CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING

Ola Bike Ride Request Demand Forecast

PG-DBDA September 2023

Submitted By:

Project Team 3

Sri Krishna Vijeta Choudhary
Pratik Kulkarni Anagha Nemade
Trupti Gujarathi

INDEX

1 INTRODUCTION
2 PROBLEM STATEMENT
3 LIBRARIES USED
3.1 PANDAS
3.2 NUMPY
3.3 MATPLOTLIB
3.4 SKLEARN
3.5 PYSPARK CORE MODULES
3.6 WINDOWS FUNCTIONS
3.7 GEOSPATIAL LIBRARIES
3.8 ADDITIONAL PYSPARK FUNCTIONS
4 TECHNOLOGIES USED
4.1 AZURE SERVICES
4.2 AZURE STORAGE ACCOUNT
4.3 AZURE MACHINE LEARNING
4.4 AZURE DATA BRICKS
4.5 POWER BI
5 PROJECT METHODOLOGY
5.1 DATASET
5.2 DATA INGESTION AND RAW DATA STORAGE
5.3 DATA PREPROCESSING IN DATA BRICKS WITH PYSPARK
5.4 GEOSPATIAL FEATURE ENGINEERING - CLUSTERING/SEGMENTATION
5.5 MACHINE LEARNING & MODEL DEVELOPMENT
5.6 DATA VISUALIZATION
6 CONCLUSION AND FUTURE SCOPE

INTRODUCTION

The ride-hailing (Ola) service sector has been expanding for a few years, and it is anticipated to continue expanding in the near future. Ola drivers must decide where to wait for passengers since they may arrive rapidly. Additionally, passengers like an immediate bike service whenever required. People who have issues with booking Ola bikes, which sometimes cannot be fulfilled or the wait time for the arrival of the trip is particularly lengthy owing to the lack of a nearby Ola bike. If you successfully reserve an Ola bike in one go, consider yourself fortunate.

Ride-sharing services like Ola Bikes have transformed urban transportation by offering a quick and affordable way to navigate busy cities. However, the success of such services heavily relies on efficiently matching riders with available bikes. This project focuses on developing a model to predict the demand for Ola Bike rides in specific areas, using historical data. Our objective is to optimize how drivers are allocated across different regions to meet demand effectively, reduce waiting times for customers, and increase the service's reliability and competitiveness.

Ride request demand prediction is the process of using historical data to forecast future ride requests in a particular area. Managers may pre-allocate bike resources in cities with the aid of accurate and real-time demand forecasting, which helps drivers find clients more quickly and cuts down on passenger waiting times. This work develops a model to forecast supply and demand mismatches using information from the leading ride-hailing company in Bangalore. The leading ride-hailing business in Bangalore, Ola, handles more than 1 lakh rides daily and gathers more than 5GB of data.

This report will outline the approach taken to build the forecasting model, including how we collected and processed the data, the types of models we considered, and how we evaluated their performance. We will also discuss how the model can be applied in practice to improve Ola Bikes' operations and decision-making processes. By leveraging data-driven insights, we aim to demonstrate how strategic planning can significantly boost service efficiency and customer satisfaction in the competitive ride-sharing market.

PROBLEM STATEMENT

Having an online system that can book rides whenever necessary offers several advantages.Firstly, it provides convenience and saves time for passengers, as they can book a ride with just a few taps on their phone. No more waiting on hold or searching for a bike/cab on the streets. Secondly, it ensures reliability and availability, as the system can connect passengers with nearby rides quickly.

Ola Bikes are suffering losses and losing out from their competition due to their inability to fulfill the ride requests of many users. To tackle this problem you are asked to predict demand for rides in a certain region and a given future time window. This would help them allocate drivers more intelligently to meet the ride requests from users.

Develop a demand forecasting model for Ola Bikes to predict ride requests in a specific region based on historical data. The goal is to optimize driver allocation and improve overall service efficiency by accurately estimating the demand for rides at a given latitude and longitude for a specified future time window. Also, allocate drivers intelligently, reduce losses, and enhance competitiveness by fulfilling a higher percentage of ride requests in the targeted region.

LIBRARIES USED

1 Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.
Pandas are also able to delete rows that are not relevant, or contain wrong values, like empty or NULL values. This is called *cleaning* the data.

2 Numpy

NumPy is a powerful library for numerical computing in Python. While not inherently a PySpark library, it is commonly used in conjunction with PySpark for array operations and numerical computations. In the provided code, it is likely used for data type conversions or other numerical transformations.

3 Matplotlib

4

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in Jupyter Lab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

5 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Important features of scikit-learn:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
- Accessible to everybody and reusable in various contexts.
- Built on the top of NumPy, SciPy, and matplotlib.
- Open source, commercially usable – BSD license.

6 PySpark Core Modules

pyspark.sql:

This module is the core component of PySpark for structured data processing. It provides classes like SparkSession, which is the entry point to any PySpark functionality. It allows users to create DataFrames, perform SQL-like operations, and interact with Spark SQL.

pyspark.sql.functions:

The functions module within PySpark provides a variety of built-in functions for data manipulation and transformation. Commonly used functions include mathematical operations, string manipulations, date/time functions, and aggregations. The col function, for example, is used to reference a DataFrame column.

7 Window Functions

pyspark.sql.window: Used for defining and working with window specifications when using window functions.

