Milestone 3: Design Document

1. Abstract

In the ever-evolving landscape of urban transportation, ride-sharing services have carved a significant niche. However, one of the persistent challenges faced by drivers is the unpredictability of passenger requests, leading to prolonged idle times. With increasing competition and dynamic demand patterns influenced by various factors like time of day, events, or even weather, there's a pressing need to assist drivers in making informed decisions on where to wait.

This design document outlines the Ubermate project with the primary objective of optimizing ride-sharing driver positioning during periods of low demand, focusing on drivers working for Uber. By harnessing data analytics and machine learning techniques we aim to identify the most advantageous locations where drivers can strategically position themselves to increase their likelihood of receiving passenger requests promptly and thus improve the drivers' earnings on the platform. The project's scope encompasses data collection, data ingestion, preprocessing, computing phase, analysis, modeling, and the development of a recommendation system to guide drivers to these strategic hotspots.

2. Project Design

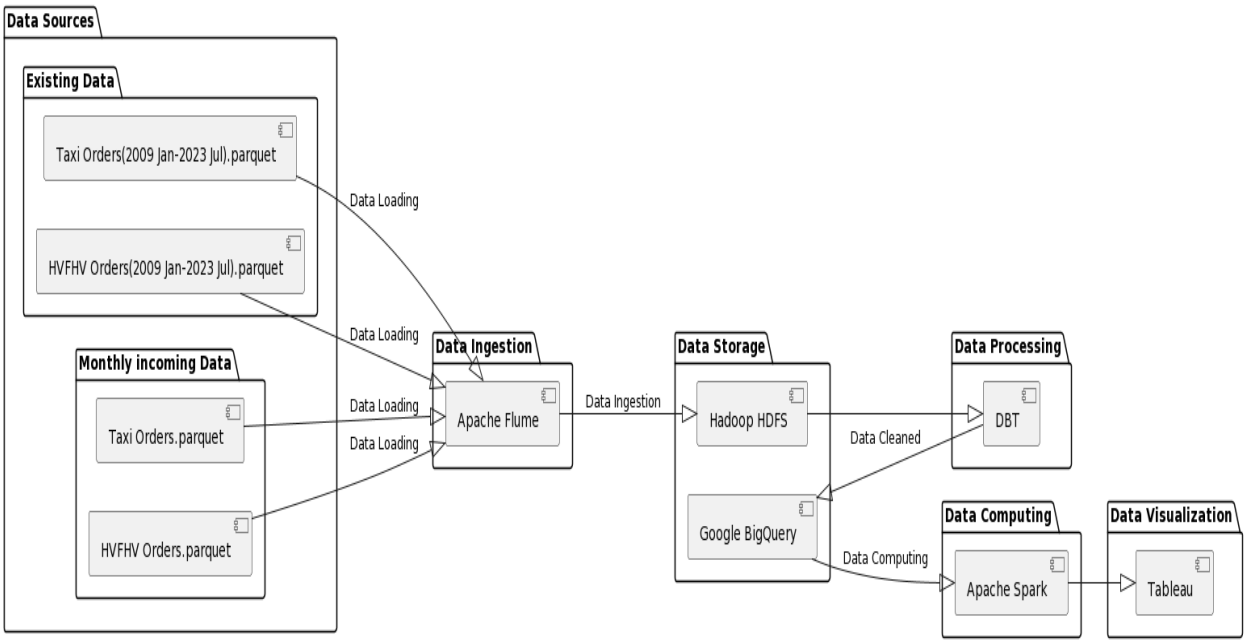

Figure. 1 Pipeline Diagram

2.1 Data Collection and and ingestion

The data source for this project is from NYC Taxi & Limousine Commission official website. The timespan of this data source covers January 2009 to July 2023. Yellow taxi records of NYC are available from January 2009 to July 2023. Green taxi records of NYC are available from August 2013 to July 2023. For-Hire Vehicle Records are available from January 2015 to July 2023. High Volume For-Hire Vehicle Records, which can be generalized as Uber/Lyft are available from February 2019 to July 2023. All of these dataset are in Parquet format with about 15 columns that contain attributes like pickup_datetime, trip_distance, pickup_location and dropoff_location. Although

the general schema and format stays similar, relatively old data in our source only contains longitude and latitude for the pickup/dropoff location, but the relatively new dataset is recording the pickup/dropoff location using LocationID without specific longitude and latitude information. The data source divided NYC into a lot of areas and each area is represented by a LocationID. Also, an additional data source we have is a smaller collection of Uber orders in New York City in csv format. The timespan ranges from 2009 to 2015. The dataset only has 200,000 rows. We are also introducing this dataset into our project because we want the pipeline to be able to understand data sources in different file formats and schemas.

