime', 'tpep_dropoff_datetime', 'trip_distance',
'PULocationID', 'DOLocationID', 'payment_type', 'fare_amount', 'tip_amount', 'total_amount',
'passenger_count']]
third_source_df = third_source_df[['Stars', 'Comment']]
```

- **Column Renaming for Consistency**: Columns from different datasets were renamed to align data fields. For instance, tpep_pickup_datetime and tpep_dropoff_datetime in NYC Taxi data were renamed to START_DATE and END_DATE, respectively.

```python
# Step 2: Rename Columns for Consistency
second_source_df.rename(columns={
    'tpep_pickup_datetime': 'START_DATE',
    'tpep_dropoff_datetime': 'END_DATE',
    'trip_distance': 'MILES',
    'PULocationID': 'START',
    'DOLocationID': 'STOP',
    'fare_amount': 'fare_amount',
    'tip_amount': 'tip_amount',
    'total_amount': 'total_amount',
    'passenger_count': 'passenger_count',
    'payment_Type': 'payment_type',
}, inplace=True)

third_source_df.rename(columns={
    'Stars': 'Rating',
    'Comment': 'Comment',
}, inplace=True)
```

- **Data Type Standardization**: Date columns were converted to date time format, and mileage columns were ensured to be numeric across datasets.

```python
# Step 3: Transform Data Formats

# Convert date columns to datetime format
first_source_df['START_DATE'] = pd.to_datetime(first_source_df['START_DATE'], format="%m/%d/%Y %H:%M",
errors='coerce')
first_source_df['END_DATE'] = pd.to_datetime(first_source_df['END_DATE'], format="%m/%d/%Y %H:%M",
errors='coerce')
second_source_df['START_DATE'] = pd.to_datetime(second_source_df['START_DATE'], errors='coerce')
second_source_df['END_DATE'] = pd.to_datetime(second_source_df['END_DATE'], errors='coerce')

# Ensure MILES is numeric in both datasets
first_source_df['MILES'] = pd.to_numeric(first_source_df['MILES'], errors='coerce')
second_source_df['MILES'] = pd.to_numeric(second_source_df['MILES'], errors='coerce')
```

- **Duplicate Removal**: Duplicate entries were identified and removed to prevent redundancy in the analysis.

```python
# Step 4: Handle Duplicates and Missing Values

# Remove duplicate records in  datasets
first_source_df.drop_duplicates(inplace=True)
second_source_df.drop_duplicates(inplace=True)
```

- **Handling Missing Values**: Critical columns with missing values, such as START_DATE, END_DATE, and MILES, were filtered out. Missing values in fields like PURPOSE were filled with 'No Purpose.'

```python
# Drop rows with missing values in critical fields (you can adjust this based on your needs)
first_source_df.dropna(subset=['START_DATE', 'END_DATE', 'START', 'STOP', 'MILES'], inplace=True)
second_source_df.dropna(subset=['START_DATE', 'END_DATE', 'START', 'STOP', 'MILES'], inplace=True)
third_source_df.dropna(subset=['Rating', 'Comment'], inplace=True)

# Fill missing values in PURPOSE with 'No Purpose'
first_source_df['PURPOSE'] = first_source_df['PURPOSE'].fillna('No Purpose')

# Optional: Standardize categorical fields for consistency
first_source_df['CATEGORY'] = first_source_df['CATEGORY'].str.lower()
first_source_df['PURPOSE'] = first_source_df['PURPOSE'].str.lower()
```

Step 3: Data Integration And Load:

After cleaning and transforming each dataset:

- **Data Merging**: The Uber, NYC Taxi, and Uber Review datasets were merged into a single dataset. Key fields from each source were aligned, allowing the consolidated dataset to represent a complete view of each ride, including ride details, fare information, and customer feedback.

```
# Step 5: Combine Datasets

first_source_df[['payment_type', 'fare_amount', 'tip_amount', 'total_amount', 'passenger_count']] =
second_source_df[['payment_type', 'fare_amount', 'tip_amount', 'total_amount', 'passenger_count']]
first_source_df[['Rating','Comment']] = third_source_df[['Rating','Comment']]

first_source_df['TIME'] = first_source_df['START_DATE'].dt.time
first_source_df['START_DATE'] = first_source_df['START_DATE'].dt.date
first_source_df['END_DATE'] = first_source_df['END_DATE'].dt.date
```

- **Time of Day Categorization**: A new column, TIME, was added to categorize rides into Morning, Afternoon, Evening, or Night, based on the start time. This enrichment provides insights into the most active times for ride services.
- **LOAD**: After all phases the data was successfully loaded into the PostgreSQL Database.

