INDIAN INSTITUTE OF TECHNOLOGY JODHPUR

NH 62 Nagaur Road, Karwar, Jodhpur, Rajasthan

Trimester 3 – Post Graduate Diploma in Data Engineering

Big Data Management

Project Report
On
Food Delivery Management System

Submitted By:

Aneerban Chowdhury, G23AI2059

Vanshika Gupta, G23AI2050

Shikha Soni, G23AI2075

Food Delivery Management System

Abstract

The Food Delivery Management System project was undertaken to design and implement an end-to-end solution for managing food delivery data at scale. This report provides a comprehensive account of the system's architecture, development process, technical challenges, and the innovations applied. The system leverages Python, Google Cloud Platform (GCP) services such as BigQuery and Google Cloud Storage (GCS), and scikit-learn for machine learning tasks. This report also discusses future enhancements, alternative approaches, and the broader implications of this system.

Introduction

Food delivery services have seen unprecedented growth, leading to an exponential increase in the volume of data generated daily. This project aims to address the challenges of managing and analysing this data efficiently using modern tools and frameworks. The goal is to design a scalable system for storing, processing, analysing, and visualizing data while incorporating predictive analytics for business insights.

Problem Statement

1. **Data Management**: Efficiently handling and processing large datasets encompassing restaurants, customers, and orders. Managing structured and unstructured data in a way that ensures consistency, accuracy, and security. Addressing the challenges of ingestion, storage, and retrieval of high-volume data while maintaining optimal performance.
2. **Analytics**: Deriving actionable insights by analyzing data trends, such as customer purchasing behavior, popular cuisines, and peak ordering times. The goal is to leverage data analytics to guide decision-making processes and improve business strategies for stakeholders.
3. **Prediction**: Building and deploying machine learning models to forecast order volumes accurately. This includes identifying patterns from historical data and predicting future trends to assist in resource allocation and operational planning.
4. **Accessibility**: Designing a user-friendly frontend interface that facilitates seamless access to key insights and visualizations. Ensuring the interface is intuitive and provides clear, actionable information to non-technical users.
5. **Scalability**: Developing a system architecture capable of scaling with increasing data size and user traffic. Incorporating strategies to handle larger datasets and concurrent users without compromising system performance or reliability

Objectives

1. **Data Storage**: Utilize Google Cloud Storage (GCS) and BigQuery for efficient storage and management of both structured and unstructured data. Implement schemas for relational datasets, ensuring easy query ability and integration with analytics tools.
2. **Data Visualization**: Create comprehensive and visually appealing reports that represent key metrics and trends. These visualizations should be designed for stakeholders to understand business performance and customer behavior at a glance.
3. **Machine Learning**: Implement predictive models using state-of-the-art machine learning techniques to forecast order trends. Employ linear regression and other algorithms to derive meaningful predictions from historical data.
4. **Frontend**: Develop a responsive and interactive web interface for end-users to interact with the system. This includes features for viewing order summaries, exploring visualized data, and accessing predictive insights in an easily digestible format.

Methodology

System Architecture

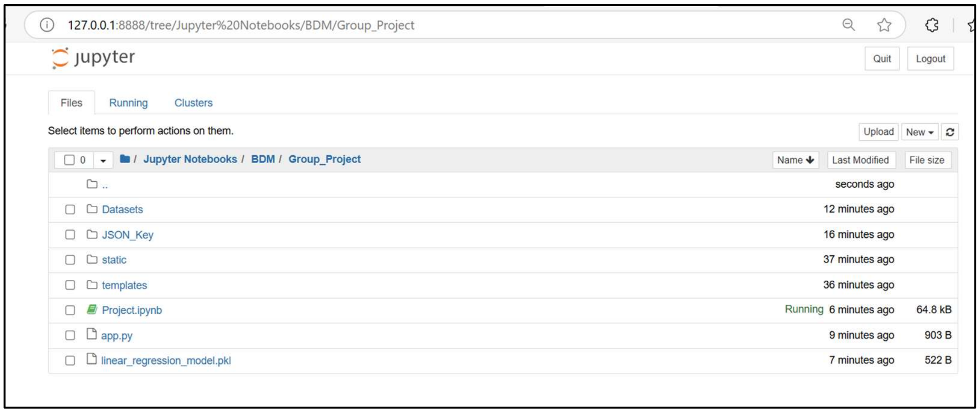
The project is built using a modular architecture that integrates the following components:

1. **Data Storage:** Google Cloud Storage and BigQuery for structured and unstructured data.
2. **Data Processing:** Python scripts for data cleaning, transformation, and loading (ETL).
3. **Visualization:** Matplotlib and Seaborn for charts and graphs.
4. **Machine Learning:** Scikit-learn for predictive analytics.
5. **Frontend:** Flask for web interface development.

Implementation Steps

Step 1: Environment Setup

- Installed Python and essential libraries like google-cloud-bigquery, google-cloud-storage, and scikit-learn.
- Set up GCP with appropriate permissions and billing.
- Configured Jupyter Notebook for development and debugging



127.0.0.1:8888/tree/Jupyter%20Notebooks/BDM/Group_Project

Select items to perform actions on them.

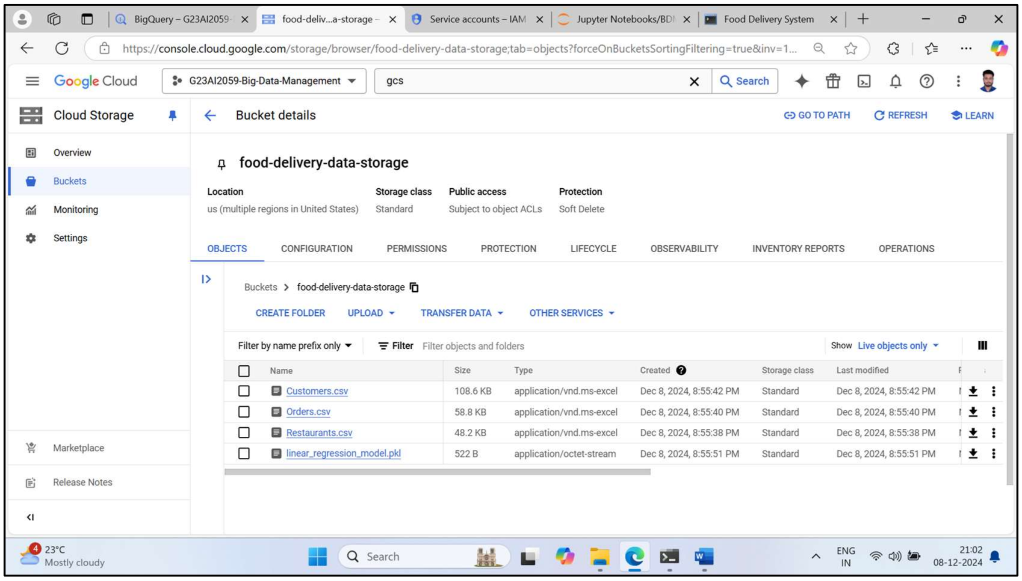
□ 0 / Jupyter Notebooks / BDM / Group_Project

Name		Last Modified	File size
..		seconds ago	
□ Datasets		12 minutes ago	
□ JSON_Key		16 minutes ago	
□ static		37 minutes ago	
□ templates		36 minutes ago	
□ Project.ipynb	Running	6 minutes ago	64.8 kB
□ app.py		9 minutes ago	903 B
□ linear_regression_model.pkl		7 minutes ago	522 B

Step 2: Data Design

Designed three main datasets:

- **Restaurants:** Contains details like RestaurantID, Name, Location, CuisineType, and Rating.
- **Customers:** Includes CustomerID, Name, Email, Phone, and Address.
- **Orders:** Records OrderID, CustomerID, RestaurantID, OrderDate, OrderStatus, and OrderAmount.



