Big Data Analytics Project

By:
Saad Ahmed Khan – 24440
Haider Saeed Khan – 24123
Talal Khan - 25253

Step By Step Procedure

Setting up Hadoop Cluster

Command used: docker-compose up -d

```
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\HBASE\docker-hbase> docker-compose -f "docker-compose-standalone.yml" up -d
time="2024-12-31T23:09:59+05:00" level=warning msg="C:\\Users\\Saad Ahmed Khan\\Projects\\BIG DATA\\BigDataProject\\BDA_try11\\HBASE\\docker-hbase\\docker-compose-standalone.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 11/11
 ✓ Network bda_network          Created                                                                    0.0s
 ✓ Container nodemanager        Started                                                                    2.5s
 ✓ Container namenode           Started                                                                    2.5s
 ✓ Container spark-master-bda   Started                                                                    2.6s
 ✓ Container hbase              Started                                                                    2.5s
 ✓ Container historyserver      Started                                                                    2.2s
 ✓ Container resourcemanager    Started                                                                    2.2s
 ✓ Container datanode           Started                                                                    2.6s
 ✓ Container zookeeperbda       Started                                                                    2.6s
 ✓ Container kafka              Started                                                                    2.9s
 ✓ Container spark-worker-bda1  Started                                                                    2.7s
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\HBASE\docker-hbase>
```

Ingesting Data into HDFS

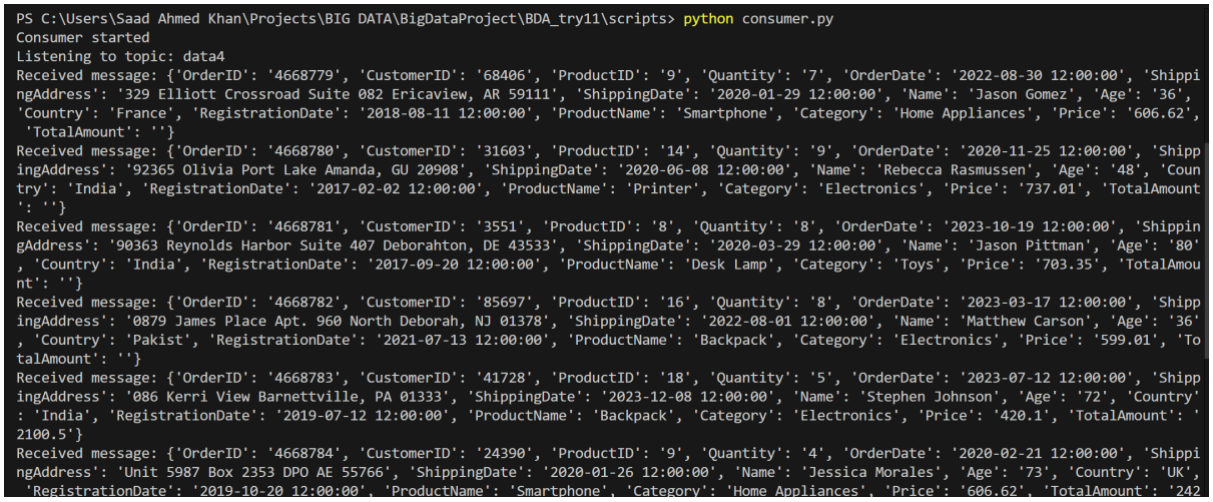
Running the upload_to_namenode.sh script (in the scripts folder)

```
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\scripts> bash upload_to_namenode.sh
Successfully copied 844MB to namenode:/data/temp_data1.csv
File successfully copied and renamed to data.csv in /data inside namenode container.
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\scripts>
```

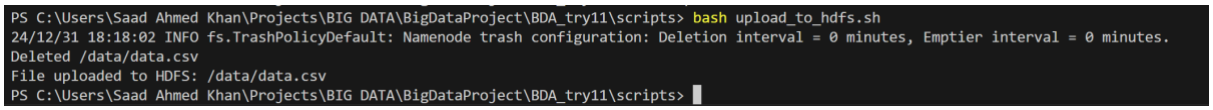
Setting up and running kafka producer.py file to generate data

```
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\scripts> python ./producer.py
start
Message delivered to data4 [0]
Message delivered to data4 [0]
Message delivered to data4 [0]
Message delivered to data4 [0]
Message delivered to data4 [0]
Message delivered to data4 [0]
```

