CHAPTER-1
INTRODUCTION

1.1 Purpose

The purpose of this project is to delve into the intricate dynamics between driver behavior and customer satisfaction within the ride-hailing services industry. By leveraging advanced data engineering techniques and analytics, the project aims to uncover valuable insights that can inform strategic decision-making and drive tangible improvements in service quality. Through meticulous data collection and analysis, the project seeks to identify key factors influencing customer ratings, such as driver acceptance rates and cancellation rates. By understanding the nuanced preferences and expectations of riders, ride-hailing companies can implement targeted strategies to optimize driver performance, enhance service quality, and foster greater customer satisfaction and loyalty. Ultimately, the overarching goal of the project is to empower industry stakeholders with the knowledge and tools needed to thrive in an increasingly competitive and dynamic landscape, thereby shaping the future of ride-hailing services.

1.2 Scope

The scope of the project encompasses a comprehensive exploration of the correlation between driver behavior metrics and customer ratings within the ride-hailing services industry. This includes the analysis of key factors such as driver acceptance rates, cancellation rates, and customer satisfaction ratings. The project aims to leverage advanced data engineering techniques and analytics to uncover valuable insights that can inform strategic decision-making and drive improvements in service quality.

1. Data Analysis: This involves the thorough examination of driver behavior metrics and customer ratings within the ride-hailing services industry. The analysis will focus on key performance indicators such as driver acceptance rates, cancellation rates, and customer satisfaction ratings.

2. Strategy Development: Building upon the insights gleaned from the data analysis, this aspect involves the formulation of data-driven strategies to optimize driver performance and enhance service quality.

The scope also includes the documentation and dissemination of research findings through reports, presentations, and academic publications, contributing to the broader knowledge base in the field of transportation analytics.

1.3 Motivation

The project is motivated by the burgeoning significance of ride-hailing services in modern urban transportation and the pressing need to enhance the overall customer experience within this industry. With the exponential growth of platforms like Uber and Lyft, understanding the nuanced interplay between driver behavior metrics and customer ratings has become indispensable for ride-hailing companies striving to maintain a competitive edge. By harnessing the power of advanced data analytics, the project seeks to unlock valuable insights that can inform strategic decision-making and drive tangible improvements in service quality. Through meticulous analysis of key performance indicators such as driver acceptance rates, cancellation rates, and customer satisfaction ratings, the project aims to identify actionable recommendations for optimizing driver performance and fostering greater customer satisfaction.

1.4 Proposed System

The project encompasses four distinct phases, each integral to unveiling the link between driver behavior and customer ratings.

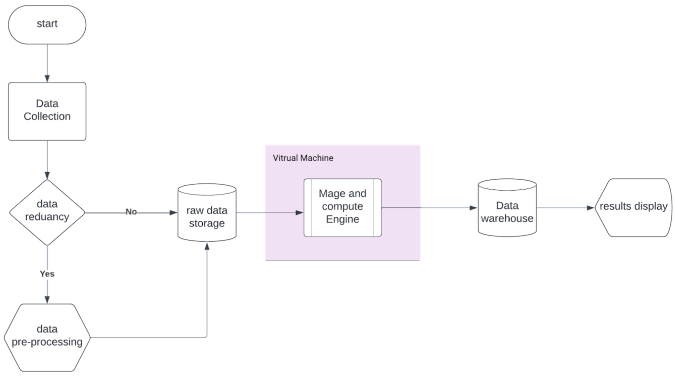

Figure 1.1. Workflow of the suggested pipeline for data engineering and analysis

Data Collection and Preprocessing:

- Identify and collect a comprehensive dataset from publicly available sources, similar to Uber's ride-hailing data.
- Implement rigorous data preprocessing techniques to ensure data integrity, handle missing values, and eliminate outliers.

Data Engineering with Mage and Google BigQuery:

- Design and implement an efficient Extract, Transform, Load (ETL) pipeline using the Mage platform to orchestrate data transformations and aggregations.
- Utilize Google BigQuery as the data warehousing solution to process and analyze the dataset using advanced SQL queries.
- Explore statistical analysis techniques to identify significant correlations between driver behavior and customer ratings.

Identifying Driver-Customer Dynamics:

- Analyze driver performance metrics, including acceptance rate and cancellation rate, concerning customer ratings.
- Gain insights into customer preferences and how driver actions influence their experience.
- Uncover patterns and trends that highlight the impact of driver behavior on customer satisfaction.

Optimizing Driver Performance and Business Growth:

- Leverage the project's findings to optimize driver performance and enhance service quality in ride-hailing services.
- Formulate data-driven strategies to incentivize drivers and align their actions with customer expectation.

CHAPTER- 2
LITERATURE SURVEY

2.1 Introduction to Literature Survey

In order to do analysis, modern data-driven applications frequently need to identify and transfer pertinent data from several sites to a single storage location. Some alternative solutions for data sharing have been offered that make use of cloud-based hosting and high-speed networks to get around what is typically a challenging, time-consuming, and arduous operation. Other alternative solutions center around the provision of shared computing resources.

2.2 Literature Survey

1.R. J. Sandusky, "Computational provenance: Dataone and implications for cultural heritage institutions," in 2016 IEEE International Conference on Big Data (Big Data). IEEE, 2016, pp. 3266–3271.

A project called DataONE [1], [2] aims to make earth and environmental research data sources easier to find, search, and access. Through the provision of networked computing resources, the Open Science Grid [3], [20] facilitates scientific research.In an effort to reduce data travel, SciServer gathers all of the necessary data at one central storage site, but it does so at the location where most of the data is already present. Additionally, SciServer moves the analyses to the shared storage location by forwarding Jupyter Notebooks [22].

2. D. Medvedev, G. Lemson, and M. Rippin, "Sciserver compute: Bringing analysis close to the data," in Proceedings of the 28th international conference on scientific and statistical database management, 2016, pp. 1–4.

Research infrastructures that combine storage, high-performance computing, and analytical tools are the goal of other data-driven applications (e.g., XSEDE [11], [23], NeCTAR [24], PRACE [25], and EGI [26]).Users can share data repositories and dispersed computing resources by using these applications. Science Gateways (SGs) [5], [17]–[20] may employ the solutions to create (web) portals and user interfaces (UIs) that let scientists (such biologists and chemists) access, create, and carry out analytic workflows.

3. A. Talukder, M. Elshambakey, S. Wadkar, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, "Vifi: Virtual information fabric infrastructure for data-driven discoveries from distributed earth science data

Scientists are relieved of the responsibility and necessary knowledge to set up and manage the underlying distributed cyber-infrastructure by SGs. Various end users can share and reuse SG services. SGs can be categorized into instances of SG, such as the Computational Neuroscience Gateway [13], and SG frameworks.

4. D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," IEEE transactions on visualization and computer graphics.

The initiative guarantees the integrity and dependability of its results by using rigorous data gathering and preprocessing methods. In order to improve service quality, the project will investigate this link using advanced analytics on datasets that are similar to those in the industry. By means of rigorous data gathering and analysis, the research explores a number of parameters, including consumer ratings and acceptance rates.

CHAPTER-3
SYSTEM ANALYSIS

3.1 Introduction

This project introduces an interactive visual analytics framework (VAF) aimed at enhancing user interactions and improving the overall user experience with data analytics systems (DAS). Through an examination of various distributed analytical systems, we identified crucial user interactions necessary for running these systems effectively. Both data analysts and end users often encounter challenges in navigating the complexities of data analysis processes. Additionally, users across different domains often struggle to access and interpret the results obtained from DAS. While DAS typically provides visualization toolkits as alternatives to command line interfaces, users are tasked with creating suitable artifacts or exploratory visualizations to evaluate the efficacy of the analysis. The VAF presented in this paper seeks to streamline these interactions and empower users to extract meaningful insights from their data more efficiently.

3.2 Problem Statement

The project aims to explore the correlation between driver behavior (e.g., acceptance rate, cancellation rate) and customer ratings in ride-hailing platforms. With a carefully collected dataset and advanced data engineering and analytics techniques, the project seeks to reveal insights into how driver actions impact customer satisfaction. By understanding this link, ride-hailing companies can optimize driver performance and enhance service quality, leading to increased customer retention and business growth.

3.3 Existing System

The existing system comprises traditional data analytics systems (DAS) that often pose challenges in user interactions and overall user experience. Users, including data analysts and end-users from diverse domains, encounter difficulties in navigating these systems effectively due to their complex interfaces and limited visualization capabilities. Additionally, retrieving and interpreting results from DAS can be cumbersome, requiring users to manually create artifacts or exploratory visualizations. While some DAS offer visualization toolkits, users are responsible for generating appropriate visualizations to assess analysis outcomes.

Moreover, users often face challenges in accessing and interpreting data analysis results, especially across different domain regions. This necessitates retrieving data from servers, which can introduce delays and hinder real-time decision-making processes. Consequently, users may find it challenging to gauge the efficacy of the analysis performed and derive meaningful insights from the data.

Furthermore, the absence of an interactive visual analytics framework (VAF) exacerbates these challenges, as users lack a unified platform to facilitate intuitive data exploration and analysis. As a result, the existing system falls short in meeting the evolving needs of users, who increasingly demand more user-friendly and interactive data analytics solutions.

3.4 Modules Description

Data Preprocessing: Thoroughly preprocess the dataset to ensure data quality, handle missing values, and remove outliers, ensuring reliable analysis.

Data Transformation and Feature Engineering: Utilize data transformation and feature engineering techniques to extract relevant information, such as acceptance rate, cancellation rate, and customer ratings.

Correlation Analysis: Perform statistical analysis to uncover the correlation between driver behavior metrics and customer ratings, quantifying their relationship.

Customer Segmentation: Segment customers based on feedback and preferences to understand distinct customer groups and tailor services accordingly.