8 Geospatial Libraries

Geopy Library: This library is used for geocoding and reverse geocoding. Geocoding involves converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), while reverse geocoding is the opposite process.

9 Additional PySpark Functions:

SparkSession:

pyspark.sql.SparkSession Class: The SparkSession class is the entry point to any PySpark functionality. It consolidates the features of the older SQLContext and HiveContext and provides a unified interface. It is required to create DataFrames, register DataFrames as tables, and execute SQL queries.

unix_timestamp Function: This function is part of PySpark's SQL functions and is used to convert a timestamp string into a Unix timestamp. The Unix timestamp represents the number of seconds that have passed since 1970-01-01 00:00:00.

udf (User Defined Function): PySpark's udf function is used to define custom functions that can be applied to PySpark DataFrames. It allows users to extend PySpark's functionality with their own logic.

geodesic Function:

This function is part of the geopy.distance module and is used for calculating the geodesic distance between two points on the Earth. It provides accurate distance measurements, considering the Earth's curvature.

TECHNOLOGIES USED

AZURE SERVICES

Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a broad range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications or run existing applications in the public cloud.

1 AZURE STORAGE ACCOUNT

A storage account is a container that bands a set of Azure Storage services together. Only data services from Azure Storage can be composed in a storage account. Integrating data services into a storage account allows the user to manage them as a group. The settings specified while creating the account, or setting that is changed after creation, is applicable everywhere. Once the storage account gets deleted, all the data stored inside gets removed.

The Azure Storage platform comprises the following data services:

1.1 **Azure Blobs** are an immensely scalable object store for text and binary data.

1.2 **Azure Files** are organized file shares for cloud or on-premises deployments**.**

1.3 **Azure Queue** is a messaging store for consistent messaging between application components.

1.4 **Azure Tables** are NoSQL stores for schema-less storage of structured data.

1.5 **Azure Disks** are block-level storage volumes for Azure Virtual Machines.

2 AZURE MACHINE LEARNING

Azure Machine Learning is a cloud service for accelerating and managing the machine learning (ML) project lifecycle. ML professionals, data scientists, and engineers can use it in their day-to-day workflows to train and deploy models and manage machine learning operations (MLOps).

Machine Learning studio offers multiple authoring experiences depending on the type of project and the level of your past ML experience, without having to install anything.

Notebooks: Write and run your own code in managed Jupyter Notebook servers that are directly integrated in the studio.

Visualize run metrics: Analyze and optimize your experiments with Visualization.

Azure Machine learning designer: Use the designer to train and deploy ML models without writing any code. Drag and drop datasets and components to create ML pipelines.

Automated machine learning UI: Learn how to create automated ML experiments with an easy-to-use interface.

Data labeling: Use Machine Learning data labeling to efficiently coordinate image labeling or text labeling projects.

3 AZURE DATABRICKS

Azure Databricks is a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale. The Databricks Data Intelligence Platform integrates with cloud storage and security in your cloud account, and manages and deploys cloud infrastructure on your behalf.
Azure Databricks provides tools that help you connect your sources of data to one platform to process, store, share, analyze, model, and monetize datasets with solutions from BI to generative AI.
The Azure Databricks workspace provides a unified interface and tools for most data tasks, including:

- Data processing scheduling and management, in particular ETL
- Generating dashboards and visualizations
- Managing security, governance, high availability, and disaster recovery
- Data discovery, annotation, and exploration
- Machine learning (ML) modeling, tracking, and model serving
- Generative AI solutions

4 POWER BI

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Your data might be an Excel.

Spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets you easily connect to your data sources, visualize and discover what's important, and share that with anyone or everyone you want.

The parts of Power BI:

Power BI consists of several elements that all work together, starting with these three basics:

- A Windows desktop application called Power BI Desktop.
- An online software as a service (SaaS) service called the Power BI service.
- Power BI Mobile apps for Windows, iOS, and Android devices.