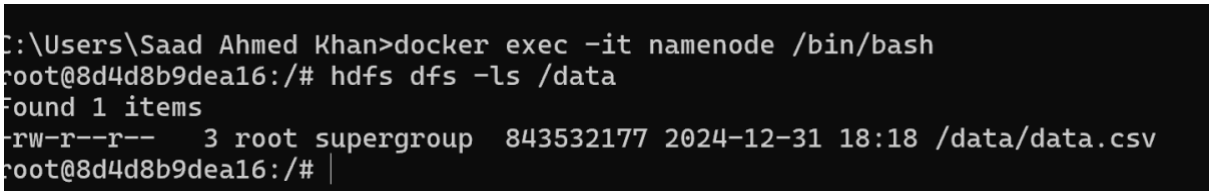
Setting up and running kafka consumer.py to ingest data inside namenode



Inserting Data into HDFS using the upload_to_hdfs.sh script



Finally verifying that the data has been moved to hdfs:



Printing data using cat:

EDA on ingested data using python:

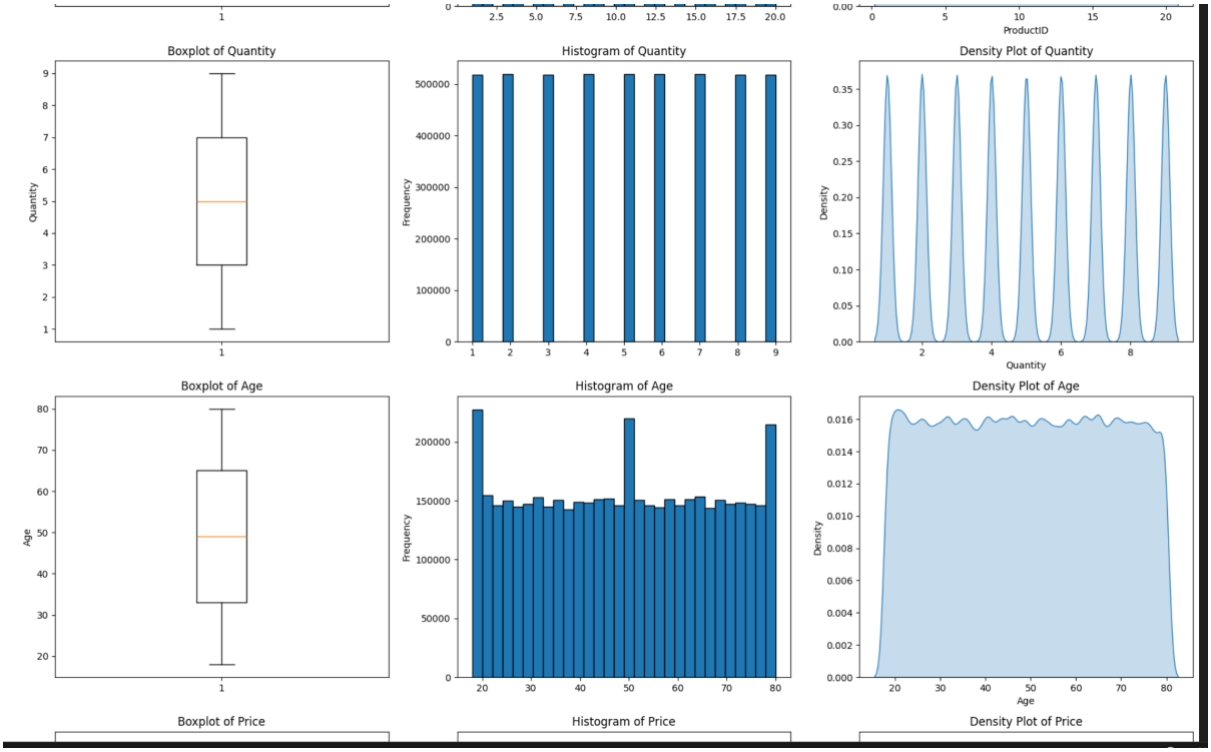
All the EDA steps can be found in the attached EDA notebook. A brief summary is provided below.