```python
# Step 6: Enrich Data

# a function to categorize the time of day
def categorize_time(time):
    hour = time.hour
    if 5 <= hour < 12:
        return 'Morning'
    elif 12 <= hour < 17:
        return 'Afternoon'
    elif 17 <= hour < 21:
        return 'Evening'
    else:
        return 'Night'

first_source_df['TIME'] = first_source_df['TIME'].apply(categorize_time)
first_source_df['TIME'] = first_source_df['TIME'].dropna()

combined_df = first_source_df

# Preview the final combined dataset with the new fie
combined_df.to_csv('sample_data/combined_data.csv', index=False)
# combined_df.head(100)
```

```python
import pandas as pd
from sqlalchemy import create_engine

# Sample connection string; replace with your actual PostgreSQL credentials
DATABASE_TYPE = 'postgresql'
DBAPI = 'psycopg2'
ENDPOINT = 'localhost'
USER = 'postgres'
PASSWORD = 'pakistan'
PORT = 5432                              # Default PostgreSQL port is 5432
DATABASE = 'UBER DATA WAREHOUSE'

# Create a connection string and engine
connection_string = f"{DATABASE_TYPE}+{DBAPI}://{USER}:{PASSWORD}@{ENDPOINT}:{PORT}/{DATABASE}"
engine = create_engine(connection_string)

# Load the DataFrame into PostgreSQL
table_name = 'UBER_DATA'  # Choose your table name
combined_df.to_sql(table_name, engine, if_exists='replace', index=False)