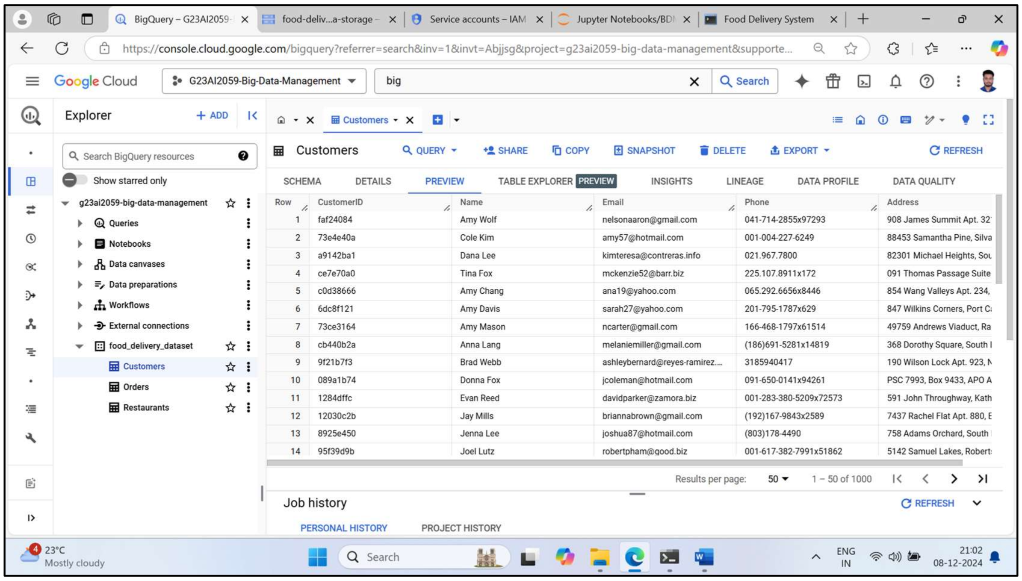
food-delivery-data-storage

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Subject to object ACLs	Soft Delete

Name	Size	Type	Created	Storage class	Last modified
Customers.csv	108.6 KB	application/vnd.ms-excel	Dec 8, 2024, 8:55:42 PM	Standard	Dec 8, 2024, 8:55:42 PM
Orders.csv	58.8 KB	application/vnd.ms-excel	Dec 8, 2024, 8:55:40 PM	Standard	Dec 8, 2024, 8:55:40 PM
Restaurants.csv	48.2 KB	application/vnd.ms-excel	Dec 8, 2024, 8:55:38 PM	Standard	Dec 8, 2024, 8:55:38 PM
linear_regression_model.pkl	522 B	application/octet-stream	Dec 8, 2024, 8:55:51 PM	Standard	Dec 8, 2024, 8:55:51 PM

Step 3: Data Capturing and Storage

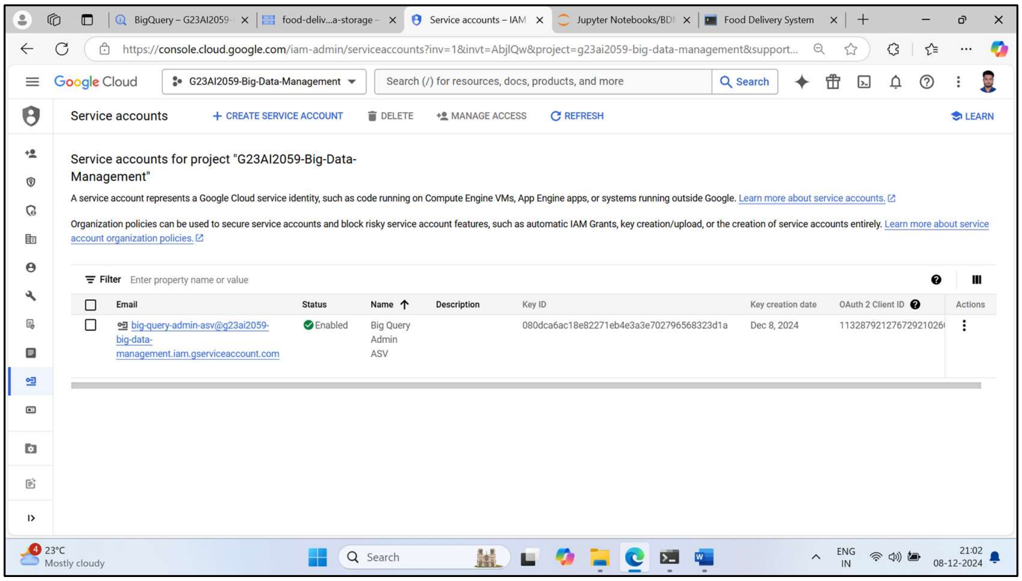
- Capture data from Kaggle or synthetic data sources 1,000 rows for each table.
- Stored CSV files in GCS and loaded them into BigQuery for querying.