1) Displaying column info;

```
df = pd.read_csv(StringIO(file_content))

# Show the DataFrame
print(df.head())
```

```
start
   OrderID  CustomerID  ProductID  Quantity           OrderDate  \
0        1       92256         15         1 2023-06-16 12:00:00
1        2        7170          2         7 2020-11-09 12:00:00
2        3       57706         20         6 2021-11-14 12:00:00
3        4       78850         16         4 2020-08-13 12:00:00
4        5       37213          7         2 2021-11-28 12:00:00

                                ShippingAddress         ShippingDate  \
0  485 Jackson Falls Apt. 624 Lake Michelleland, ...  2021-05-23 12:00:00
1     074 Brown Estate Suite 175 Robinbury, MO 37341  2022-10-01 12:00:00
2          0362 Adam Vista Patriciaborough, UT 39097  2023-09-27 12:00:00
3             329 Isabel Ranch Grimesshire, HI 37219  2022-03-05 12:00:00
4  711 Skinner Street Suite 169 East Annette, SC ...  2020-06-23 12:00:00

               Name  Age  Country     RegistrationDate ProductName  \
0         Kyle Hunt   20   Canada  2022-07-01 12:00:00       Mouse
1      Taylor Black   22   Pakist  2021-12-21 12:00:00  Headphones
2    Mario Bartlett   73       UK  2014-01-19 12:00:00  Smartphone
3  Danielle Williams   57   Pakist  2018-01-27 12:00:00    Backpack
4        Jerry Hall   27  Germany  2014-10-23 12:00:00     Printer

      Category   Price  TotalAmount
0  Electronics  589.75       589.75
1  Electronics  803.02          NaN
2  Electronics  420.68      2524.08
3  Electronics  599.01          NaN
4        Books  251.30          NaN
```

2) Box plot, histograms and frequency density charts for all numerical columns:



3) Frequency tables for categorical columns

```
start
Frequency Table for OrderDate:
OrderDate
2020-12-28 12:00:00    3406
2022-10-29 12:00:00    3391
2021-03-18 12:00:00    3369
2022-08-01 12:00:00    3364
2023-05-30 12:00:00    3355
                       ...
2023-08-16 12:00:00    3036
2022-04-13 12:00:00    3030
2021-06-15 12:00:00    3024
2023-11-01 12:00:00    3012
2021-11-28 12:00:00    3010
Name: count, Length: 1461, dtype: int64

Frequency Table for ShippingAddress:
ShippingAddress
USNS King FPO AE 57227                                2
USNV Smith FPO AA 93867                               2
USCGC Wagner FPO AA 61054                             2
USNV Johnson FPO AP 44491                             2
USS Torres FPO AE 29227                               2
                                                     ..
PSC 4349, Box 2735 APO AE 79661                       1
...
08300 Benjamin Shoals East Annettefurt, HI 73623      1
864 Garcia Ramp Gonzalezborough, NH 74311             1
Name: count, Length: 4668759, dtype: int64
```

4) Finding numerical Column outliers

```
            outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)][column]

            if not outliers.empty:
                print(f"{column} Outliers:")
                for outlier in outliers:
                    print(f"- {outlier}")
                print()

    numerical_outliers(data)
```

[7]

```
start
TotalAmount Outliers:
- 7779.51
- 7227.18
- 7227.18
- 7227.18
- 7072.2
- 7072.2
- 7227.18
- 7072.2
- 7779.51
- 7227.18
- 7779.51
- 7779.51
- 7227.18
- 7779.51
- 7779.51
- 7779.51
- 7072.2
- 7072.2
- 7072.2
- 7072.2
- 7779.51
- 7779.51
- 7779.51
...
```