During data ingestion, there are going to be two distinct pathways that arise depending on the file type. Whether it's a CSV file or a Parquet file, the data first must go through Apache Flume. Flume allows for quick and efficient acceptance of streaming data. Flume takes this data and sends it to Hadoop HDFS, a distributed files system. HDFS uses MapReduce to store large amounts of data, which involves partitioning the data into chunks. Parquet is the ideal data format since it is in a columnar format, allowing for quicker queries, and we only need 5 of the 15 columns. DBT will be used to clean and transform our data, including removing unnecessary dimensions in the case for the Parquet files. For both, we will also use DBT to remove entities with nulls. The CSV and Parquet files differ in how they present their Pickup and Drop off locations. For CSV, they use a latitude and longitudinal format. For Parquet, each location is associated with a location ID. The Parquet files are unique in that they also have an associated CSV and PNG file accompanied with it. The CSV houses a legend that associates the location ID number with a location name. The PNG file contained a segmented image of the New York Map. WIth this knowledge, we also plan to convert the locations fields to match that of the CSV files. This concludes the data cleaning process as the data will be sent to BigQuery, a data warehousing toolkit.

2.2 Computing, Analysis and Visualization

A-Segmentation

Segmentation is a crucial step in our data analytics process to identify optimal locations for ride-sharing drivers. Each data frame consists of ride-sharing data with timestamps, latitude, and longitude. Segmentation will be carried out to group data points into clusters using the K-means algorithm. The segmentation process works by examining time and geographical features. By considering the hour of the day and the day of the week in conjunction with driver location data, we can categorize ride requests into different time periods and days. We will then apply the K-means clustering algorithm to each segment of the datasets. This segmentation approach will allow us to pinpoint optimal driver locations for specific time and date scenarios, enabling drivers to enhance their earnings and improve service efficiency during periods of low demand.

B - Clustering algorithms

Although there are many clustering algorithms, we decided to implement the K-means algorithm for this project. In fact, it plays a crucial role in optimizing the positioning of Uber drivers, during periods of low demand. K-Means offers an effective means of identifying and categorizing spatial clusters of ride request locations, thereby revealing patterns in passenger demand. By partitioning the geographical area into 'k' clusters, where 'k' represents the desired number of optimal locations for drivers, K-Means enables data analytics to determine where these strategic spots should be. The algorithm operates based on the proximity of locations with high ride request density, allowing it to pinpoint areas where drivers are most likely to receive passenger requests. This not only enhances the drivers' earnings but also optimizes the overall service efficiency of the ride-sharing platform, leading to higher passenger satisfaction.

The selection of the parameter "k" in the K-Means algorithm is also important, as it aims to evenly distribute data points into clusters of roughly equal sizes. To determine the optimal value for "k" in this context, we will employ the elbow method for each set of data frames. The following flowchart illustrates the working flow of the k-means algorithm.