METHODOLOGY

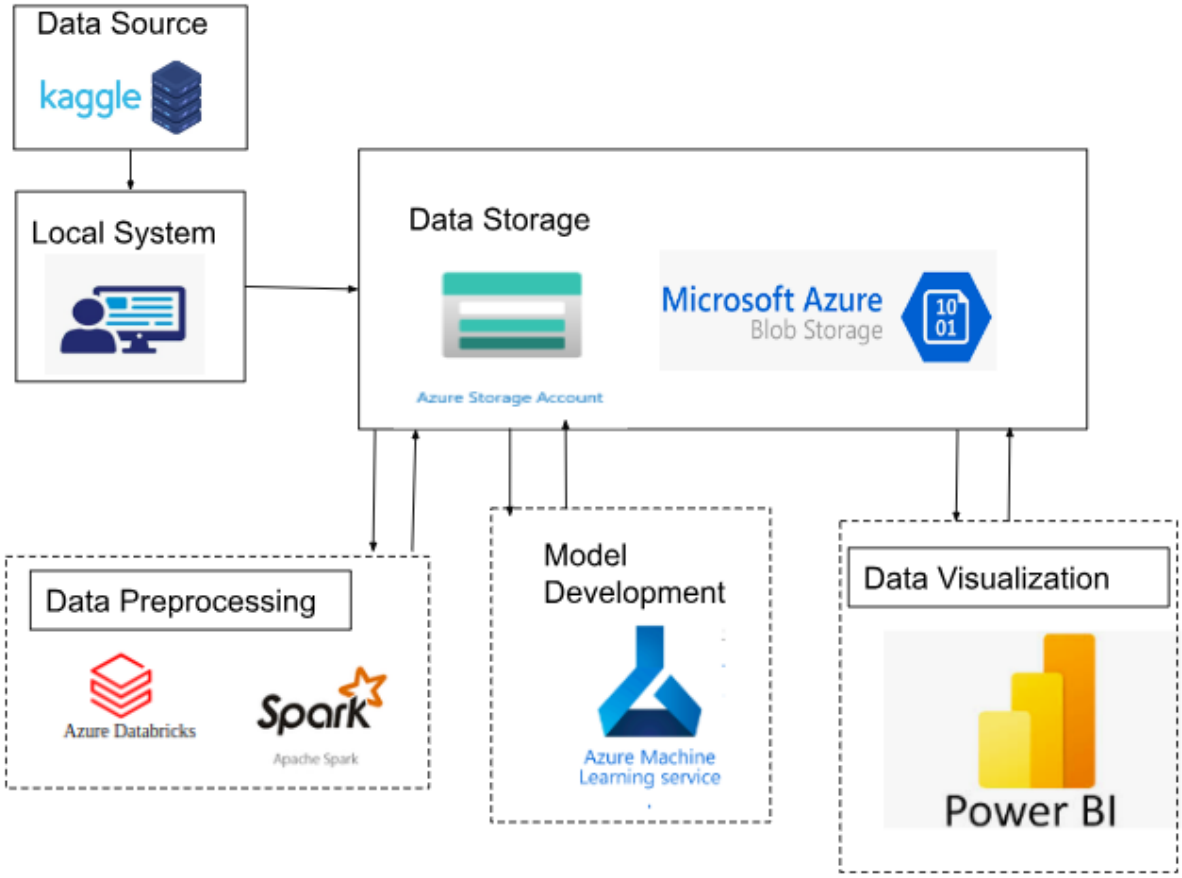


Fig.1. Project Architecture

1 DATASET

The data set used in this study was a ride request dataset. This dataset would have the following attributes:

Ride booking time, pickup and drop point latitude-longitude. The number of data points related to ride requests are, the columns of the data are Id of the customer, timestamp booking, pickup latitude, pickup longitude, drop latitude and drop longitude.

Every user has a unique customer id, the booking timestamp is the date and time of the ride booking (IST time), the pickup latitude is the ride request pickup latitude, the pickup longitude is the ride request pickup longitude, the drop latitude is the ride request drop latitude, and the drop longitude is the ride request drop longitude.

2 DATA INGESTION AND RAW DATA STORAGE:

Source of Raw Data: Raw data uploaded from the local system. Data Ingestion to Azure Data Lake using a container.

3 DATA PREPROCESSING IN DATABRICKS WITH PYSPARK

To build a predictive model on the demand of rides in a particular region at a given time, we first need to preprocess the data to find the estimated true demand by customers.

To get estimated true demand, here we have used the below criteria to get rid of high probable bad ride requests.

3.1 Count only 1 ride request by a user, if there are multiple bookings from the same latitude and longitude within `h` hours of the last booking time.

3.2 If there are ride requests within `m` minutes of the last booking time consider only 1 ride request from a user (latitude and longitude may or may not be the same).

3.3 If the geodesic distance from pickup and drop point is less than 50 meters consider that ride request as a fraud ride request.

3.4 Consider all ride requests where pick up or drop location is outside India bounding box: [“6.2325274”, “35.6745457”, “68.1113787”, “97.395561”] as a system error.

3.5 We would not love to serve intercity rides, or long trips on bike hence if pick up and drop geodesic distance > 500kms; we remove those rides.

3.6 We don't want to provide intercity rides or lengthy bike excursions; thus, we remove such services if the geodesic distance between the pick-up and drop-off points is more than 500 km.

4 GEOSPATIAL FEATURE ENGINEERING - CLUSTERING/SEGMENTATION

Due to the fact that geographical data cannot be used for demand forecasting activities, geospatial engineering was necessary. Given 4 million data points, using standard K-means for clustering would take hours of computing time. Consequently, via a technique known as "Mini Batch K-Means Clustering," we have subdivided the whole of Bangalore into 50 distinct zones.

Mini-Batch-K-Means is a variation of the K-Means method that still aims to optimize the same objective function while using mini-batches to speed up processing. In each training cycle, mini-batches, which are subsets of the input data, are randomly picked. By using these mini-batches, the amount of computation needed to get a local solution is reduced.

5 MACHINE LEARNING & MODEL DEVELOPMENT

5.1 K-Means

K-means is a popular clustering algorithm used in unsupervised machine learning. K-means aims to partition a dataset into K clusters, where each data point belongs to the cluster with the nearest mean (centroid). K-means typically uses Euclidean distance to measure the similarity between data points and centroids.

Algorithm Selection:

K-means is a centroid-based clustering algorithm that iteratively assigns data points to the nearest cluster centroid and updates the centroids to minimize the within-cluster sum of squares.