5) Missing Value analysis

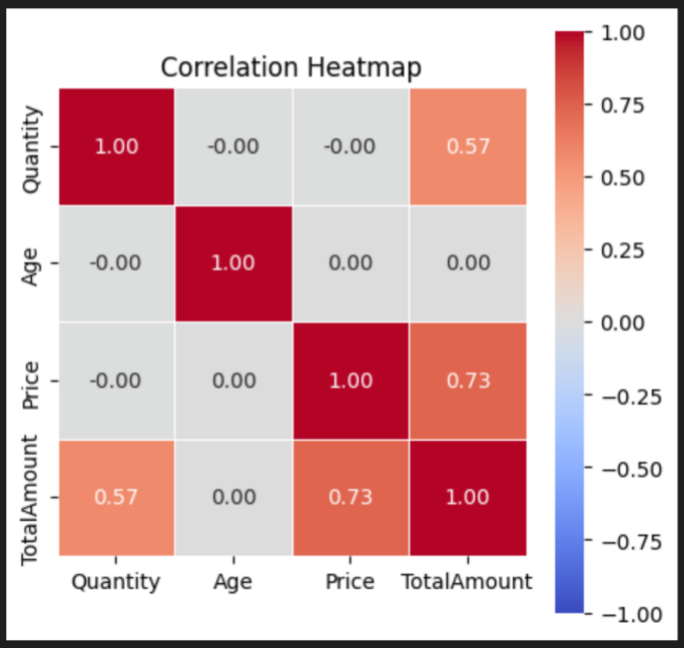
```python
print('start')
def check_missing_data(df):
    missing_summary = df.isnull().sum()
    missing_percentage = (missing_summary / len(df)) * 100

    print("Missing Values Summary:")
    print(pd.DataFrame({
        'Missing Count': missing_summary,
        'Percentage': missing_percentage
    }))

    return missing_summary, missing_percentage

missing_summary, missing_percentage = check_missing_data(data)
print(missing_summary)
```

```
start
Missing Values Summary:
                 Missing Count  Percentage
OrderID                      0    0.000000
CustomerID                   0    0.000000
ProductID                    0    0.000000
Quantity                     0    0.000000
OrderDate                    0    0.000000
ShippingAddress              0    0.000000
ShippingDate                 0    0.000000
Name                         0    0.000000
Age                          0    0.000000
Country                      0    0.000000
RegistrationDate             0    0.000000
ProductName                  0    0.000000
Category                     0    0.000000
Price                        0    0.000000
TotalAmount            3268360   70.004269
OrderID                      0
```

6) Correlation Analysis:

```
    print("Correlation Table:")
    print(correlation_matrix)
```