Row	CustomerID	Name	Email	Phone	Address
1	faf24084	Amy Wolf	nelsonaaron@gmail.com	041-714-2855x97293	908 James Summit Apt. 32
2	73e4e40a	Cole Kim	amy57@hotmail.com	001-004-227-6249	88453 Samantha Pine, Silva
3	a9142ba1	Dana Lee	kimteresa@contreras.info	021.967.7800	82301 Michael Heights, Sou
4	ce7e70a0	Tina Fox	mckenzie52@barr.biz	225.107.8911x172	091 Thomas Passage Suite
5	c0d38666	Amy Chang	ana19@yahoo.com	065.292.6656x8446	854 Wang Valleys Apt. 234,
6	6dc8f121	Amy Davis	sarah27@yahoo.com	201-795-1787x629	847 Wilkins Corners, Port C
7	73ce3164	Amy Mason	ncarter@gmail.com	166-468-1797x61514	49759 Andrews Viaduct, Ra
8	cb440b2a	Anna Lang	melaniemiller@gmail.com	(186)691-5281x14819	368 Dorothy Square, South
9	9f21b7f3	Brad Webb	ashleybernard@reyes-ramirez....	3185940417	190 Wilson Lock Apt. 923, N
10	089a1b74	Donna Fox	jcoleman@hotmail.com	091-650-0141x94261	PSC 7993, Box 9433, APO A
11	1284dffc	Evan Reed	davidparker@zamora.biz	001-283-380-5209x72573	591 John Throughway, Kath
12	12030c2b	Jay Mills	briannabrown@gmail.com	(192)167-9843x2589	7437 Rachel Flat Apt. 880, E
13	8925e450	Jenna Lee	joshua87@hotmail.com	(803)178-4490	758 Adams Orchard, South
14	95f39d9b	Joel Lutz	robertpham@good.biz	001-617-382-7991x51862	5142 Samuel Lakes, Robert

Step 4: Data Analytics

- Wrote SQL queries to analyze trends and summarize data.
- Explored insights like high-performing restaurants, customer spending patterns, and peak order times.



https://console.cloud.google.com/iam-admin/serviceaccounts?inv=1&invt=AbjlQw&project=g23ai2059-big-data-management&support...