Model Training:

1. **Initialization:** Randomly initialize K cluster centroids.
2. **Assignment of the cluster**: Assign each data point to the nearest cluster centroid based on a distance metric (e.g., Euclidean distance).
3. **Calculation of mean**: Update the cluster centroids by computing the mean of all data points assigned to each cluster.
4. **Flow after calculation**: Repeat the assignment and update steps until convergence criteria are met, such as stable cluster assignments or a maximum number of iterations.
5. **Output**: The final cluster centroids represent the centers of the clusters, and each data point is assigned to one of the clusters based on its proximity to the centroids.

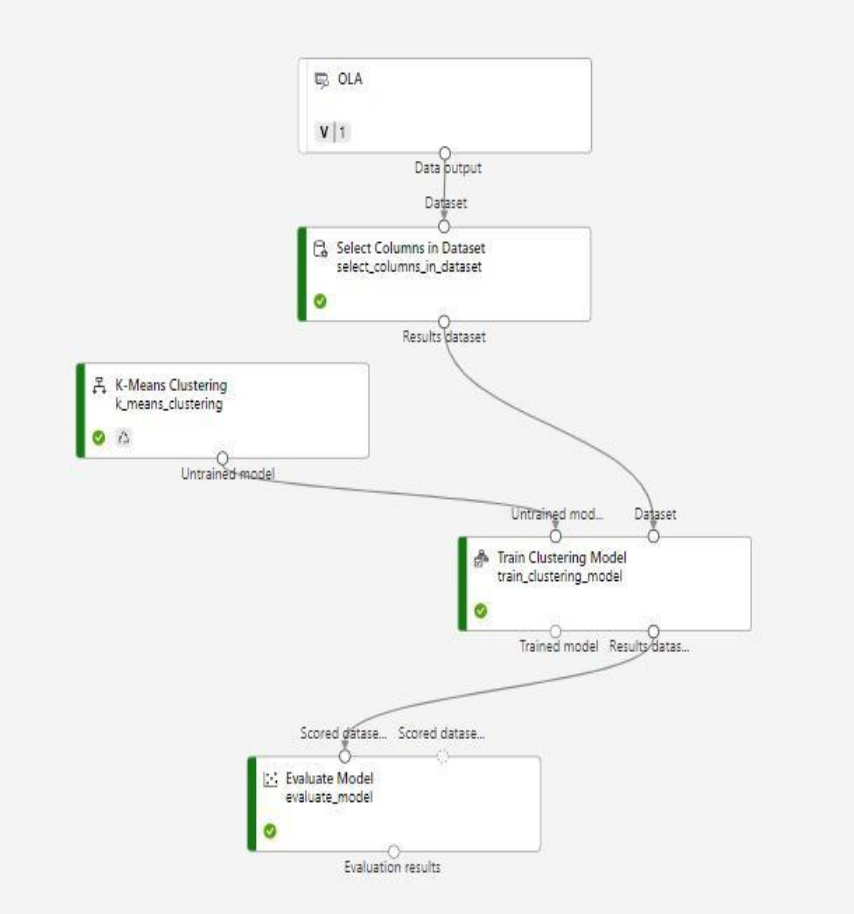
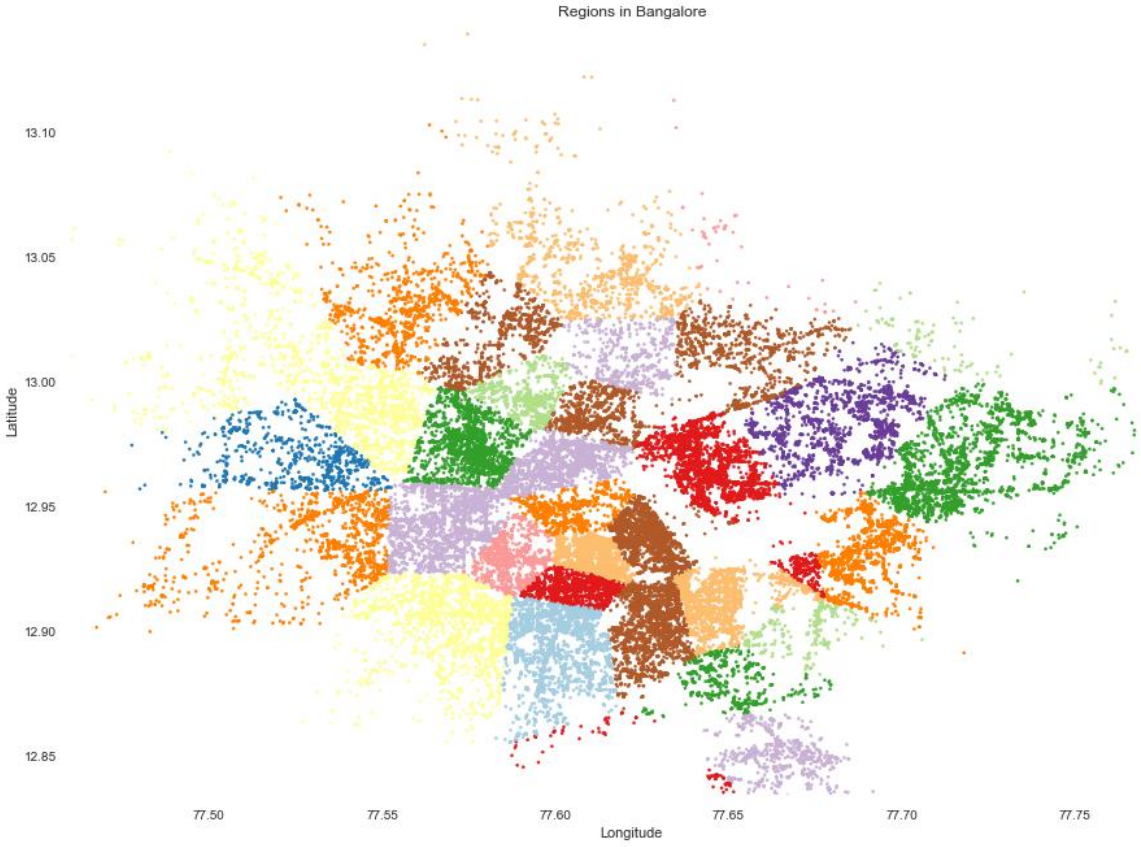
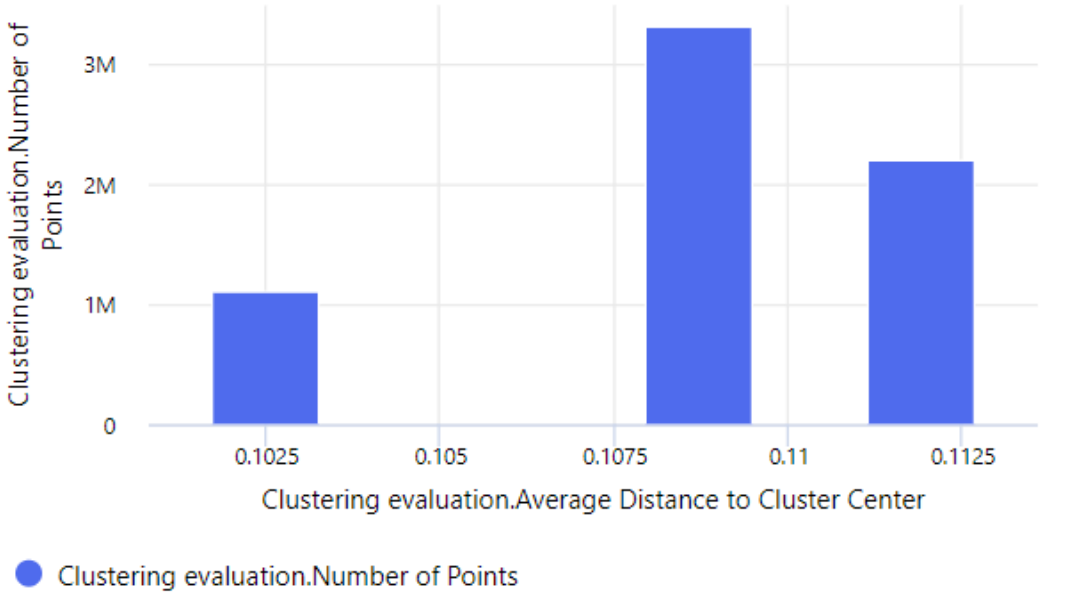