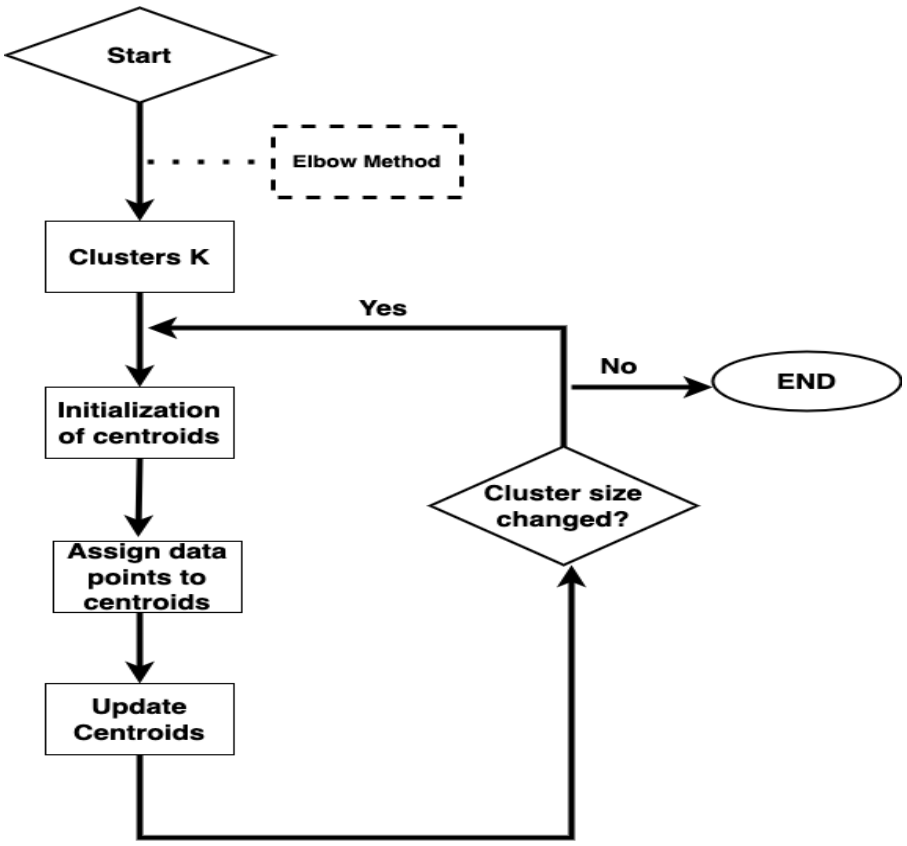

Figure2. Flowchart of K-means algorithm

Once the data are distributed into clusters, the page rank algorithm will be used for the remaining part of the project.

C - Page Rank

Page Rank is an algorithm by Google that is used to rank web page searches. We will be using it to rank hotspots offered by the clustered data points. Beforehand, we must develop a graph for it to operate on. Our graph will be weighted and directional. Our plan is to separate the map into a grid. Each grid, with data points, will be represented by a node. Any points within a node will be used in the weight calculation. If a pickup location and dropoff location are in different nodes than an edge is formed. Since the locations use latitude and longitude values, we are going to find the Euclidean distance. The weight calculation is also gonna have the average cost of all routes between the two nodes. In addition, The number of rides has a positive correlation to weight, while distance has a negative relationship to it. After this graph is created, the Page Rank will rank the nodes and will provide an accompanying score for each. The score will represent the recommendation to drivers.

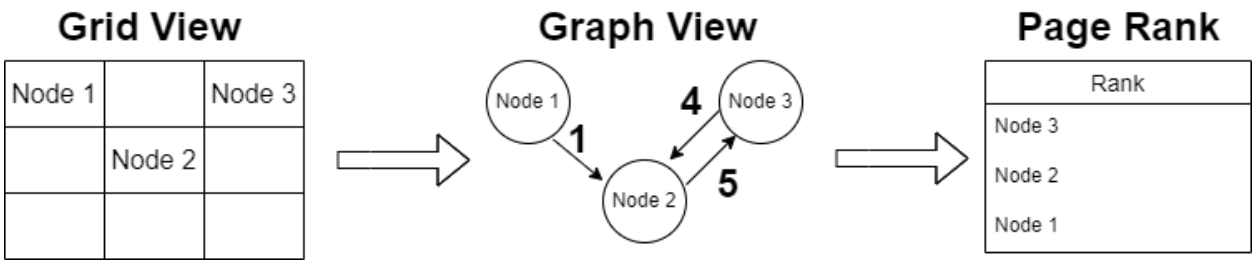

Figure 3. Flowchart for Page Rank

D - Visualization

The visualization process starts with importing data into Tableau. This involves connecting Tableau to the data source and thoroughly reviewing the imported data to ascertain its format and alignment. Following data import, the visualization phase encompasses various elements:

1. Geospatial Visualization: This leverages the latitude and longitude data to generate map-based visuals. Nodes, based on their Page Rank scores or ranks, can be represented as differently-sized points on this map.
2. Node Ranking: To comprehend the rank of nodes based on the Page Rank scores, bar charts or analogous visualizations are instrumental.
3. Edge Visualization: Representing connections between nodes in Tableau can be intricate. However, path maps can be employed to showcase edges, with line thickness indicating edge weight.