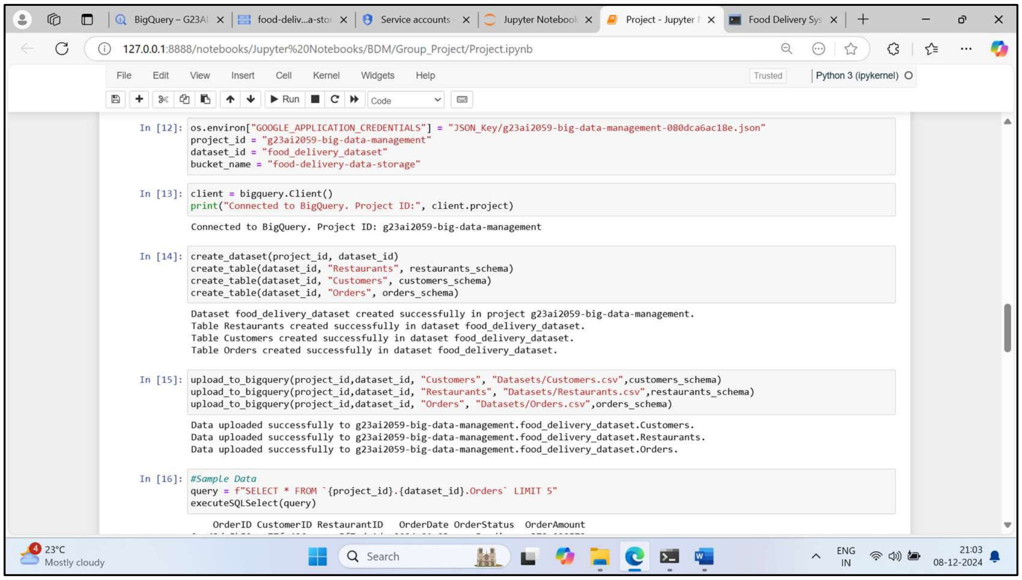
Service accounts

Service accounts for project "G23AI2059-Big-Data-Management"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

□	Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
□	big-query-admin-asv@g23ai2059-big-data-management.iam.gserviceaccount.com	Enabled	Big Query Admin ASV		080dca6ac18e82271eb4e3a3e702796568323d1a	Dec 8, 2024	11328792127672921026	⋮