Fig.2. K-Means clustering Modelling


Regions in Bangalore
13.10
13.05
13.00
12.95
12.90
12.85
Latitude
77.50
77.55
77.60
77.65
77.70
77.75
Longitude


Clustering evaluation.Number of Points
3M
2M
1M
0
0.1025
0.105
0.1075
0.11
0.1125
Clustering evaluation.Average Distance to Cluster Center
Clustering evaluation.Number of Points

5.2 Random Forest

Purpose of ML Model:

The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement. The Random Forest algorithm aims to create a reliable and precise predictive model using machine learning principles. Primarily, Random Forest excels in addressing regression and classification challenges, especially when dealing with intricate relationships between the target variable and multiple features.

Algorithm Selection:

Random Forest minimizes the chance of overfitting by combining the forecasts of individual trees and injecting randomness during tree generation. Its selection is attributed to its established capability in managing intricate datasets and delivering precise forecasts across diverse domains.

Describe the process of training the machine learning model using the preprocessed data:

1. First, the preprocessed data, including features and target variables, are partitioned into training and validation sets to assess the model's performance.
2. An ensemble of decision trees is constructed, where each tree is trained on a bootstrapped subset of the training data.
3. Random subsets of features are considered at each split to introduce diversity and reduce correlation among individual trees. Once all trees are trained, the model aggregates their predictions through averaging (for regression) or voting (for classification) to produce the final output.
4. Finally, the model is evaluated using the validation set to assess its performance metrics such as accuracy, RMSE (Root Mean Squared Error), or other relevant metrics.

5.3 XGBoost

Purpose of ML Model:

XGBoost, or Extreme Gradient Boosting, is a sophisticated ensemble learning algorithm that iteratively builds a collection of decision trees. Each tree in the ensemble is trained on a subset of the data, with each successive tree focusing on correcting the errors made by the previous ones. XGBoost is particularly effective at handling regression and classification tasks, especially when confronted with intricate relationships between the target variable and multiple features.