print(f"Data successfully loaded into table '{table_name}' in the '{DATABASE}' database.")
```

5. Challenges in Data Integration

During data integration, several challenges were identified:

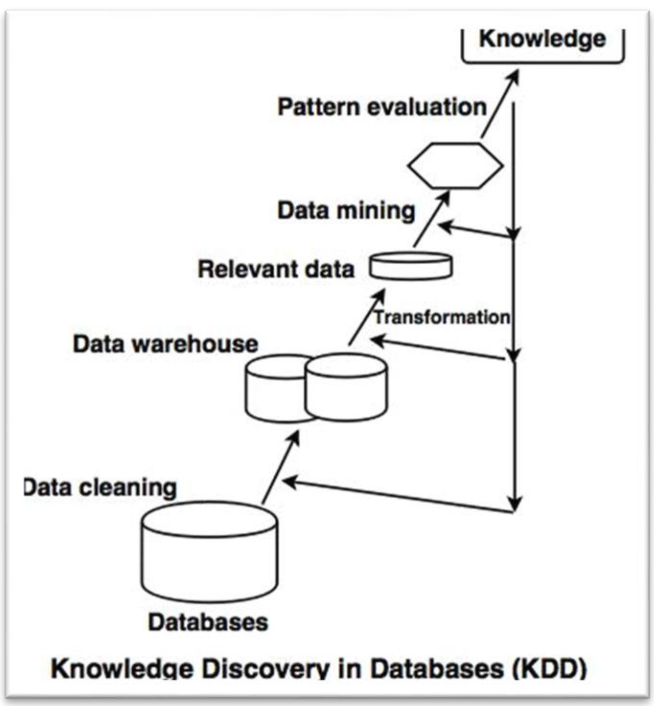
- **Data Inconsistencies**: Variations in data formats across datasets led to challenges in aligning timestamps, location identifiers, and category names.
- **Duplication**: With data from multiple sources, duplicate entries posed a risk. Ensuring that each record represented a unique event required extensive filtering and validation.
- **Missing Data**: Some datasets, such as Uber Drive and NYC Taxi, had missing fields like ride purpose and fare amounts. Techniques such as imputation were explored to handle these gaps without skewing analysis.

- **Data Synchronization**: Coordinating time-based data from multiple sources required a consistent time zone and format standardization to align rides, feedback, and revenue records accurately.

6. Application of Data Mining Techniques

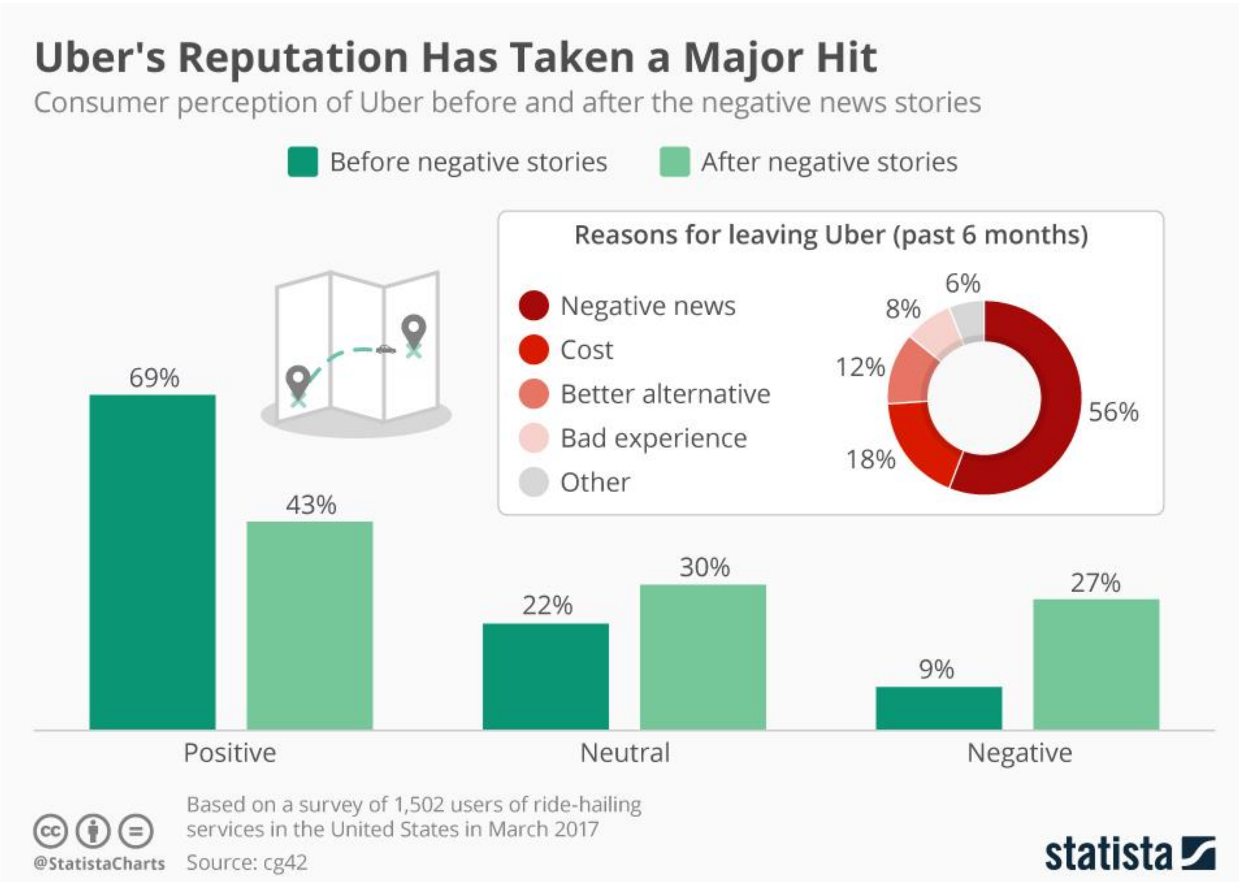
After loading the data, advanced data mining and machine learning algorithms were applied to uncover insights that could support business decisions:

- **Clustering for Customer Segmentation**: Clustering algorithms were employed to segment customers based on ride frequency, type, and spending behavior. This



segmentation helps in identifying high-value customers, which can be useful for personalized marketing campaigns.

- **Sentiment Analysis for Customer Experience**: Using classification algorithms on Uber review data allowed the categorization of comments into positive, neutral, and negative sentiments. This analysis provides insights into customer satisfaction and identifies recurring service-related issues.



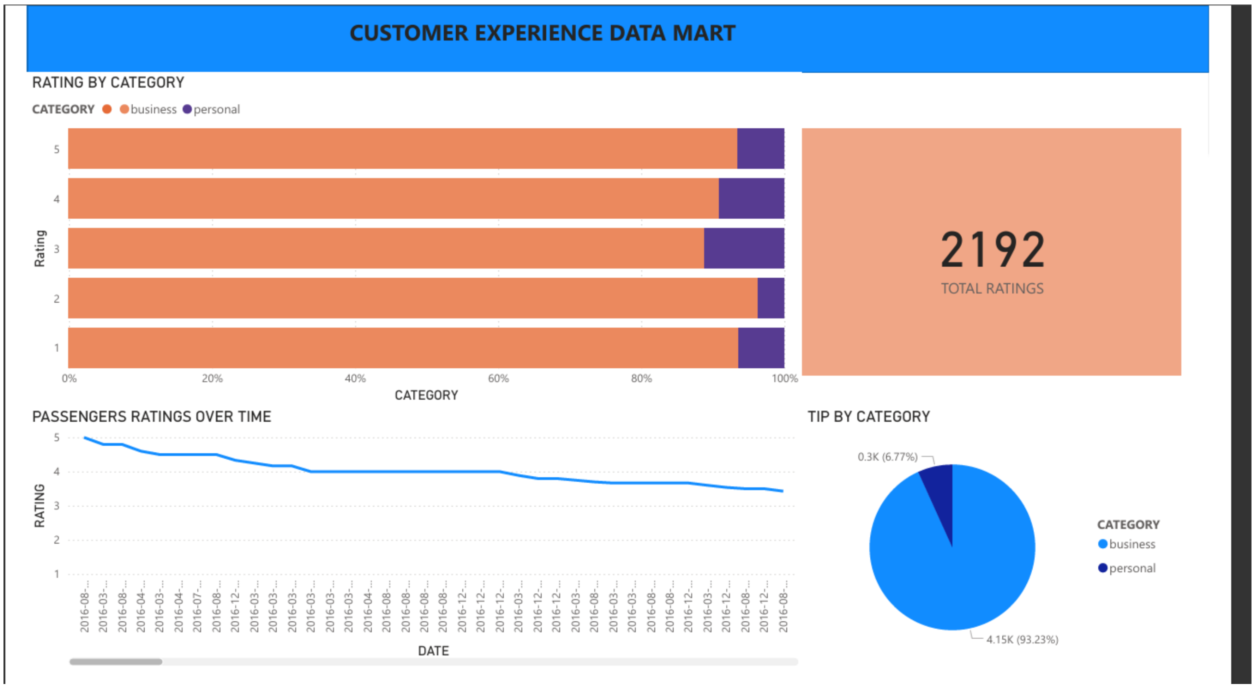
- **Revenue Prediction and Trend Analysis**: Regression models were used to predict revenue based on ride categories, distance, and other attributes. By forecasting revenue patterns, the organization can plan operational resources and marketing strategies more effectively.

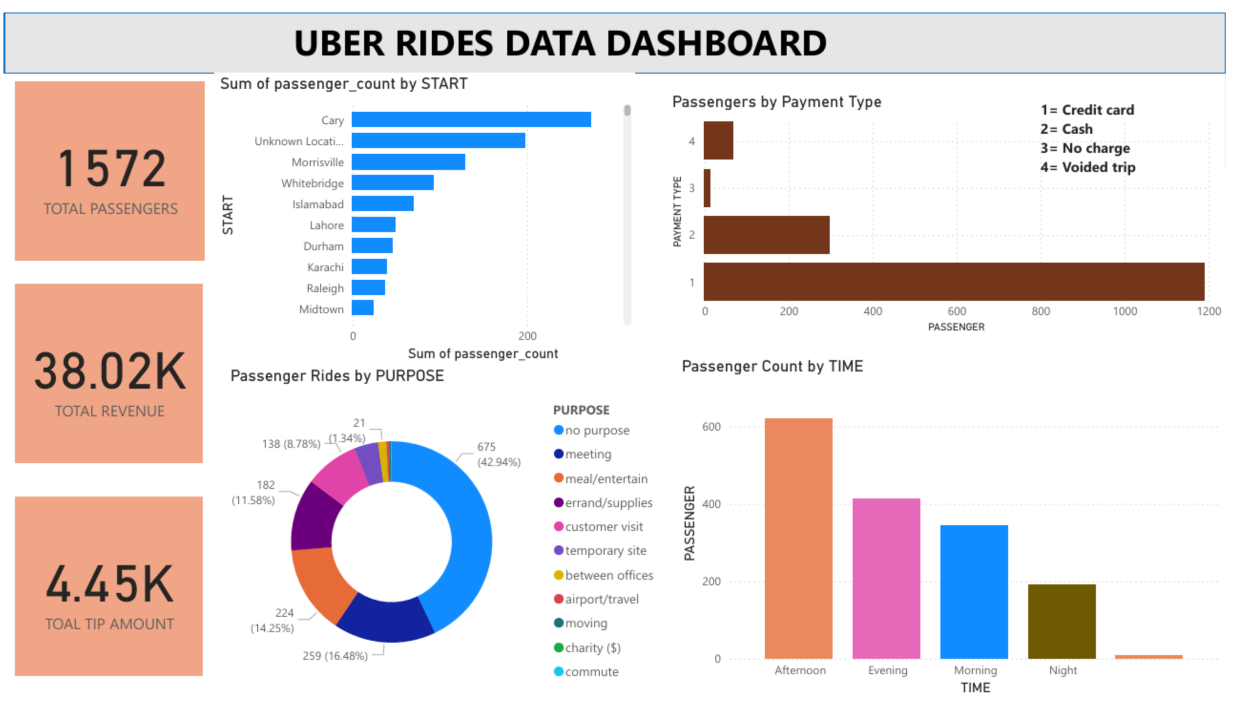
7. Key Insights for Enhanced Reporting and Decision-Making

The data mining insights significantly enhance the organization’s ability to make data-driven decisions:

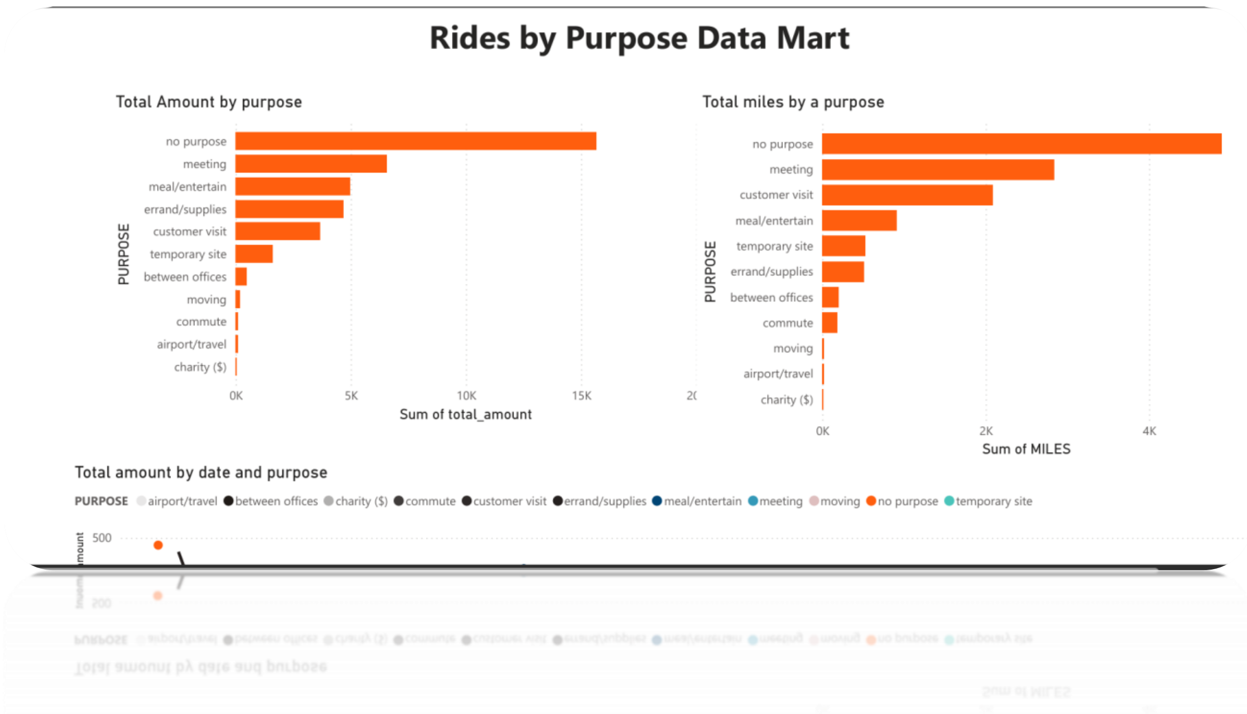
- **Ride Trends**: Detailed analysis on ride categories and purposes allows for an understanding of popular services and peak usage times, aiding in resource allocation and promotional campaigns.
- **Customer Experience Insights**: Sentiment analysis of reviews highlights areas for improvement in service quality, allowing for targeted actions to enhance customer satisfaction.
- **Revenue Insights**: Revenue-related insights, such as average revenue per mile and tipping behavior, support financial planning and pricing strategy adjustments.

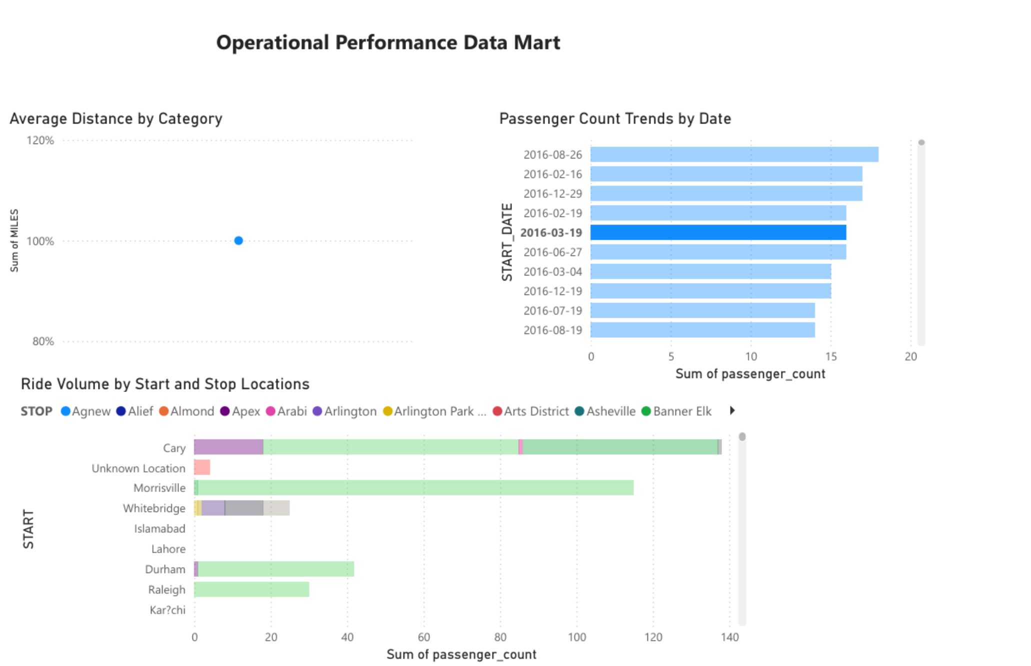
POWER BI DASBOARD:





DATA MARTS:


Rides by Purpose Data Mart
Total Amount by purpose
no purpose
meeting
meal/entertain
errand/supplies
customer visit
temporary site
between offices
moving
commute
airport/travel
charity ($)
PURPOSE
0K
5K
10K
15K
Sum of total_amount
Total miles by a purpose
no purpose
meeting
customer visit
meal/entertain
temporary site
errand/supplies
between offices
commute
moving
airport/travel
charity ($)
PURPOSE
0K
2K
4K
Sum of MILES
Total amount by date and purpose
PURPOSE airport/travel between offices charity ($) commute customer visit errand/supplies meal/entertain meeting moving no purpose temporary site
500


Operational Performance Data Mart
Average Distance by Category
120%
100%
80%
Sum of MILES
Passenger Count Trends by Date
2016-08-26
2016-02-16
2016-12-29
2016-02-19
2016-03-19
2016-06-27
2016-03-04
2016-12-19
2016-07-19
2016-08-19
START_DATE
0
5
10
15
20
Sum of passenger_count
Ride Volume by Start and Stop Locations
STOP Agnew Alief Almond Apex Arabi Arlington Arlington Park ... Arts District Asheville Banner Elk
Cary
Unknown Location
Morrisville
Whitebridge
Islamabad
Lahore
Durham
Raleigh
Kar?chi
START
0
20
40
60
80
100
120
140
Sum of passenger_count

8. Conclusion

This data warehouse project successfully integrated data from Uber, NYC Taxi, and Uber Reviews into a consolidated data warehouse, providing a foundation for advanced reporting and analytics. By overcoming data integration challenges and applying data mining techniques, the project delivers valuable insights that empower the organization to improve operational efficiency, optimize customer experience, and make informed business decisions.