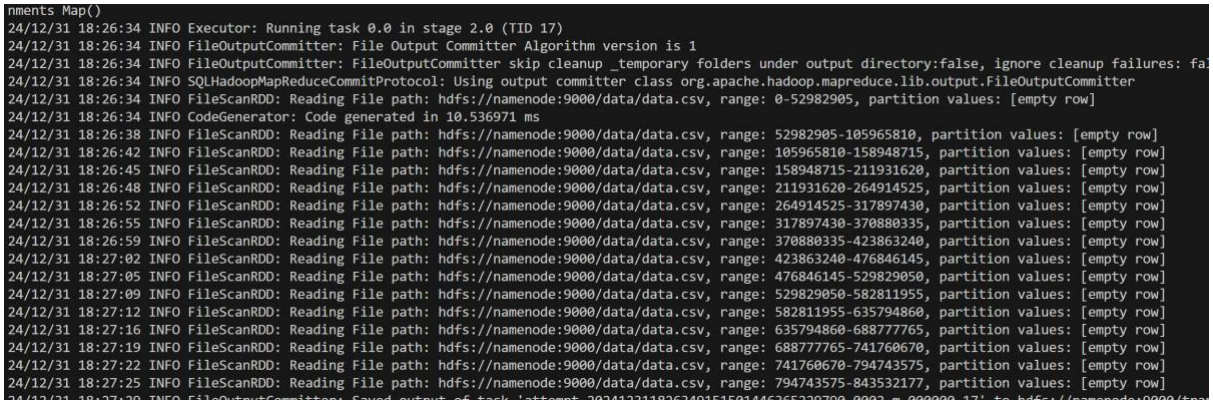
start



Applying Transformation on ingested data

Using the run_spark_job.sh file

```
PS C:\Users\Saad Ahmed Khan\Projects\BIG DATA\BigDataProject\BDA_try11\scripts> bash run_spark_job.sh
24/12/31 18:26:20 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /transformation
Successfully copied 4.1kB to spark-master-bda:/spark/spark_analysis.py
24/12/31 18:26:23 INFO SparkContext: Running Spark version 3.3.0
24/12/31 18:26:23 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/12/31 18:26:23 INFO ResourceUtils: ==============================================================
24/12/31 18:26:23 INFO ResourceUtils: No custom resources configured for spark.driver.
24/12/31 18:26:23 INFO ResourceUtils: ==============================================================
24/12/31 18:26:23 INFO SparkContext: Submitted application: Transform and Combine Partitioned Files
24/12/31 18:26:23 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -
> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.
0)
24/12/31 18:26:23 INFO ResourceProfile: Limiting resource is cpu
24/12/31 18:26:23 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/12/31 18:26:23 INFO SecurityManager: Changing view acls to: root
24/12/31 18:26:23 INFO SecurityManager: Changing modify acls to: root
24/12/31 18:26:23 INFO SecurityManager: Changing view acls groups to:
24/12/31 18:26:23 INFO SecurityManager: Changing modify acls groups to:
24/12/31 18:26:23 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(root); groups with view
permissions: Set(); users  with modify permissions: Set(root); groups with modify permissions: Set()
24/12/31 18:26:23 INFO Utils: Successfully started service 'sparkDriver' on port 45249.
24/12/31 18:26:23 INFO SparkEnv: Registering MapOutputTracker
24/12/31 18:26:23 INFO SparkEnv: Registering BlockManagerMaster
24/12/31 18:26:24 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
24/12/31 18:26:24 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/12/31 18:26:24 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/12/31 18:26:24 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-c68cc44e-c480-4e0d-b3d2-0f7ceed61ac4
24/12/31 18:26:24 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
```

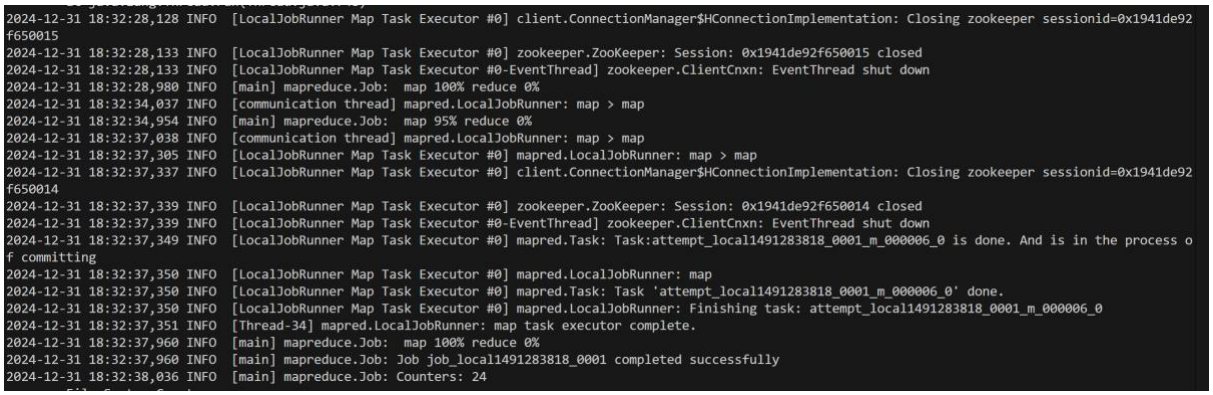


Verifying that spark job has done its job (saved the transomed data into transformation directory in hdfs)

```
C:\Users\Saad Ahmed Khan>docker exec -it namenode /bin/bash
root@8d4d8b9dea16:/# hdfs dfs -ls /
Found 8 items
drwxrwxrwt   - root root                0 2024-12-30 17:37 /app-logs
drwxr-xr-x   - root supergroup          0 2024-12-31 18:18 /data
-rw-r--r--   3 root supergroup  843411626 2024-12-30 16:08 /data.csv
drwxr-xr-x   - root supergroup          0 2024-12-31 18:10 /hbase
drwxr-xr-x   - root supergroup          0 2024-12-30 17:52 /output
drwxr-xr-x   - root supergroup          0 2024-12-30 14:19 /rmstate
drwx------   - root supergroup          0 2024-12-30 17:36 /tmp
drwxr-xr-x   - root supergroup          0 2024-12-31 18:27 /transformation
root@8d4d8b9dea16:/# hdfs dfs -ls /transformation
Found 2 items
-rw-r--r--   3 root supergroup          0 2024-12-31 18:27 /transformation/_SUCCESS
-rw-r--r--   3 root supergroup  860892470 2024-12-31 18:27 /transformation/part-00000-95176dd5-8e08-4a55-af2a-eda6706a0d
b0-c000.csv
root@8d4d8b9dea16:/#
```