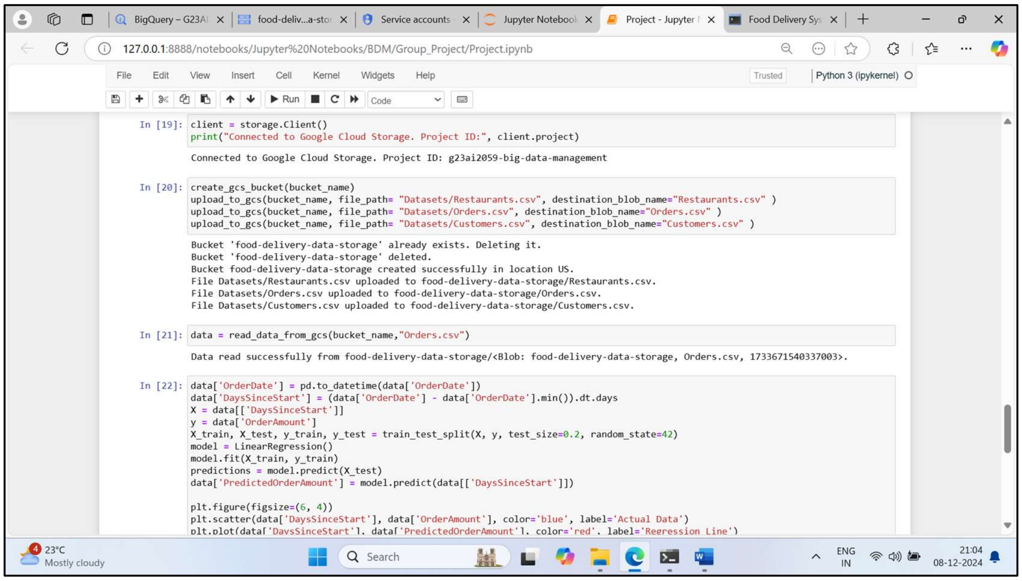


127.0.0.1:8888/notebooks/Jupyter%20Notebooks/BDM/Group_Project/Project.ipynb

```
In [12]: os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "JSON_Key/g23ai2059-big-data-management-080dca6ac18e.json"
         project_id = "g23ai2059-big-data-management"
         dataset_id = "food_delivery_dataset"
         bucket_name = "food-delivery-data-storage"

In [13]: client = bigquery.Client()
         print("Connected to BigQuery. Project ID:", client.project)

         Connected to BigQuery. Project ID: g23ai2059-big-data-management

In [14]: create_dataset(project_id, dataset_id)
         create_table(dataset_id, "Restaurants", restaurants_schema)
         create_table(dataset_id, "Customers", customers_schema)
         create_table(dataset_id, "Orders", orders_schema)

         Dataset food_delivery_dataset created successfully in project g23ai2059-big-data-management.
         Table Restaurants created successfully in dataset food_delivery_dataset.
         Table Customers created successfully in dataset food_delivery_dataset.
         Table Orders created successfully in dataset food_delivery_dataset.

In [15]: upload_to_bigquery(project_id,dataset_id, "Customers", "Datasets/Customers.csv",customers_schema)
         upload_to_bigquery(project_id,dataset_id, "Restaurants", "Datasets/Restaurants.csv",restaurants_schema)
         upload_to_bigquery(project_id,dataset_id, "Orders", "Datasets/Orders.csv",orders_schema)

         Data uploaded successfully to g23ai2059-big-data-management.food_delivery_dataset.Customers.
         Data uploaded successfully to g23ai2059-big-data-management.food_delivery_dataset.Restaurants.
         Data uploaded successfully to g23ai2059-big-data-management.food_delivery_dataset.Orders.

In [16]: #Sample Data
         query = f"SELECT * FROM `{project_id}.{dataset_id}.Orders` LIMIT 5"
         executeSQLSelect(query)

            OrderID CustomerID RestaurantID   OrderDate OrderStatus  OrderAmount
```

Step 5: Machine Learning

- Developed a Linear Regression model to predict order volume based on historical data.
- Evaluate the model using scikit-learn.



In [19]:
```
client = storage.Client()
print("Connected to Google Cloud Storage. Project ID:", client.project)
```

```
Connected to Google Cloud Storage. Project ID: g23ai2059-big-data-management
```

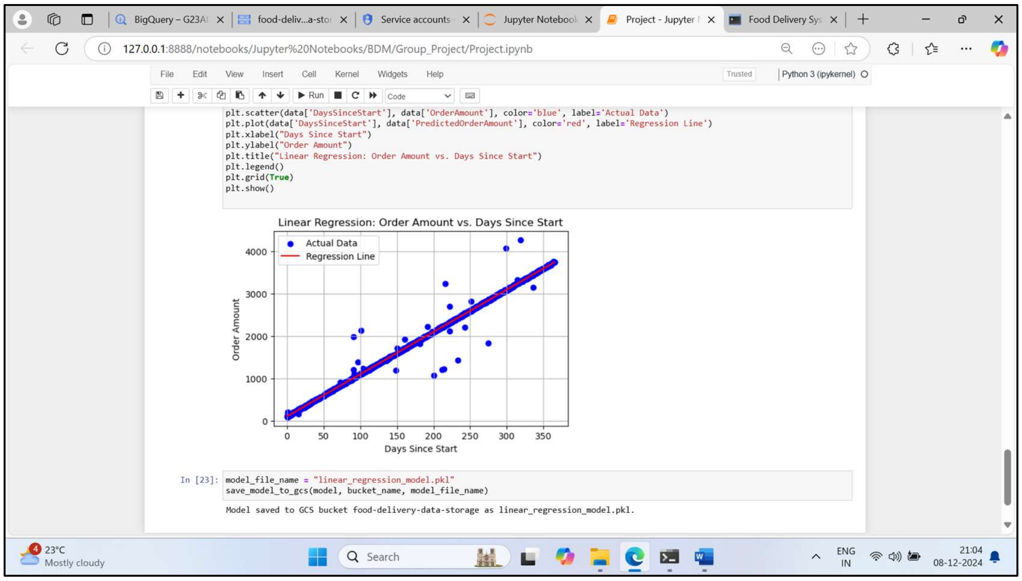
In [20]:
```
create_gcs_bucket(bucket_name)
upload_to_gcs(bucket_name, file_path= "Datasets/Restaurants.csv", destination_blob_name="Restaurants.csv" )
upload_to_gcs(bucket_name, file_path= "Datasets/Orders.csv", destination_blob_name="Orders.csv" )
upload_to_gcs(bucket_name, file_path= "Datasets/Customers.csv", destination_blob_name="Customers.csv" )
```

```
Bucket 'food-delivery-data-storage' already exists. Deleting it.
Bucket 'food-delivery-data-storage' deleted.
Bucket food-delivery-data-storage created successfully in location US.
File Datasets/Restaurants.csv uploaded to food-delivery-data-storage/Restaurants.csv.
File Datasets/Orders.csv uploaded to food-delivery-data-storage/Orders.csv.
File Datasets/Customers.csv uploaded to food-delivery-data-storage/Customers.csv.
```