Algorithm Selection:

XGBoost is preferred for its ability to minimize overfitting by sequentially refining the ensemble of decision trees. By iteratively learning from the errors of preceding trees and introducing regularization techniques, XGBoost exhibits robust performance across a wide array of datasets and domains. Its selection is based on its proven track record of managing complex data structures and generating accurate predictions.

Describe the process of training the machine learning model using the preprocessed data:

1. The preprocessed data, including features and target variables, are divided into training and validation sets to evaluate the model's performance.
2. XGBoost constructs an ensemble of decision trees, with each tree trained on a subset of the training data.
3. During tree construction, XGBoost employs gradient boosting techniques to iteratively optimize the model's performance by minimizing the loss function.
4. Regularization parameters are utilized to control the complexity of the model and prevent overfitting.
5. Once all trees are trained, XGBoost aggregates their predictions using a weighted sum to generate the final output.
6. The model's performance is assessed using the validation set, and relevant metrics such as accuracy, RMSE, or others are calculated to evaluate its effectiveness.

DATA VISUALIZATION

The importance of data visualization is simple: it helps people see, interact with, and better understand data. Whether simple or complex, the right visualization can bring everyone on the same page, regardless of their level of expertise.

Data visualization can be used in many contexts in nearly every field, like public policy, finance, marketing, retail, education, sports, history, and more.
Here are the benefits of data visualization:

Storytelling: People are drawn to colors and patterns in clothing, arts and culture, architecture, and more. Data is no different—colors and patterns allow us to visualize the story within the data.

Accessibility: Information is shared in an accessible, easy-to-understand manner for a variety of audiences.

Visualize relationships: It's easier to spot the relationships and patterns within a data set when the information is presented in a graph or chart.

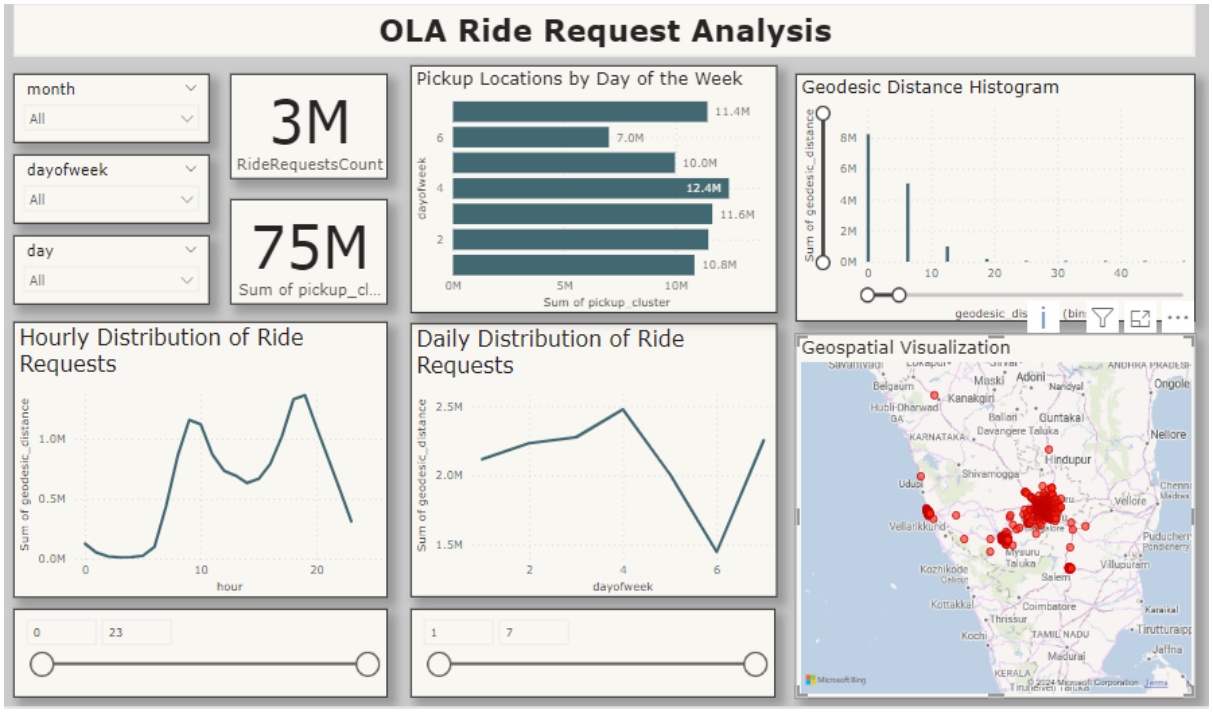
Exploration: More accessible data means more opportunities to explore, collaborate, and inform actionable decisions.