Ingesting data from hdfs to hbase

Running hdfs_to_hbase script



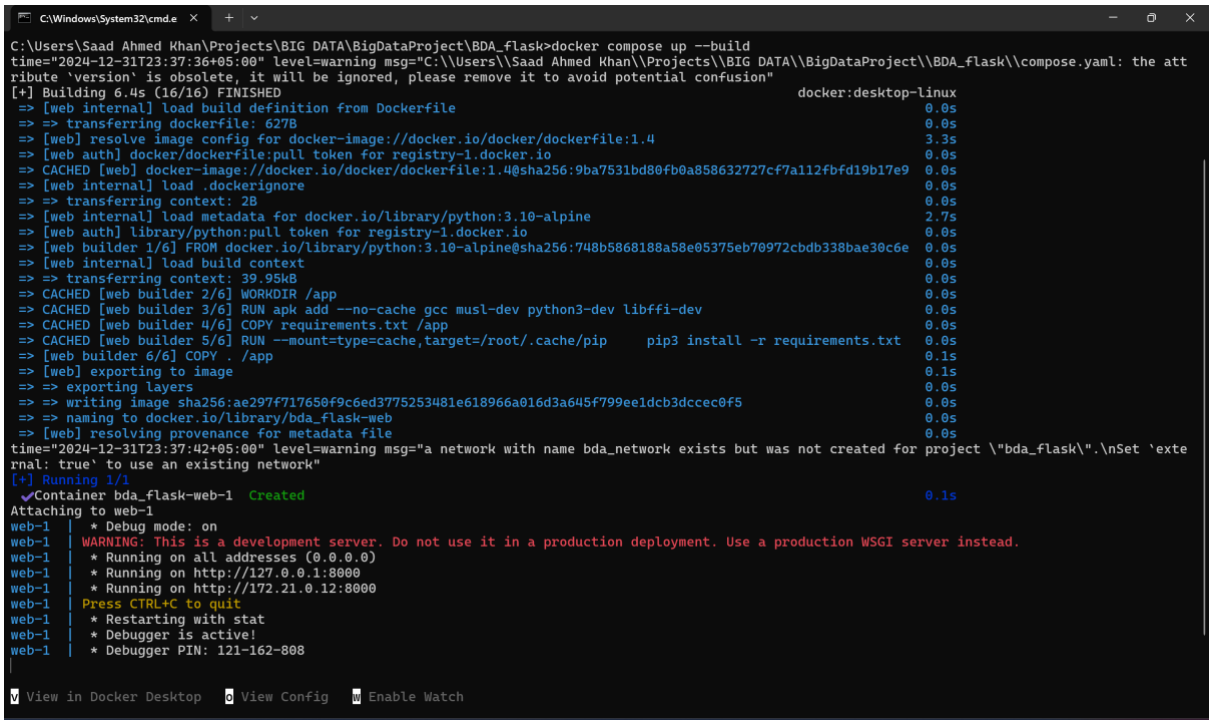
```
    File System Counters
            FILE: Number of bytes read=179718186
            FILE: Number of bytes written=183172633
            FILE: Number of read operations=0
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=3679575350
            HDFS: Number of bytes written=0
            HDFS: Number of read operations=42
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=0
    Map-Reduce Framework
            Map input records=4668792
            Map output records=4668792
            Input split bytes=777
            Spilled Records=0
            Failed Shuffles=0
            Merged Map outputs=0
            GC time elapsed (ms)=8096
            CPU time spent (ms)=0
            Physical memory (bytes) snapshot=0
            Virtual memory (bytes) snapshot=0
            Total committed heap usage (bytes)=879886336
    ImportTsv
            Bad Lines=0
    File Input Format Counters
            Bytes Read=860917046
    File Output Format Counters
```