Service Quality Improvement Strategies: Formulate action plans to address service quality issues based on data insights, enhancing the overall ride-hailing experience.

Predictive Analytics for Customer Satisfaction: Utilize predictive analytics to anticipate potential customer satisfaction issues and take proactive measures.

CHAPTER – 4
SYSTEM REQUIREMENTS

4.1 Software Requirements

- Operating System: Windows 11 or Linux
- Server-side Script: Python 3.6
- IDE : PyCharm, VS code
- Libraries Used : Big query, mage, jupyter notebook, lucidchart, google compute engine, looker, vscode.

4.2 Hardware Requirements

- Processor : I5/Intel Processor
- RAM : 8GB
- Hard Disk : 128 GB

Apart from these additional components like internet connectivity, camera or storage devices, an image display device, and a computer of a server are required.

4.3 Project Perquisites

- OS module
- programming skills
- Big Query
- Django web framework
- PyCharm
- Python
- Lucid Chart
- Google Computer Engine
- looker

It is important to note that while the project may require proficiency in these technologies and concepts, it also provides an opportunity to enhance and strengthen these skills as part of the project's learning process.

1. Data Understanding and Analysis: Familiarity with data analysis concepts and techniques to explore, clean, and preprocess the ride-hailing dataset effectively.

2. SQL: Proficiency in SQL to perform queries, aggregations, and data manipulations in Google BigQuery.

3. Data Engineering Concepts: Understanding of data engineering principles, including ETL (Extract, Transform, Load) pipelines and data warehousing.

4. Cloud Platforms: Familiarity with Google Cloud Platform (GCP) services, specifically Google Cloud Storage and BigQuery, for data storage and analysis.

5. Data Visualization: Knowledge of data visualization tools such as Looker Studio to create interactive dashboards and visualizations for data exploration.

7. Python Programming: Proficiency in Python programming for data manipulation, data preprocessing, and data analysis tasks.

4.4 **Technologies as Prerequisites:**

1. SQL: Proficiency in SQL is essential for querying and analyzing data in Google BigQuery.

2. Google Cloud Platform (GCP): Understanding GCP services, specifically Google Cloud Storage and BigQuery, is crucial for data storage and analysis in the cloud.

3. Python: Knowledge of Python is essential for data manipulation and analysis tasks, as well as for implementing data engineering pipelines.

4. Data Visualization Tools: Familiarity with data visualization tools like Looker Studio to create interactive dashboards for data exploration and presentation. 5. Statistical Tools: Familiarity with statistical tools and libraries in Python, such as NumPy and Pandas, for performing correlation analysis and other statistical tasks

CHAPTER – 5
SYSTEM DESIGN

5.1 Introduction

In order to streamline user interactions and improve the user experience with a data analytics system (DAS), this paper presents an interactive visual analytics framework (VAF). To do this, we examined a number of distributed analytical systems (e.g., [4], [8], [22], [23]) and determined the essential user interactions needed to run these systems. As such, both the data analysts and the end users may find the entire process of doing a data analysis to be equally difficult. Furthermore, users from other domain regions had to retrieve the resultant data from the server in order to examine the results. DAS frequently offers a visualization toolkit in place of command line interfaces [22], [24]. But users are in charge of creating the appropriate artifacts or exploratory visualizations to gauge how well the analysis performed [25].

Data pre -processing and engineering:

5.2 **Data Identification and collection :** The system shall identify and collect a comprehensive dataset akin to Uber's ride-hailing data from publicly available sources.
It shall ensure the dataset includes relevant information such as driver behavior metrics and customer ratings.

5.3 **Data Pre-processing:** The system shall implement data preprocessing techniques to ensure data integrity. Using the Pandas module to pre-process the data.

5.4 **Data Storage:** The system shall utilize Google Cloud Storage or equivalent for storing the dataset securely.

Data Analysis and Insights Generation:

5.5 **ETL Pipeline:** The system shall design and implement an Extract, Transform, Load (ETL) pipeline for data processing.It shall orchestrate data transformations and aggregations efficiently. Using Mage, a software data engineering pipeline tool to construct the ETL.

5.6 **Statistical Analysis:** The system shall employ statistical analysis techniques to identify correlations between driver behavior and customer ratings.It shall provide insights into customer preferences and the impact of driver actions.

Strategy Formulations and Recommendations:

5.7 **Optimization Strategies:** The system shall formulate data-driven strategies to optimize driver performance and enhance service quality. It shall align driver actions with customer expectations effectively.

5.8 **Actionable Insights:** The system shall generate actionable insights for ride-hailing companies to improve business growth, increase customer retention, and foster a loyal customer base

5.9 System Architecture:

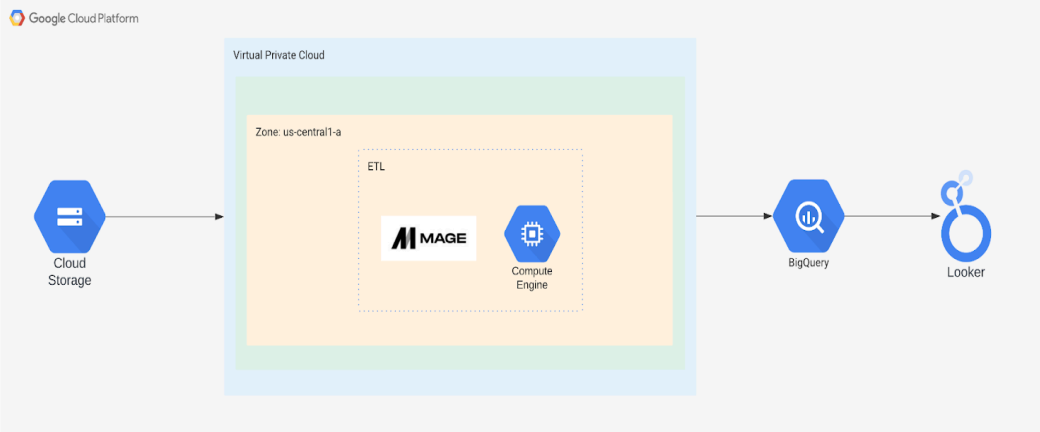

Figure 1.1 Architecture Model

The system we have in mind for our research project is a comprehensive approach to data engineering and analytics that makes use of a range of tools and technologies, including GCP services from Google Cloud Platform. This all-inclusive ecosystem is intended to simplify the data processing workflow, enable effective analysis, and enable researchers to extract meaningful insights from intricate datasets. The GCP services, which provide dependable infrastructure and scalable solutions for managing complex data processing activities, are the foundation of our system design. Google Cloud Storage offers safe and dependable data storage capabilities, acting as the basis for file archiving and retrieval from any location in the cloud. Furthermore, virtual machine deployment and maintenance are made possible by Google Compute Engine, which makes it simple and effective for researchers to execute their applications

BigQuery, Google's data warehousing tool, is a crucial part of our system since it provides strong analytical capabilities and a recognisable SQL-type interface. BigQuery is perfect for processing the enormous volumes of data that are usually encountered in research projects since it allows academics to store and analyze large-scale datasets. BigQuery's highly scalable and cost-effective design guarantees that researchers may execute intricate analytical activities without sacrificing scalability or performance. Looker Studio is a web-based business intelligence application that enhances the GCP services by offering sophisticated reporting and visualization features.

Researchers can easily develop interactive dashboards and visualizations using Looker Studio, which seamlessly connects with GCP services to improve the readability of insights into their data and communicate their findings effectively.To assist the data processing pipeline, our solution integrates multiple tools and technologies in addition to GCP services and Looker Studio. Jupyter Notebooks offer an interactive environment for testing and executing code, allowing academics to try various techniques and approaches.

5.10 UML Diagrams:

UML diagrams are a standardized way of representing different aspects of a software system or process. UML diagrams are not code, but rather a graphical way to visualize and communicate the different components, relationships, and behaviours of a system. UML diagrams can help to improve communication and understanding between stakeholders, developers, and designers.

5.11 Use Case Diagram

Use case diagrams are a particular kind of behavioural diagram that depicts how actors interact with the system. The users or external systems that communicate with the modeled system are represented by actors. The different use cases or scenarios that the technology can be employed in are displayed in use case diagrams. They can aid in identifying system needs and design features by illuminating the connections between use cases and actors.

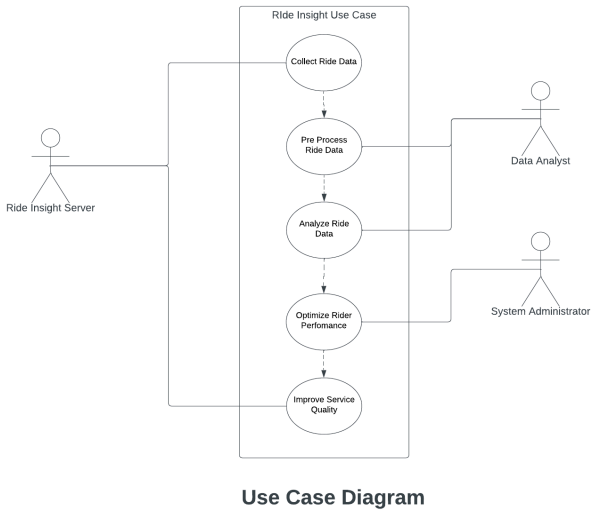

Figure 1.2 Use Case Diagram

5.12 Class Diagram

Class diagrams are a form of a structural diagram that depicts the classes, their characteristics and methods, and the relationships between them as well as the static structure of a system. Class diagrams are useful for creating and comprehending the architecture of a system since they are used to model the data or objects in a system. They can also be used to create system code.