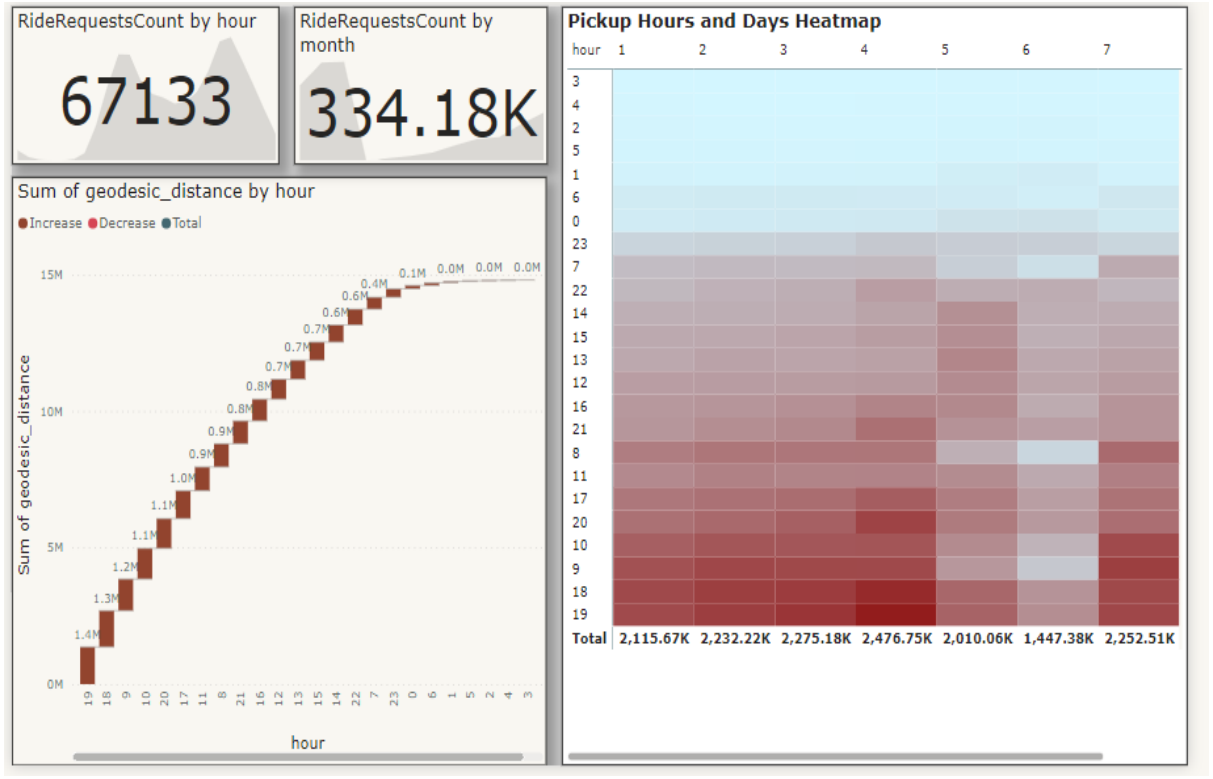
There are plenty of data visualization tools out there to suit your needs, we have used Power BI for our project, Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Your data might be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets you easily connect to your data sources, visualize and discover what's important, and share that with anyone or everyone you want.

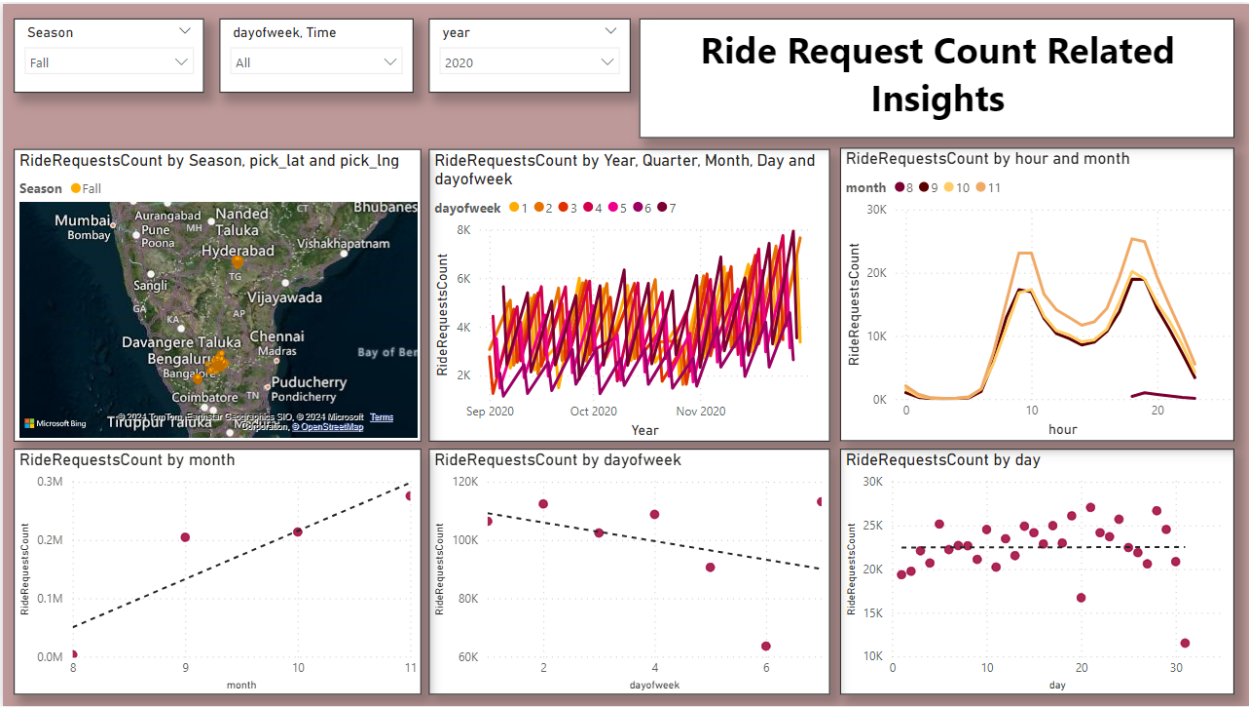
The parts of Power BI

Power BI consists of several elements that all work together, starting with these three basics:

- A Windows desktop application called Power BI Desktop.
- An online software as a service (SaaS) service called the Power BI service.
- Power BI Mobile apps for Windows, iOS, and Android devices.