Verifying that data has been inserted inside hbase table

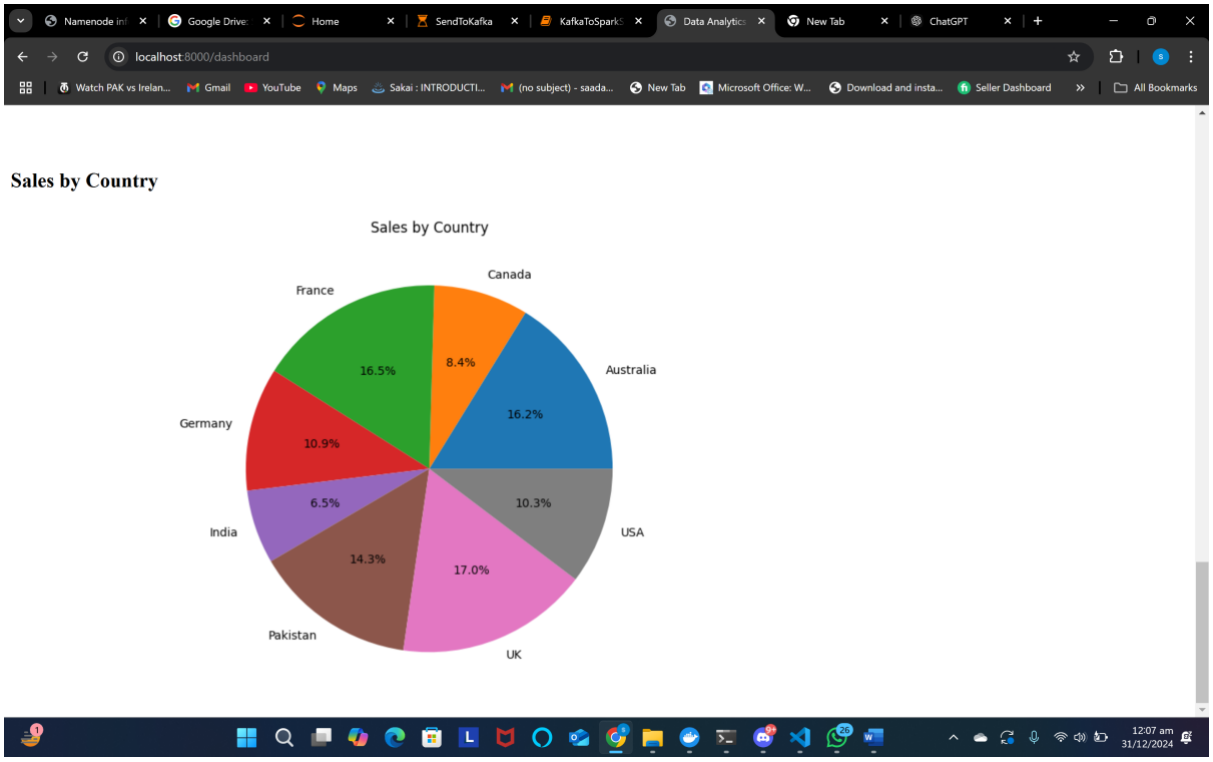
```
hbase(main):004:0> scan 'ec'
ROW                             COLUMN+CELL
 1                              column=customer:Age, timestamp=1735669811231, value=20
^C 1                              column=customer:Country, timestamp=1735669811231, value=Canada

 1                              column=customer:CustomerID, timestamp=1735669811231, value=92256
 1                              column=customer:Name, timestamp=1735669811231, value=Kyle Hunt
 1                              column=customer:RegistrationDate, timestamp=1735669811231, value=2022-07-01 12:00:00
 1                              column=order:OrderDate, timestamp=1735669811231, value=2023-06-16 12:00:00
 1                              column=order:Quantity, timestamp=1735669811231, value=1
 1                              column=order:TotalAmount, timestamp=1735669811231, value=589.75
 1                              column=product:Category, timestamp=1735669811231, value=Electronics
 1                              column=product:Price, timestamp=1735669811231, value=589.75
 1                              column=product:ProductID, timestamp=1735669811231, value=15
 1                              column=product:ProductName, timestamp=1735669811231, value=Mouse
 1                              column=shipment:ShippingAddress, timestamp=1735669811231, value=485 Jackson Falls Apt. 6
                                24 Lake Michelleland LA 31239
 1                              column=shipment:ShippingDate, timestamp=1735669811231, value=2021-05-23 12:00:00
```