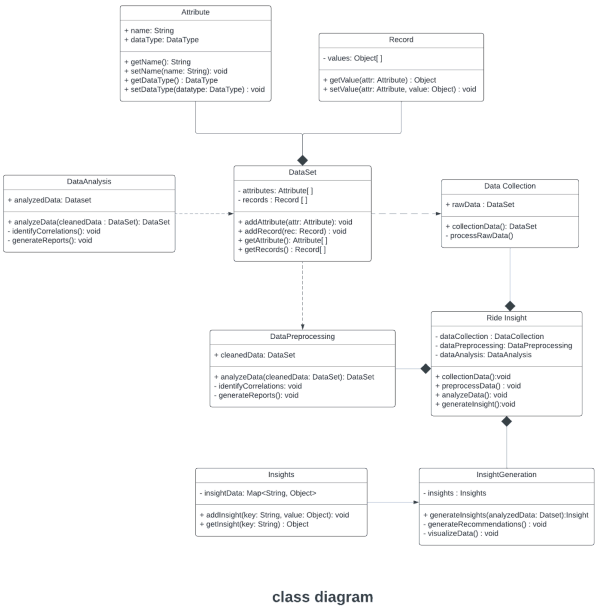

Figure 1.3 Class Diagram

5.13 Sequence Diagram

Sequence diagrams, a sort of behavioural diagram, display how various system components interact with one another across time. Sequence diagrams display the order in which messages are transmitted and received between objects. They help comprehend a system's dynamic behaviour particularly how different objects work together to complete tasks or run processes.

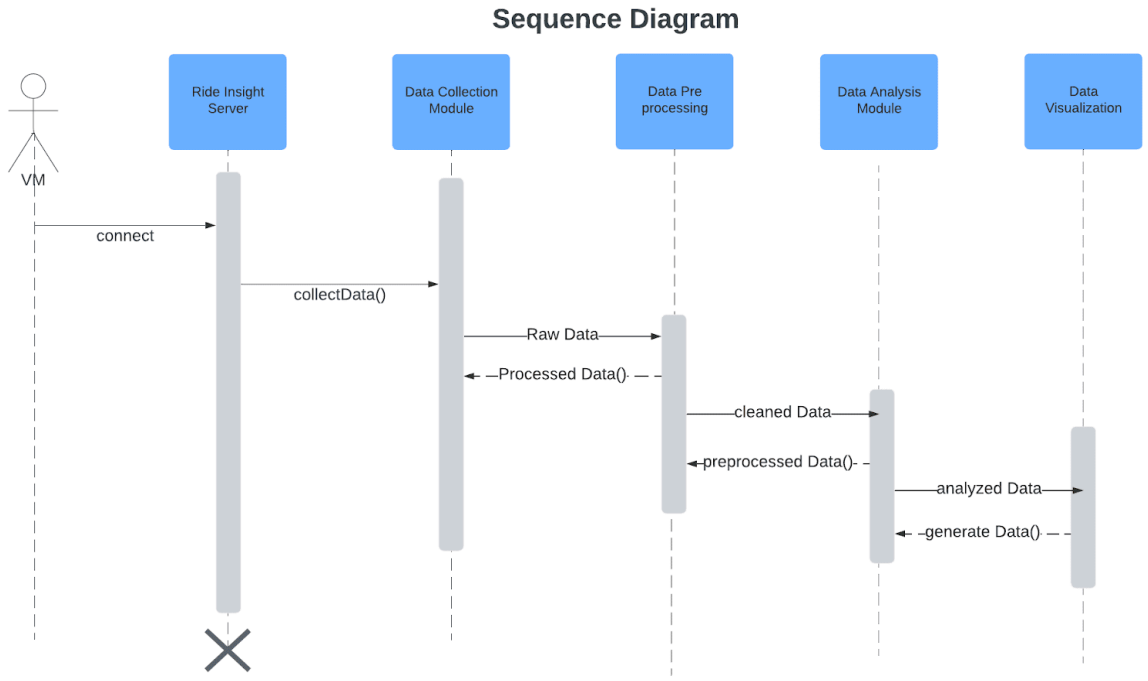

Figure 1.4 Sequence Diagram

5.14 DATA ER MODEL

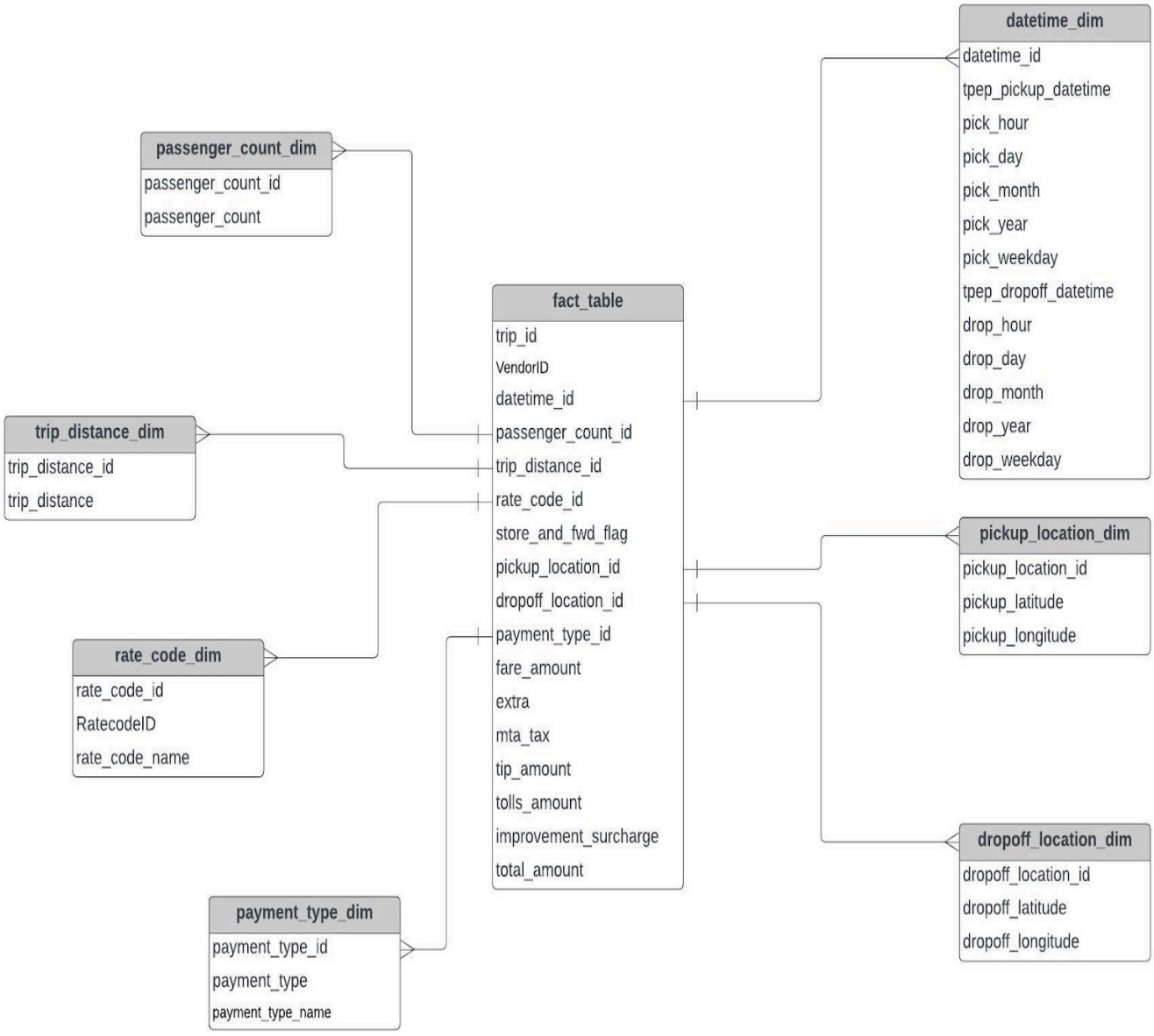

Figure 1.5 ER Diagram

5.16 Deployment Diagram

The deployment diagram depicts a system's deployment view. It is associated with the component diagram. Because the deployment diagrams are used to deploy the components. A deployment diagram is made of nodes. Nodes are simply pieces of actual hardware that are used to deploy the application.

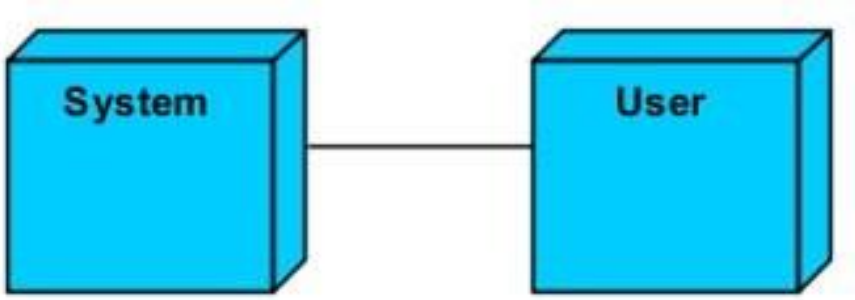

Figure 1.6 Deployment Diagram

5.17 DFD Diagram

A Data Flow Diagram (DFD) is a conventional method for visualizing how information flows within a system. A tidy and unambiguous DFD can graphically display a large portion of the system requirements. It might be manual, automatic, or a combination of the two. It demonstrates how data enters and exits the system, what alters the data, and where data is stored. A DFD's goal is to indicate the extent and bounds of a system. It can be used as a communication tool between a systems analyst and anyone involved in the system which serves as the beginning point for system change.