There are also several techniques that can enhance the depth of the visualization. These encompass using distinct colors to spotlight the highest-scoring nodes, integrating filters for viewers to zone in on particular nodes or edges, and crafting tooltips for a more detailed view on hover.

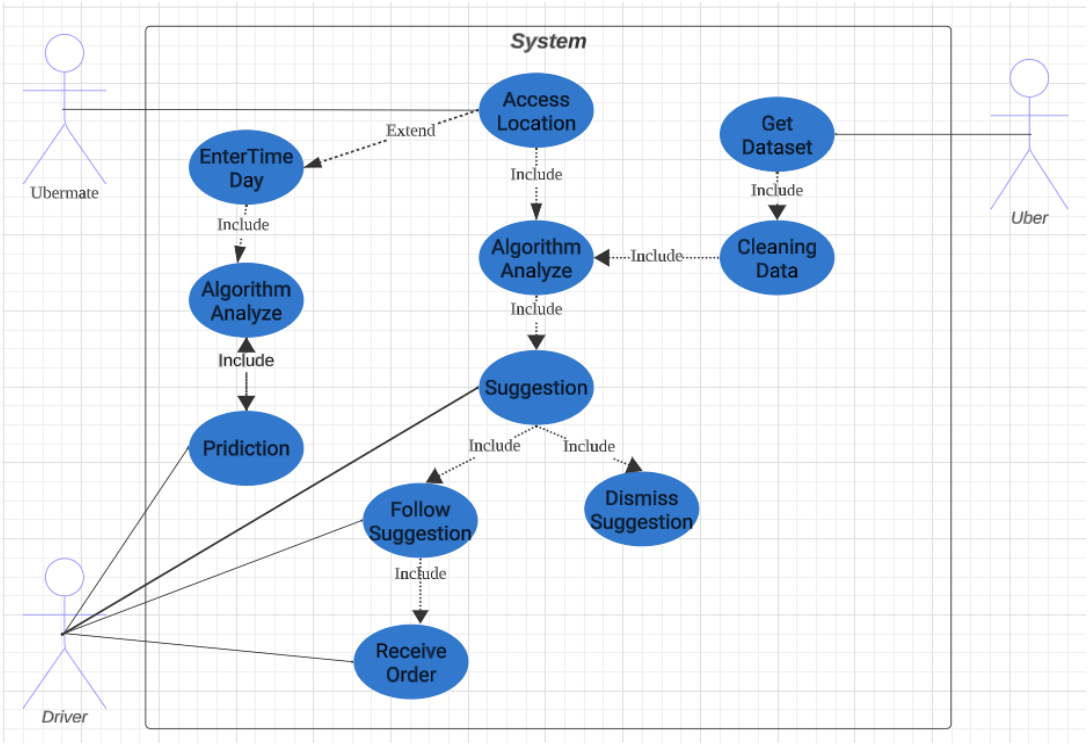

Figure3. Use Case Diagram

2.4 Ubermate platform

The provided diagram represents a system workflow for the Ubermate platform. The key actors include Ubermate, an entity that is a user or subsystem; Driver, representing an individual operating a vehicle; and Uber, denoting the overarching ride-sharing platform. The main process requires inputting a specific time and day, using various algorithms analyzing data, producing a prediction related to suggestions, and subsequently presenting a recommendation to the Driver. The Driver then decides to either act on the suggestion or dismiss it, culminating in receiving a specific order. Supplementary processes involve the system accessing the Driver's or passenger's location, data procurement, and refining the acquired data before its analysis. Relationship-wise, "Include" signifies a process being a subset of a primary process, as seen with data access and cleaning being components of data analysis. Conversely, "Extend" demonstrates a process building on another, akin to how "Enter Time Day" leads to location access.

Overall, Ubermate provides specific time and day details, the system then fetches, cleans, and analyzes data, and based on that, makes a prediction. This prediction is transformed into a suggestion for the Driver, who can then decide to follow or dismiss it. The final action results in the Driver receiving an order or directive. It’s a platform to optimize driving tasks based on real-time data and predictive algorithms.