In [21]:
```
data = read_data_from_gcs(bucket_name,"Orders.csv")
```

```
Data read successfully from food-delivery-data-storage/<Blob: food-delivery-data-storage, Orders.csv, 1733671540337003>.
```

In [22]:
```
data['OrderDate'] = pd.to_datetime(data['OrderDate'])
data['DaysSinceStart'] = (data['OrderDate'] - data['OrderDate'].min()).dt.days
X = data[['DaysSinceStart']]
y = data['OrderAmount']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
data['PredictedOrderAmount'] = model.predict(data[['DaysSinceStart']])

plt.figure(figsize=(6, 4))
plt.scatter(data['DaysSinceStart'], data['OrderAmount'], color='blue', label='Actual Data')
plt.plot(data['DaysSinceStart'], data['PredictedOrderAmount'], color='red', label='Regression Line')
```



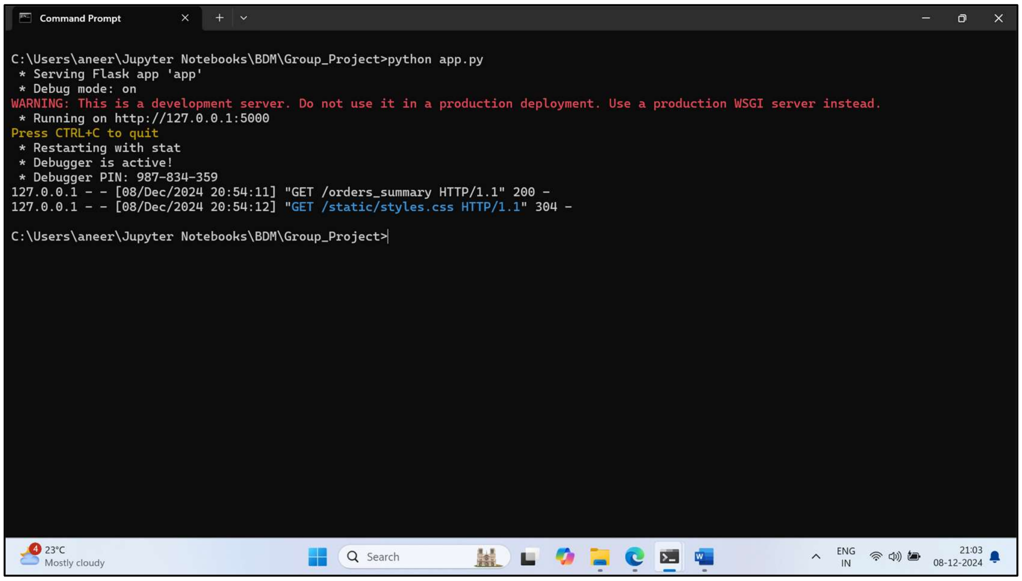
```
plt.scatter(data['DaysSinceStart'], data['OrderAmount'], color='blue', label='Actual Data')
plt.plot(data['DaysSinceStart'], data['PredictedOrderAmount'], color='red', label='Regression Line')
plt.xlabel("Days Since Start")
plt.ylabel("Order Amount")
plt.title("Linear Regression: Order Amount vs. Days Since Start")
plt.legend()
plt.grid(True)
plt.show()
```

In [23]:
```
model_file_name = "linear_regression_model.pkl"
save_model_to_gcs(model, bucket_name, model_file_name)
```