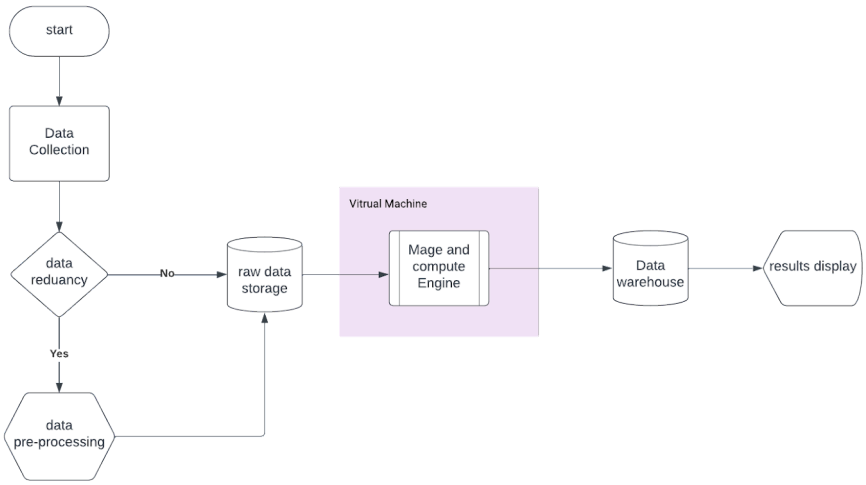

Fig 1.7 DATA flow daigram

CHAPTER-6
IMPLEMENTATION

Technology Description:

Python

Programming paradigms including functional, aspect-oriented, object-oriented, and structured programming are all supported by Python, a flexible language. Additionally, it offers extensions for other paradigms like design by contract and logic programming. Python supports dynamic name resolution and uses dynamic typing, reference counting, and garbage collection to manage memory. Python emphasizes code readability over slight performance gains, but when necessary, programmers can employ C extensions or just-in-time compilation to achieve quicker performance. With built-in capabilities like filter, map, and reduce functions, along with list comprehensions, dictionaries, sets, and generator expressions, Python's design also supports functional programming. The language has a helpful open-source community, as well as extensive support libraries and third-party modules.

Python's Advantages:

Third-Party Modules Availability Comprehensive Support Libraries Open Source and Community Development Learning Ease and Support Available User-Friendly Data Structures Productivity and Speed

6.1.1 Techniques Used

1. **SQL (Structured Query Language):** Having SQL under your belt is key to unlocking data stored in Google BigQuery. It lets you ask specific questions of your data, like a detective sifting through clues. SQL helps you filter, sort, and analyze information to uncover hidden insights.

2. **Google Cloud Platform (GCP):** Think of GCP as a digital toolbox for managing data in the cloud. Services like Google Cloud Storage act as secure vaults for your information, while BigQuery is a super-powered engine for analyzing massive datasets. Mastering GCP empowers you to store and analyze data efficiently.

3. **Python:** Python is a versatile programming language that's a data pro's best friend. It lets you manipulate and analyze data with ease, like a sculptor shaping raw materials into a masterpiece. Python is also perfect for building data pipelines, which are automated workflows that keep your data flowing smoothly.

4. **Data Visualization Tools:** Data visualization tools like Looker Studio are like translators for your data, transforming numbers into clear and compelling visuals. Charts, graphs, and dashboards created with these tools help you explore and understand your data in a flash, making it easy to share insights with others.

Sample Code:

main.ipyb

```
import pandas as pd
#loading the data from dataset
df = pd.read_csv("../data/uber_data_final.csv")
df.info()
df['trip_id'] = df.index
df.head()
df.info()
#converting the pickupdatetime and dropofftime from object to datatime
df['tpep_pickup_datetime'] = pd.to_datetime(df['tpep_pickup_datetime'])
df['tpep_dropoff_datetime'] = pd.to_datetime(df['tpep_dropoff_datetime'])
df.info()
#removing the duplicates in dataframes and transforming them into the table
datetime_dim = df[['tpep_pickup_datetime', 'tpep_dropoff_datetime']].drop_duplicates().reset_index(drop=True)
datetime_dim['tpep_pickup_datetime'] = datetime_dim['tpep_pickup_datetime']
datetime_dim['pick_hour'] = datetime_dim['tpep_pickup_datetime'].dt.hour
datetime_dim['pick_day'] = datetime_dim['tpep_pickup_datetime'].dt.day
datetime_dim['pick_month'] = datetime_dim['tpep_pickup_datetime'].dt.month
datetime_dim['pick_year'] = datetime_dim['tpep_pickup_datetime'].dt.year
datetime_dim['pick_weekday'] = datetime_dim['tpep_pickup_datetime'].dt.weekday

datetime_dim['tpep_dropoff_datetime'] = datetime_dim['tpep_dropoff_datetime']
datetime_dim['drop_hour'] = datetime_dim['tpep_dropoff_datetime'].dt.hour
datetime_dim['drop_day'] = datetime_dim['tpep_dropoff_datetime'].dt.day
datetime_dim['drop_month'] = datetime_dim['tpep_dropoff_datetime'].dt.month
datetime_dim['drop_year'] = datetime_dim['tpep_dropoff_datetime'].dt.year
datetime_dim['drop_weekday'] = datetime_dim['tpep_dropoff_datetime'].dt.weekday
datetime_dim
#check in the index's
datetime_dim.index
# adding primary key(datetime_id) to the datetime_dim table
datetime_dim['datetime_id'] = datetime_dim.index

datetime_dim
#arranging in proper order
    datetime_dim = datetime_dim[['datetime_id', 'tpep_pickup_datetime', 'pick_hour', 'pick_day', 'pick_month',
    'pick_year', 'pick_weekday',
                            'tpep_dropoff_datetime', 'drop_hour', 'drop_day', 'drop_month', 'drop_year',
    'drop_weekday']]
    datetime_dim.head()
    # creating passenger_count dim table
    passenger_count_dim = df[['passenger_count']].reset_index(drop=True) #droping duplicates
    passenger_count_dim['passenger_count_id'] = passenger_count_dim.index
    passenger_count_dim = passenger_count_dim[['passenger_count_id','passenger_count']]
    # creating trip_distance dim table
    trip_distance_dim = df[['trip_distance']].reset_index(drop=True) #droping duplicates
    trip_distance_dim['trip_distance_id'] = trip_distance_dim.index
    trip_distance_dim = trip_distance_dim[['trip_distance_id','trip_distance']]
    passenger_count_dim.head()
```

```
trip_distance_dim.head()
# created dictionary based on the yellow taxi dict
rate_code_type = {
   1:"Standard rate",
   2:"JFK",
   3:"Newark",
   4:"Nassau or Westchester",
   5:"Negotiated fare",
   6:"Group ride"
}
```

structure.py

```
# creating the rate_code dim table
rate_code_dim = df[['RatecodeID']].reset_index(drop=True)
rate_code_dim['rate_code_id'] = rate_code_dim.index
rate_code_dim['rate_code_name'] = rate_code_dim['RatecodeID'].map(rate_code_type)
rate_code_dim = rate_code_dim[['rate_code_id','RatecodeID','rate_code_name']]
rate_code_dim.head()
#creating pickup_location dim table
pickup_location_dim = df[['pickup_longitude', 'pickup_latitude']].reset_index(drop=True)
pickup_location_dim['pickup_location_id'] = pickup_location_dim.index
pickup_location_dim = pickup_location_dim[['pickup_location_id','pickup_latitude','pickup_longitude']]
dropoff_location_dim = df[['dropoff_longitude', 'dropoff_latitude']].reset_index(drop=True)
dropoff_location_dim['dropoff_location_id'] = dropoff_location_dim.index
dropoff_location_dim = dropoff_location_dim[['dropoff_location_id','dropoff_latitude','dropoff_longitude']]
payment_type_name = {
   1:"Credit card",
   2:"Cash",
   3:"No charge",
   4:"Dispute",
   5:"Unknown",
   6:"Voided trip"
}
payment_type_dim = df[['payment_type']].reset_index(drop=True)
payment_type_dim['payment_type_id'] = payment_type_dim.index
payment_type_dim['payment_type_name'] = payment_type_dim['payment_type'].map(payment_type_name)
payment_type_dim = payment_type_dim[['payment_type_id','payment_type','payment_type_name']]
payment_type_dim.head()
fact_table = df.merge(passenger_count_dim, left_on='trip_id', right_on='passenger_count_id') \
           .merge(trip_distance_dim, left_on='trip_id', right_on='trip_distance_id') \
           .merge(rate_code_dim, left_on='trip_id', right_on='rate_code_id') \
             .merge(pickup_location_dim, left_on='trip_id', right_on='pickup_location_id') \
             .merge(dropoff_location_dim, left_on='trip_id', right_on='dropoff_location_id')\
             .merge(datetime_dim, left_on='trip_id', right_on='datetime_id') \
             .merge(payment_type_dim, left_on='trip_id', right_on='payment_type_id') \
             [['trip_id','VendorID', 'datetime_id', 'passenger_count_id',
               'trip_distance_id', 'rate_code_id', 'store_and_fwd_flag', 'pickup_location_id', 'dropoff_location_id',
               'payment_type_id', 'fare_amount', 'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
               'improvement_surcharge', 'total_amount']]
```

query.sql

```
CREATE OR REPLACE TABLE
`pelagic-bastion-417411.uber_data_engineering.tbl_analytics1` AS (

SELECT
f.VendorID, d.tpep_pickup_datetime, d.tpep_dropoff_datetime, p.passenger_count, t.trip_distance,
r.rate_code_name, pick.pickup_latitude, pick.pickup_longitude, drop.dropoff_latitude,
drop.dropoff_longitude, pay.payment_type_name, f.fare_amount, f.extra, f.mta_tax, f.tip_amount,
f.tolls_amount, f.improvement_surcharge, f.total_amount

FROM
 `pelagic-bastion-417411.uber_data_engineering.fact_table` f JOIN
`pelagic-bastion-417411.uber_data_engineering.datetime_dim` d ON f.datetime_id=d.datetime_id

JOIN `pelagic-bastion-417411.uber_data_engineering.passenger_count_dim` p ON
P.passenger_count_id=f.passenger_count_id

JOIN `pelagic-bastion-417411.uber_data_engineering.trip_distance_dim` t ON
t.trip_distance_id=f.trip_distance_id

 JOIN `pelagic-bastion-417411.uber_data_engineering.rate_code_dim` r ON
r.rate_code_id=f.rate_code_id JOIN
`pelagic-bastion-417411.uber_data_engineering.pickup_location_dim` pick ON
pick.pickup_location_id=f.pickup_location_id

JOIN `pelagic-bastion-417411.uber_data_engineering.dropoff_location_dim` drop ON
drop.dropoff_location_id=f.dropoff_location_id

JOIN `pelagic-bastion-417411.uber_data_engineering.payment_type_dim` pay ON
pay.payment_type_id=f.payment_type_id )
```