3. Project Schedule

We made a gantt chart for each step of the project.

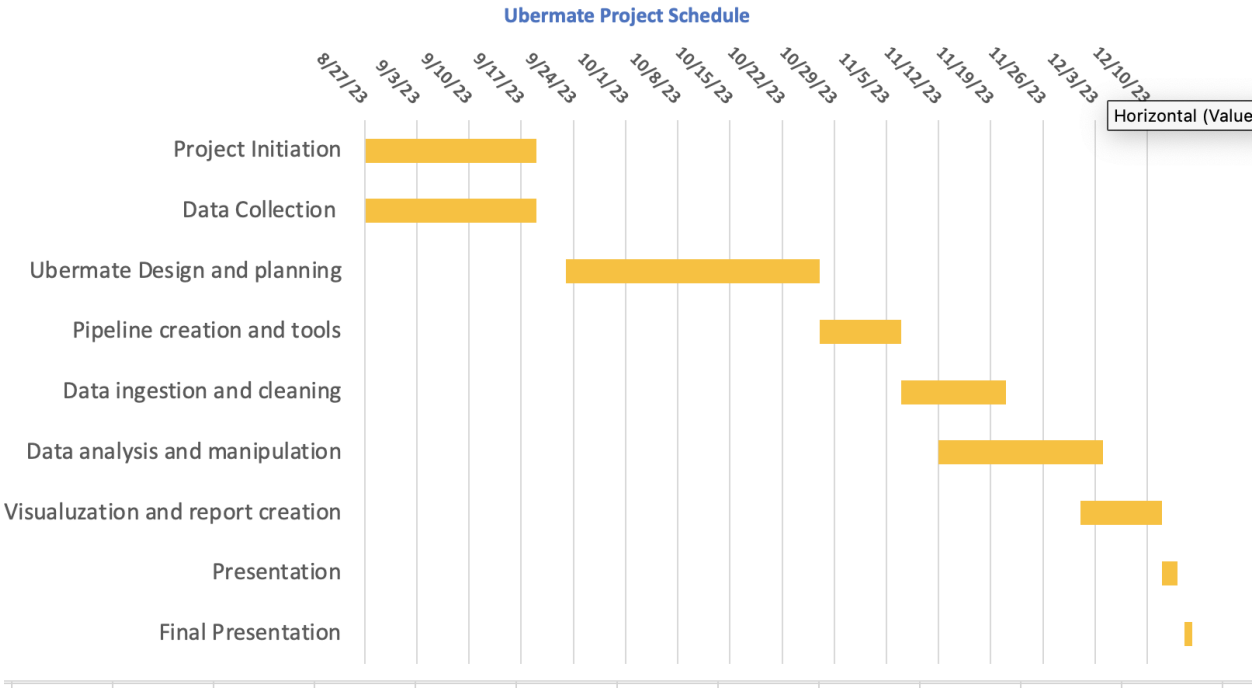

Figure4. Gantt Chart for the Ubermate Project Schedule

4. Evaluation

In order to evaluate the performance of our model, we are taking a stepwise approach. We will first evaluate the time it takes for data to be ingested, cleaned, and formatted by feeding data partitions of various sizes to the pipeline. We will then be running our product in various locations throughout New York. Within each hotspot selected, We will observe the variation in average price amongst the locations and the distance between the driver and the hotspot at that time to see if the ranking is appropriate. We will start with a small plot of New York land and proceed to increase the evaluation range to test if the result remains consistent and observe how performance speed is impacted. Ultimately, we want to make sure the performance speed of the product is stable throughout the process, from the datapoint being entered to the recommended locations being displayed to the user.

5. Conclusion

In conclusion, the project planning phase has provided us with a comprehensive roadmap for the successful execution of the Ubermate Project. Through meticulous research, scoping, and risk assessment, we have established a clear understanding of the project's objectives, requirements, and tools needed. The development of a detailed work breakdown structure and project plan ensures that all necessary tasks and milestones are accounted for and properly sequenced. With a well-defined plan in place, we are now poised to move into the subsequent project phases with confidence, knowing that our approach is structured and well-prepared to achieve the project's goal of enhancing the probability of drivers receiving passenger requests within the shortest amount of time possible.