OLA Ride Request Analysis
month
All
dayofweek
All
day
All
3M
RideRequestsCount
75M
Sum of pickup_cl...
Pickup Locations by Day of the Week
dayofweek
11.4M
7.0M
10.0M
12.4M
11.6M
10.8M
0M
5M
10M
Sum of pickup_cluster
Geodesic Distance Histogram
Sum of geodesic_distance
8M
6M
4M
2M
0M
0
10
20
30
40
geodesic_dis
Hourly Distribution of Ride Requests
Sum of geodesic_distance
1.0M
0.5M
0.0M
0
10
20
hour
Daily Distribution of Ride Requests
Sum of geodesic_distance
2.5M
2.0M
1.5M
2
4
6
dayofweek
Geospatial Visualization
Belgaum
Kanakgiri
Adoni
Nandyal
Ongole
Hubli-Dharwad
Ballari
Guntakal
Devangere Taluka
KARNATAKA
Nellore
Hindupur
Shivamogga
Udupi
Vellore
Chennai
Madras
Puducherry
Pondicherry
Mysuru
Taluka
Villupuram
Salem
Kozhikode
Calicut
Kottakkal
Coimbatore
Karaikal
Thrissur
Kochi
TAMIL NADU
Tirutturaipp
Madurai
Jaffna
KERALA
0
23
1
7


RideRequestsCount by hour
67133
RideRequestsCount by month
334.18K
Sum of geodesic_distance by hour
Increase
Decrease
Total
15M
10M
5M
0M
Sum of geodesic_distance
hour
Pickup Hours and Days Heatmap
hour
1
2
3
4
5
6
7
Total
2,115.67K
2,232.22K
2,275.18K
2,476.75K
2,010.06K
1,447.38K
2,252.51K


Season
Fall
dayofweek, Time
All
year
2020
Ride Request Count Related Insights
RideRequestsCount by Season, pick_lat and pick_lng
Season Fall
Mumbai
Bombay
Aurangabad
Pune
Poona
Nanded
Taluka
Hyderabad
Vishakhapatnam
Bhubanes
Sangli
Vijayawada
Davangere Taluka
Chennai
Madras
Bengaluru
Bangalore
Puducherry
Pondicherry
Coimbatore
Tiruppur Taluka
Bay of Ben
RideRequestsCount by Year, Quarter, Month, Day and dayofweek
dayofweek 1 2 3 4 5 6 7
RideRequestsCount
Sep 2020
Oct 2020
Nov 2020
Year
RideRequestsCount by hour and month
month 8 9 10 11
RideRequestsCount
hour
RideRequestsCount by month
RideRequestsCount
month
RideRequestsCount by dayofweek
RideRequestsCount
dayofweek
RideRequestsCount by day
RideRequestsCount
day

CONCLUSION AND FUTURE SCOPE

CONCLUSION

In order to handle the issue of ride demand forecasting, a novel XGBoost regressor model is proposed in this work. The data preprocessing, geospatial engineering methods are utilized to convert latitude and longitude, to cluster Id using Mini-Batch K Means algorithm, and then multi-step forecasting is used to forecast the demand for ride requests coming from an area at a certain time.

As the Random Forest Algorithm was overfitting and affecting the accuracy of the model we went ahead with XGBoost algorithm through that the accuracy came out to be 85.85%

Algorithm	RMSE Train	RMSE Test
Random Forest	1.9048	4.3232
XGBoost	2.5953	4.3232

FUTURE SCOPE

User Interface (UI) Enhancement:
Design and implement a user-friendly interface for the fare prediction system. Include intuitive visuals and interactive elements to enhance user experience.

Real-time Weather Integration:
Integrate weather forecasting data into the prediction model. Consider variables like rain or traffic delays caused by weather conditions to improve accuracy.

Multi-city Support:
Extend the prediction model to cover multiple cities, considering unique traffic patterns and variables in different locations.

Advanced Predictive Models:
Development of more sophisticated predictive models using advanced machine learning techniques. This could involve incorporating additional features such as weather conditions, local events, and traffic patterns to enhance the accuracy of demand predictions.

Real-time Data Integration:
Implementation of real-time data integration to continuously update and refine predictive models.This includes leveraging real-time traffic data, user behavior, and external factors that might influence ride demand.

Expansion to New Services:
Exploration of additional services beyond traditional ride-sharing, such as integrating micro-mobility options like electric scooters or bicycles. This diversification can help capture a broader market and address specific transportation needs.

Collaboration with Public Transportation:
Collaboration with public transportation agencies to create a seamless and integrated transportation network.This could involve offering incentives for users to switch between different modes of transportation based on predicted demand and traffic conditions.

Continuous Improvement and Adaptation:
Establishing a continuous improvement framework to adapt to changing user behaviors, market dynamics, and technological advancements.Regularly updating models and strategies based on new data and insights.