CHAPTER-7
OUTPUT SCREENSHOTS

7. Output Screenshots

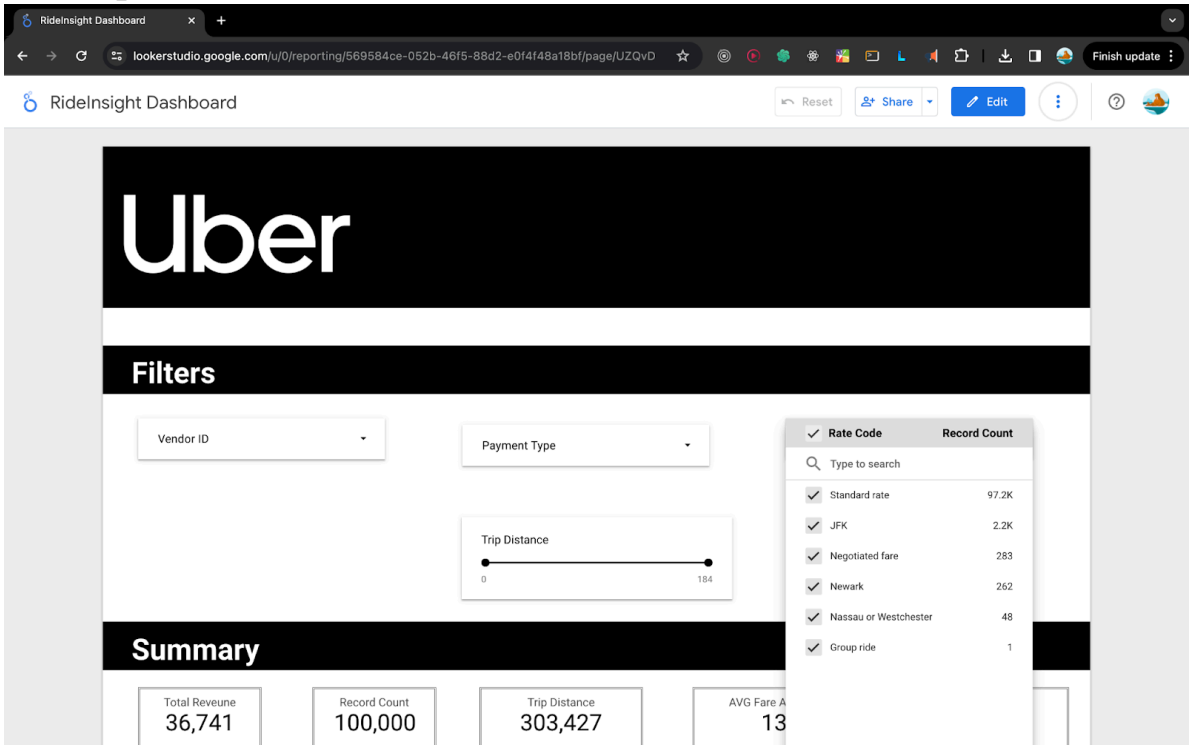

Figure 7.1 Filter by Vendor ID

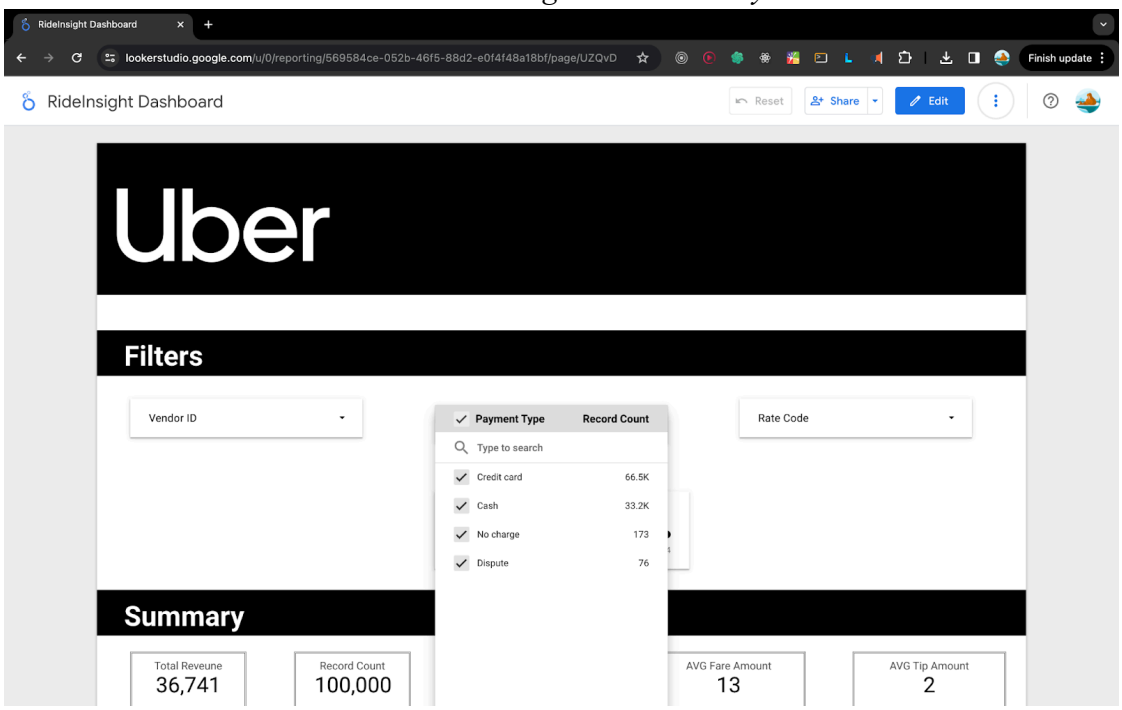

Figure 7.2 Customer payment method

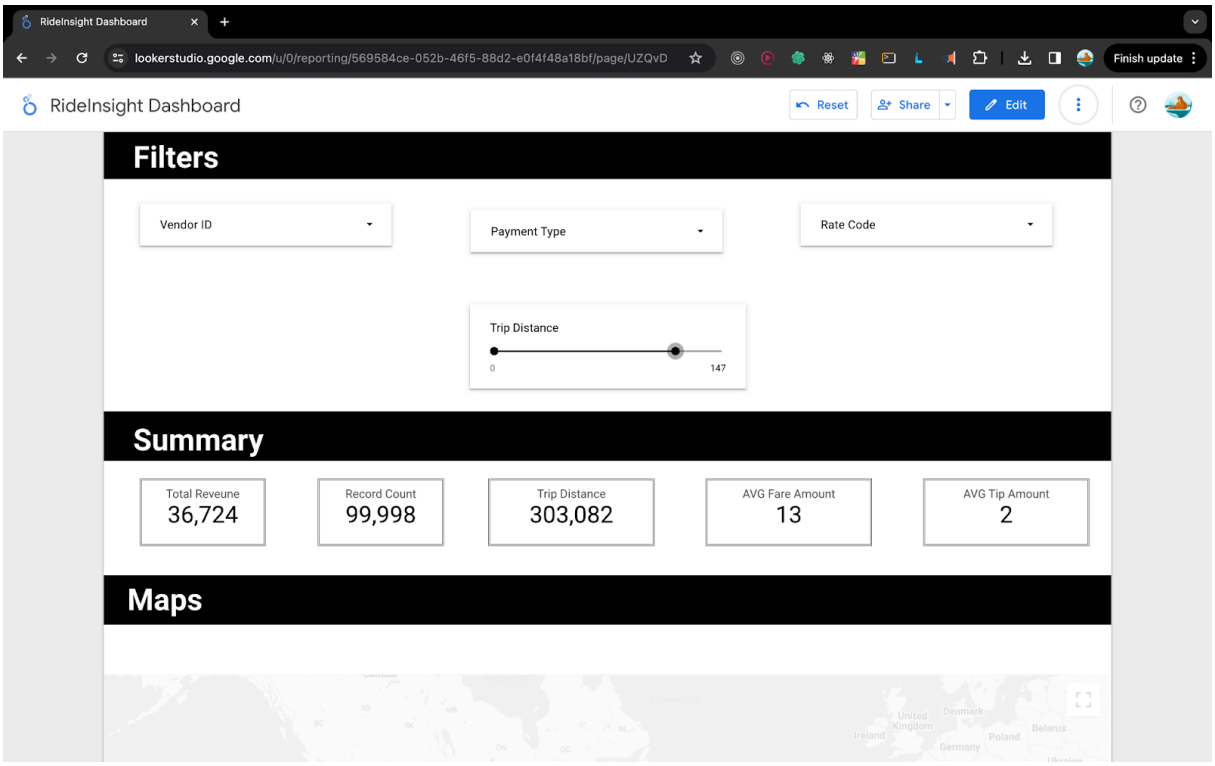

Figure 7.3 Filter by Trip Distance

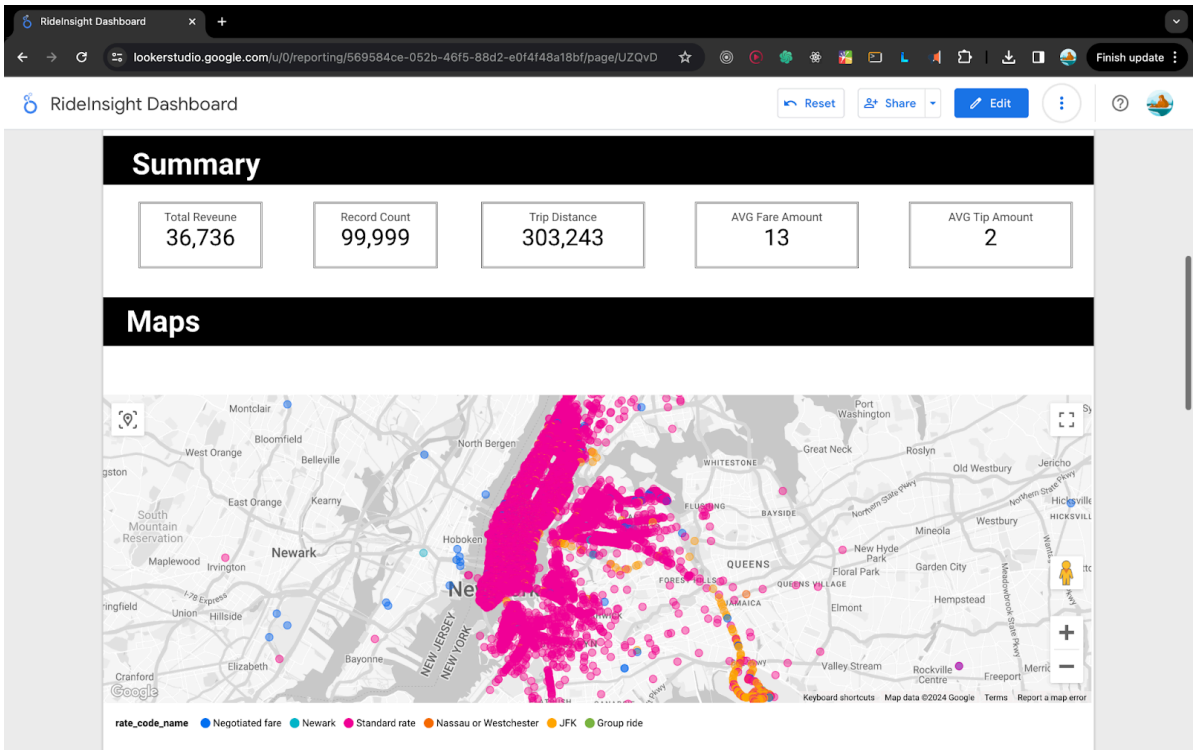

Figure 7.4 Pointing on Google Maps

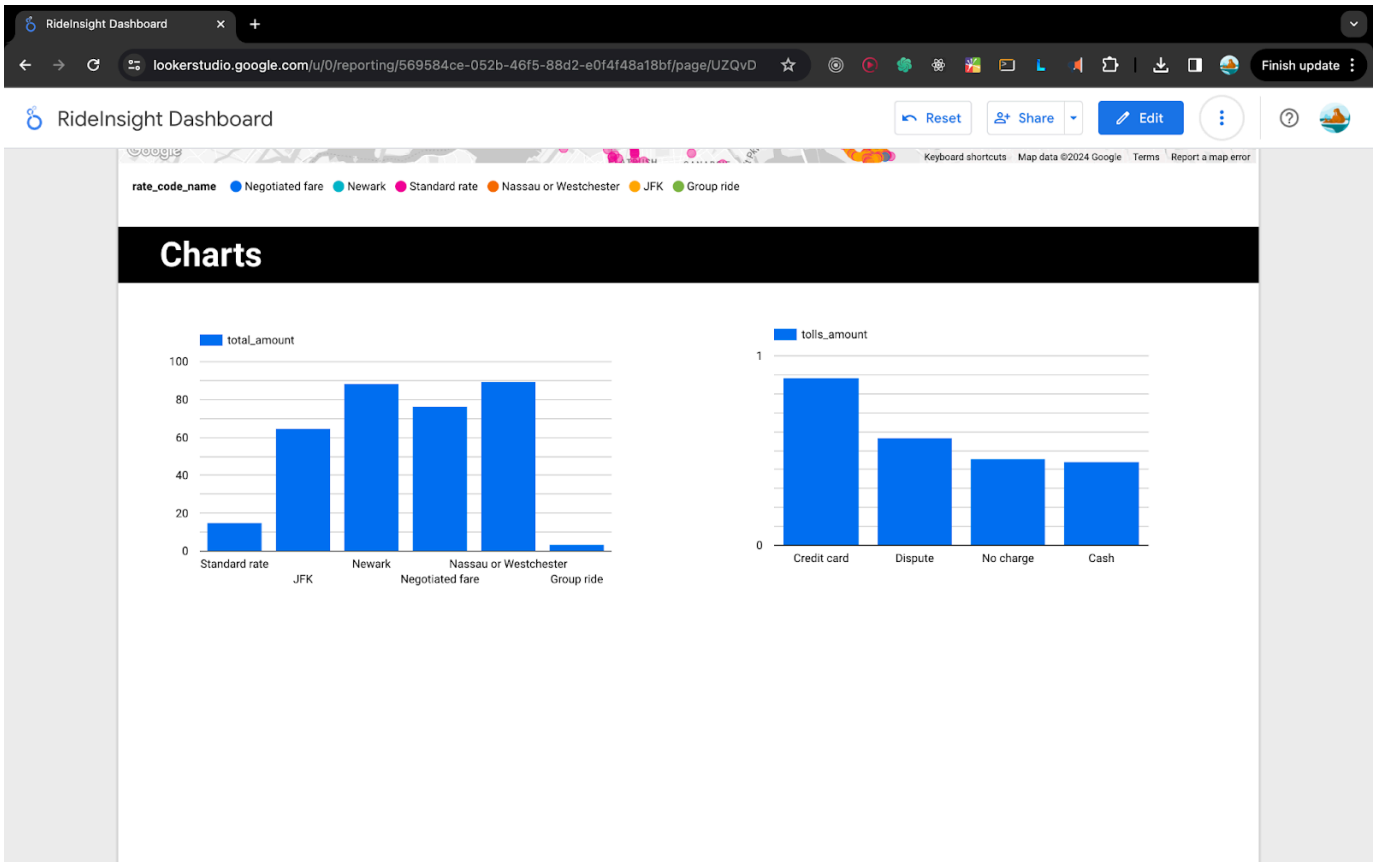

Figure 7.5 Different payment analysis

CHAPTER-8
TESTING

8.1. Introduction to Testing

Testing is a procedure that identifies program errors. It is the primary quality metric used in software development. During testing, the program is run under a set of conditions known as test cases, and the output is analyzed to see if it is operating as expected. The process of executing software to validate its functionality and correctness is known as software testing. The process of running a program to identify an error. An excellent test case has a high likelihood of discovering an as-yet-undiscovered fault. A successful test reveals a previously unknown mistake. Software testing is typically done for two reasons:

• Detection of flaws
• Estimation of reliability

8.2. Types of Testing:

To ensure that the system is error-free, the following tiers of testing methodologies are used at various stages of software development:

1. Unit testing: This is performed on individual models as they are finalized and made executable. It is solely limited to the designer's specifications. Each module can be tested using one of the two methods listed below:

Black Box Testing: With this method, some test cases are created as input conditions that fully execute all the program's functional requirements. This testing was used to identify faults in the following areas:

a) Functions that are incorrect or missing.