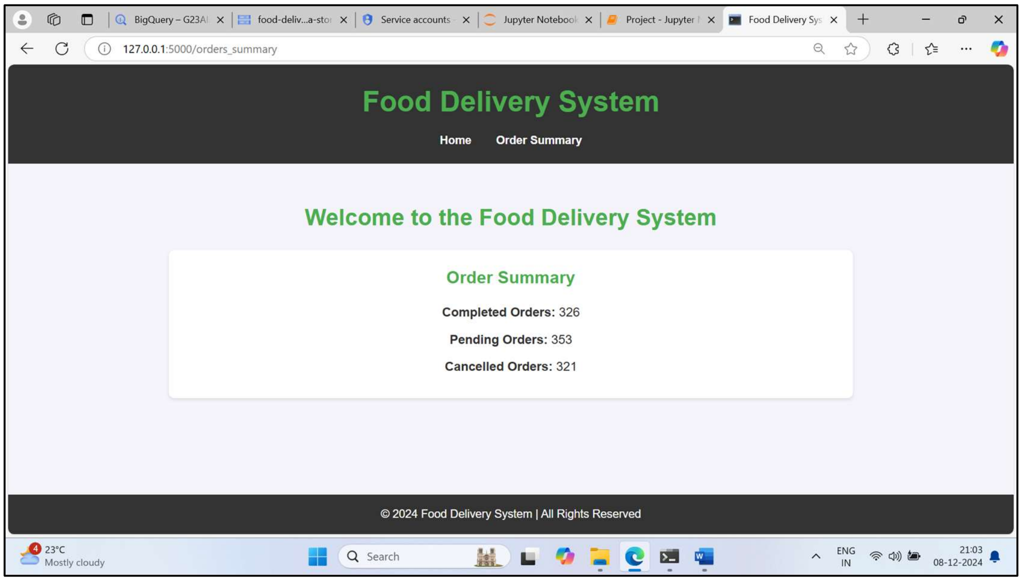
```
Model saved to GCS bucket food-delivery-data-storage as linear_regression_model.pkl.
```

Step 6: Frontend

- Designed a basic yet functional HTML interface integrated with Flask templates.
- Displayed order summaries and visualizations dynamically.



```
C:\Users\aneer\Jupyter Notebooks\BDM\Group_Project>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 987-834-359
127.0.0.1 - - [08/Dec/2024 20:54:11] "GET /orders_summary HTTP/1.1" 200 -
127.0.0.1 - - [08/Dec/2024 20:54:12] "GET /static/styles.css HTTP/1.1" 304 -

C:\Users\aneer\Jupyter Notebooks\BDM\Group_Project>
```



Food Delivery System

Home   Order Summary

Welcome to the Food Delivery System

Order Summary

Completed Orders: 326

Pending Orders: 353

Cancelled Orders: 321

© 2024 Food Delivery System | All Rights Reserved

Results

Achievements

1. **Scalability:** Successfully stored and queried large datasets in BigQuery.
2. **Efficiency:** Reduced data processing time using GCP tools.
3. **Predictive Insights:** Achieved acceptable accuracy in predicting order trends.
4. **Accessibility:** Enabled stakeholders to interact with the system through frontend.

Challenges

1. **Data Cleaning:** Ensuring data consistency during synthetic data generation.
2. **Visualization:** Balancing clarity and complexity in visual outputs.

Future Scope

1. **Advanced Analytics:** Incorporate customer segmentation and recommendation engines.
2. **Real-time Processing:** Use tools like Apache Kafka for real-time data ingestion.
3. **Enhanced Frontend:** Develop a fully responsive web application.
4. **Scalable ML Models:** Integrate more complex models for better predictions.
5. **Multi-cloud Support:** Extend functionality to other cloud platforms.

Conclusion

The Food Delivery Management System demonstrates the potential of cloud-based solutions for managing large-scale data. By integrating BigQuery, GCS, and Python, the project showcases a seamless workflow from data ingestion to visualization. This system not only meets the current requirements but also provides a robust foundation for future enhancements.

Project Link: **https://github.com/aneerban10/BDM-Group-Project-Aneerban-Shikha-Vanshika**

References

1. Google Cloud Platform Documentation
2. Scikit-learn Reference
3. Flask Documentation