b) Errors in the interface.
c) Data structure errors or access to an external database.
d) Mistakes in performance.
e) Errors in initialization and termination.

Only the output is examined for correctness during this testing. The data's logical flow is not examined.

White Box testing: In this method, test cases are built based on the logic of each module by sketching flow diagrams of that module's logic, and logical judgments are tested on all situations. It was used to create test cases in the following situations:

a) Ensure that all independent pathways are followed.
b) Carry out all reasonable decisions on both the truthful and false sides.
c) Run all loops inside their operational constraints and boundaries.
d) Test internal data structures for correctness.

2. Integration Testing: Integration testing guarantees that software and subsystems work in concert. It tests the interfaces of all modules to ensure that they work properly when combined.

3. Acceptance Testing: This is a type of pre-delivery testing in which the entire system is evaluated on real-world data at the client's location to identify faults.

4. Validation: The system has been successfully tested and implemented, ensuring that all of the requirements mentioned in the software requirements specification are properly met. In the event of incorrect input, the associated error messages are presented.

Compiling Test: Doing our stress testing early on was a smart idea because it allowed us time to fix some of the unforeseen deadlocks and stability issues that only appeared when components were exposed to extremely high transaction volumes.

Execution Test: The software was loaded and run successfully. There were no execution errors because of solid programming.

8.3 Sample Test Cases

Test Case 1: Testing by single Vendor ID.

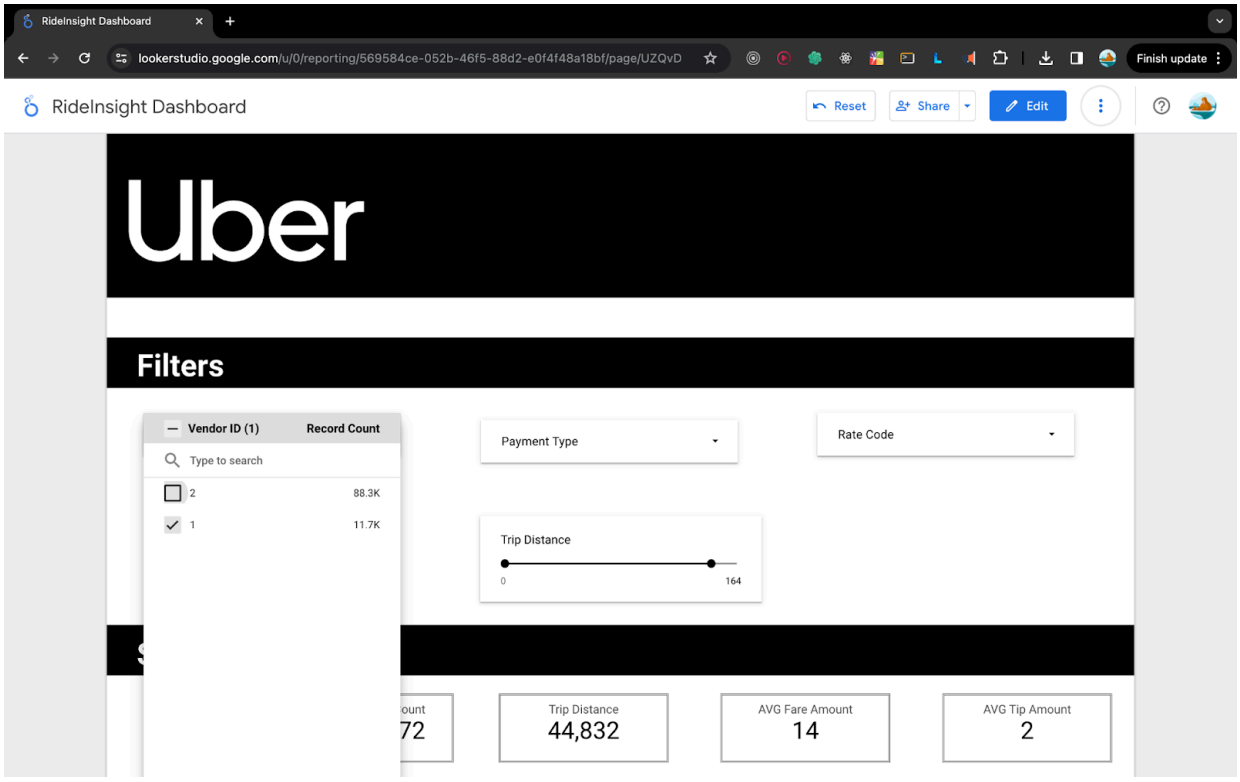

Figure 8.1 Testing by vendor Id

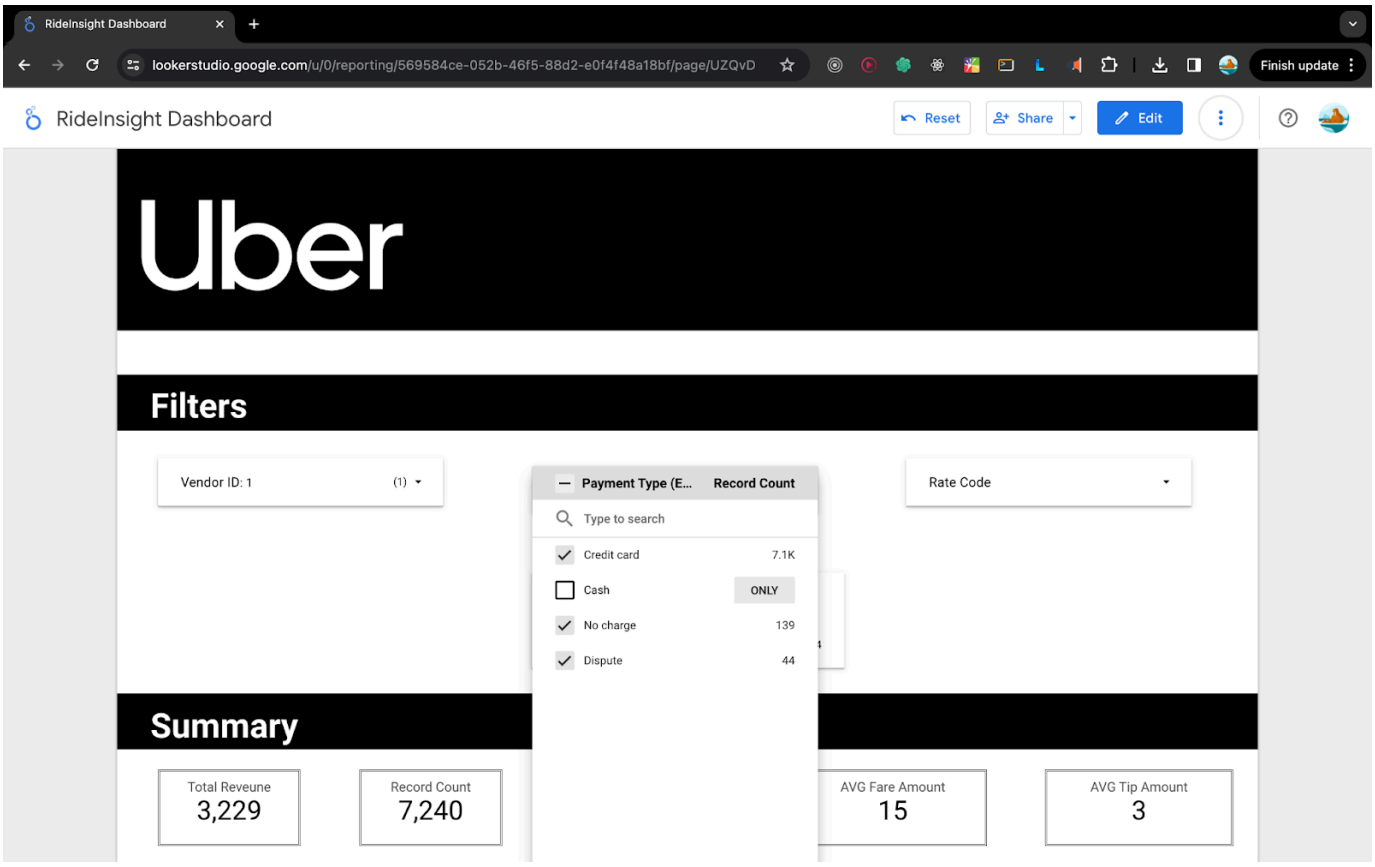

Figure 8.2 By Different Payment Methods

Test Case - 2 : Testing by Different Payment Methods

Test Case - 3: Testing by Different Rate Codes

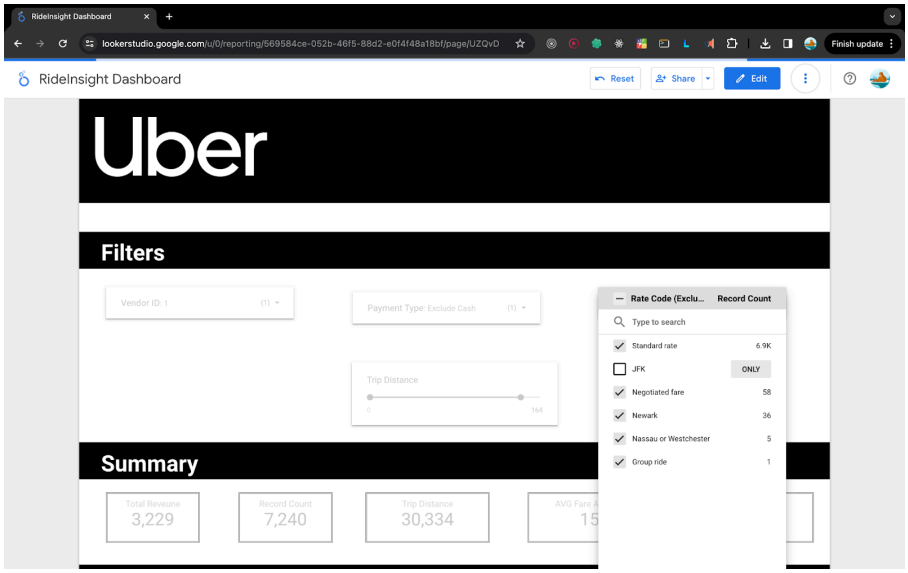

Figure 8.3 The results of different rate code

CHAPTER-9
CONCLUSION
&
FUTURE ENHANCEMENTS

This experiment has unveiled the intricate web that connects driver behavior and user happiness within the realm of ride-hailing services. By meticulously combing through the data, we've unearthed crucial insights that illuminate how customer ratings, driver acceptance rates, and cancellation rates all have a significant impact on the quality of the service experienced by riders. These findings translate into a practical roadmap for ride-hailing companies to elevate driver performance and, consequently, enhance the overall customer experience.

One key takeaway lies in the undeniable link between driver acceptance rates and user satisfaction. When drivers consistently accept ride requests, particularly during peak hours or in less desirable locations, wait times decrease for passengers. This translates to a more convenient and positive experience for the user. Conversely, low acceptance rates lead to frustration for riders, potentially causing them to switch to alternative transportation options.

The data also sheds light on the significance of cancellation rates. Cancellations by drivers, especially after accepting a ride, can leave passengers stranded and disrupt their schedules. Similarly, frequent cancellations by users can negatively impact drivers' earnings and overall morale. By implementing measures to discourage unnecessary cancellations on both ends, ride-hailing services can create a more reliable and predictable experience for all parties involved.

Customer ratings, unsurprisingly, emerge as a powerful indicator of driver behavior and service quality. High ratings often reflect a smooth, safe, and courteous ride. Conversely, low ratings can point towards aggressive driving, unsafe maneuvers, or a lack of professionalism by the driver. By leveraging this feedback loop, ride-hailing companies can incentivize positive behavior through bonus programs or recognition systems, while also implementing targeted training programs to address areas highlighted by poor ratings.

This emphasis on data-driven insights unlocks significant opportunities for ride-hailing companies to achieve sustainable business growth. By prioritizing driver performance and user satisfaction, companies can cultivate a loyal customer base. This translates to increased ride volume, higher retention rates, and ultimately, a stronger bottom line.

Looking ahead, the foundation laid by this research will pave the way for future advancements in ride-hailing systems. By continuously analyzing user data and evolving alongside changing consumer needs, ride-hailing services can adapt their offerings to ensure optimal efficiency and cater to a broader range of preferences. This might include features like driver profiles that showcase positive ratings and preferred routes, or dynamic pricing models that take into account real-time demand and traffic conditions.

Ultimately, the success of ride-hailing services hinges on creating a seamless and satisfying experience for both riders and drivers. This research serves as a valuable stepping stone for achieving that goal, fostering a mutually beneficial ecosystem where everyone wins. By harnessing the power of data and implementing strategic changes, ride-hailing companies can refine their operations to deliver a superior service that keeps users coming back for more.

CHAPTER-10
REFERENCES

10. References

[1] R. J. Sandusky, “Computational provenance: Dataone and implications for cultural heritage institutions,” in 2016 IEEE International Conference on Big Data (Big Data). IEEE, 2016, pp. 3266–3271.

[2] J. P. Cohn, “Dataone opens doors to scientists across disciplines,” 2012.

[3] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Wurthwein ¨ et al., “The open science grid,” in Journal of Physics: Conference Series, vol. 78, no. 1. IOP Publishing, 2007, p. 012057.

[4] D. Medvedev, G. Lemson, and M. Rippin, “Sciserver compute: Bringing analysis close to the data,” in Proceedings of the 28th international conference on scientific and statistical database management, 2016, pp. 1–4.

[5] S. Gesing, J. Kruger, R. Grunzke, S. Herres-Pawlis, and A. Hoffmann, ¨ “Using science gateways for bridging the differences between research infrastructures,” Journal of Grid Computing, vol. 14, no. 4, pp. 545–557, 2016.

[6] I. Foster, “Globus online: Accelerating and democratizing science through cloud-based services,” IEEE Internet Computing, vol. 15, no. 3, pp. 70–73, 2011.

[7] S. Gugnani, C. Blanco, T. Kiss, and G. Terstyanszky, “Extending science gateway frameworks to support big data applications in the cloud,” Journal of Grid Computing, vol. 14, no. 4, pp. 589–601, 2016.

[8] A. Talukder, M. Elshambakey, S. Wadkar, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, “Vifi: Virtual information fabric infrastructure for data-driven discoveries from distributed earth science data,” in 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2017, pp. 1–8.

[9] P. Pirolli and S. Card, “The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis,” in Proceedings of international conference on intelligence analysis, vol. 5. McLean, VA, USA, 2005, pp. 2–4.

[10] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, “Knowledge generation model for visual analytics,” IEEE transactions on visualization and computer graphics, vol. 20, no. 12, pp. 1604–1613, 2014.

[11] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, “Xsede: Accelerating scientific discovery,” Computing in Science Engineering, vol. 16, no. 5, pp. 62–74, Sep. 2014.

[12] R. J. Sandusky, “Computational provenance: Dataone and implications for cultural heritage institutions,” in IEEE International Conference on Big Data (Big Data), Dec 2016, pp. 3266–3271.

[13] S. Shahand, M. M. Jaghoori, A. Benabdelkader, J. L. Font-Calvo, J. Huguet, M. W. Caan, A. H. van Kampen, and S. D. Olabarriaga, Computational Neuroscience Gateway: A Science Gateway Based on the WS-PGRADE/gUSE. Cham: Springer International Publishing, 2014,

[14] M. Elshambakey, M. Khalefa, W. J. Tolone, S. D. Bhattacharjee, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, “Towards a distributed infrastructure for data-driven discoveries &

analysis," in 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017, pp. 4738–4740.

[15] C. Chokwitthaya, Y. Zhu, R. Dibiano, and S. Mukhopadhyay, "Combining context-aware design-specific data and building performance models to improve building performance predictions during design," Automation in construction, vol. 107, p. 102917, 2019.

[16] O. T. Karaguzel, M. Elshambakey, Y. Zhu, T. Hong, W. J. Tolone, S. Das Bhattacharjee, I. Cho, W. Dou, H. Wang, S. Lu et al., "Open computing infrastructure for sharing data analytics to support building energy simulations," Journal of Computing in Civil Engineering, vol. 33, no. 6, p. 04019037, 2019.

[1] R. J. Sandusky, "Computational provenance: Dataone and implications for cultural heritage institutions," in 2016 IEEE International Conference on Big Data (Big Data). IEEE, 2016, pp. 3266–3271.

[2] J. P. Cohn, "Dataone opens doors to scientists across disciplines," 2012.

[3] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Wurthwein ¨ et al., "The open science grid," in Journal of Physics: Conference Series, vol. 78, no. 1. IOP Publishing, 2007, p. 012057.

[4] D. Medvedev, G. Lemson, and M. Rippin, "Sciserver compute: Bringing analysis close to the data," in Proceedings of the 28th international conference on scientific and statistical database management, 2016, pp. 1–4.

[5] S. Gesing, J. Kruger, R. Grunzke, S. Herres-Pawlis, and A. Hoffmann, ¨ "Using science gateways for bridging the differences between research infrastructures," Journal of Grid Computing, vol. 14, no. 4, pp. 545–557, 2016.

[6] I. Foster, "Globus online: Accelerating and democratizing science through cloud-based services," IEEE Internet Computing, vol. 15, no. 3, pp. 70–73, 2011.

[7] S. Gugnani, C. Blanco, T. Kiss, and G. Terstyanszky, "Extending science gateway frameworks to support big data applications in the cloud," Journal of Grid Computing, vol. 14, no. 4, pp. 589–601, 2016.

[8] A. Talukder, M. Elshambakey, S. Wadkar, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, "Vifi: Virtual information fabric infrastructure for data-driven discoveries from distributed earth science data," in 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2017, pp. 1–8.

[9] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in Proceedings of international conference on intelligence analysis, vol. 5. McLean, VA, USA, 2005, pp. 2–4.

[10] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," IEEE transactions on visualization and computer graphics, vol. 20, no. 12, pp. 1604–1613, 2014.

[11] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," Computing in Science Engineering, vol. 16, no. 5, pp. 62–74, Sep. 2014.

[12] R. J. Sandusky, "Computational provenance: Dataone and implications for cultural heritage institutions," in IEEE International Conference on Big Data (Big Data), Dec 2016, pp. 3266–3271.

[13] S. Shahand, M. M. Jaghoori, A. Benabdelkader, J. L. Font-Calvo, J. Huguet, M. W. Caan, A. H. van

Kampen, and S. D. Olabarriaga, Computational Neuroscience Gateway: A Science Gateway Based on the WS-PGRADE/gUSE. Cham: Springer International Publishing, 2014,

[14] M. Elshambakey, M. Khalefa, W. J. Tolone, S. D. Bhattacharjee, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, "Towards a distributed infrastructure for data-driven discoveries & analysis," in 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017, pp. 4738–4740.

[15] C. Chokwitthaya, Y. Zhu, R. Dibiano, and S. Mukhopadhyay, "Combining context-aware design-specific data and building performance models to improve building performance predictions during design," Automation in construction, vol. 107, p. 102917, 2019.

[16] O. T. Karaguzel, M. Elshambakey, Y. Zhu, T. Hong, W. J. Tolone, S. Das Bhattacharjee, I. Cho, W. Dou, H. Wang, S. Lu et al., "Open computing infrastructure for sharing data analytics to support building energy simulations," Journal of Computing in Civil Engineering, vol. 33, no. 6, p. 04019037, 2019.

[17] R. Zhang and O. T. Karaguzel, "Development and calibration of reduced order building energy models by coupling with high-order simulations," Global journal of advanced engineering technologies and sciences, vol. 7, no. 2, 2020.

[18] W. J Tolone, "Application of the virtual information fabric infrastructure (vifi) to building performance simulations," Current Trends in Civil & Structural Engineering, vol. 4, no. 2, 2019.

[19] D. Merkel, "Docker: Lightweight linux containers for consistent development and deployment," Linux J., vol. 2014, no. 239, Mar. 2014.

[20] I. Miell and A. H. Sayers, Docker in Practice, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2016.

[21] https://nifi.apache.org/.

[22] https://docs.docker.com/engine/swarm/

[23] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johhson, "Visualization of uncertainty and ensemble data: Exploration of climate modeling and weather forecast data with integrated visus-cdat systems," in Journal of Physics: Conference Series, vol. 180, no. 1. IOP Publishing, 2009, p. 012089.

[24] A. Matˇ acut¸ ˇ a and C. Popa, "Big data analytics: Analysis of features ˇ and performance of big data ingestion tools," Informatica Economica, vol. 22, no. 2, pp. 25–34, 2018.

[25] P. Kacsuk, Science gateways for distributed computing infrastructures: Development framework and exploitation by scientific user communities. Springer International Publishing, 8 2014.

[26] Visual Analytics Frameworks (IEEE Xplore):
The paper titled "Visual Analytics Frameworks for Distributed Data Analysis" (https://ieeexplore.ieee.org/document/9671768) elucidates the relevance of visual analytics in distributed data analysis systems.

[27]The article "Modern Data Engineering with Mage: Empowering Efficient Data(https://www.analyticsvidhya.com/blog/2023/06/modern-data-engineering-with-mage-empowering-efficient-data-processing/)

[28] https://cloud.google.com/docs/tutorials: gcp cloud documentation

PAPER PUBLICATION

Journal Title (in English Language)	Industrial Engineering Journal (print only) | (Current Table of Content)
Publication Language	English
Publisher	Indian Institution of Industrial Engineering
ISSN	0970-2555
E-ISSN	0970-2555
Discipline	Science
Subject	Engineering (all)
Focus Subject	Industrial and Manufacturing Engineering
UGC-CARE coverage years	from June-2019 to Present