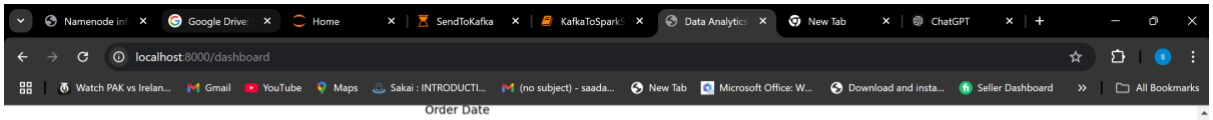
```
1000742                         column=product:Price, timestamp=1735669811231, value=38.31
1000742                         column=product:ProductID, timestamp=1735669811231, value=12
1000742                         column=product:ProductName, timestamp=1735669811231, value=Gaming Console
1000742                         column=shipment:ShippingAddress, timestamp=1735669811231, value=54515 Rodriguez Glens Su
                                ite 695 Denisebury NE 27642
1000742                         column=shipment:ShippingDate, timestamp=1735669811231, value=2021-08-18 12:00:00
1000743                         column=customer:Age, timestamp=1735669811231, value=22
1000743                         column=customer:Country, timestamp=1735669811231, value=Australia
1000743                         column=customer:CustomerID, timestamp=1735669811231, value=42919
1000743                         column=customer:Name, timestamp=1735669811231, value=Lawrence Coleman
1000743                         column=customer:RegistrationDate, timestamp=1735669811231, value=2015-07-03 12:00:00
1000743                         column=order:OrderDate, timestamp=1735669811231, value=2021-08-29 12:00:00
1000743                         column=order:Quantity, timestamp=1735669811231, value=4
1000743                         column=order:TotalAmount, timestamp=1735669811231, value=1456.52
1000743                         column=product:Category, timestamp=1735669811231, value=Electronics
1000743                         column=product:Price, timestamp=1735669811231, value=364.13
1000743                         column=product:ProductID, timestamp=1735669811231, value=17
1000743                         column=product:ProductName, timestamp=1735669811231, value=Desk Lamp
1000743                         column=shipment:ShippingAddress, timestamp=1735669811231, value=9753 Roberts Hills Bryan
                                burgh OR 70954
1000743                         column=shipment:ShippingDate, timestamp=1735669811231, value=2021-07-13 12:00:00
1000744                         column=customer:Age, timestamp=1735669811231, value=79
1000744                         column=customer:Country, timestamp=1735669811231, value=Australia
1000744                         column=customer:CustomerID, timestamp=1735669811231, value=62641
1000744                         column=customer:Name, timestamp=1735669811231, value=Ronald Greene
1000744                         column=customer:RegistrationDate, timestamp=1735669811231, value=2023-12-07 12:00:00
1000744                         column=order:OrderDate, timestamp=1735669811231, value=2022-06-22 12:00:00
1000744                         column=order:Quantity, timestamp=1735669811231, value=2
```

Setting up Flask Dashboard



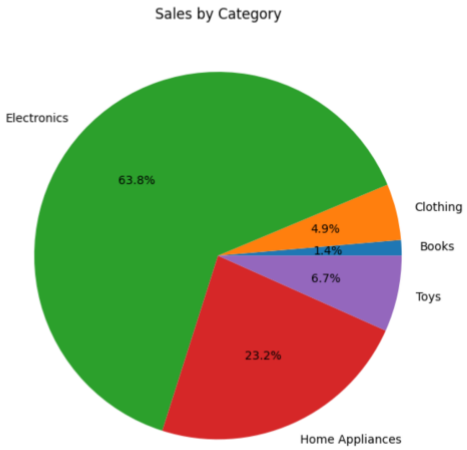
Dashboard:



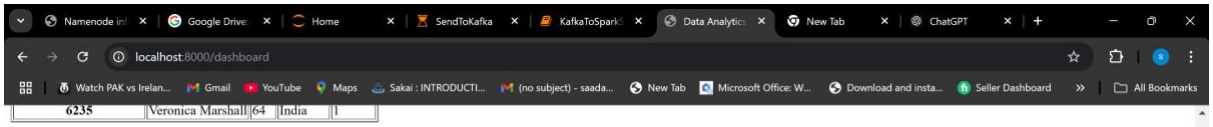


Order Date

Sales by Category

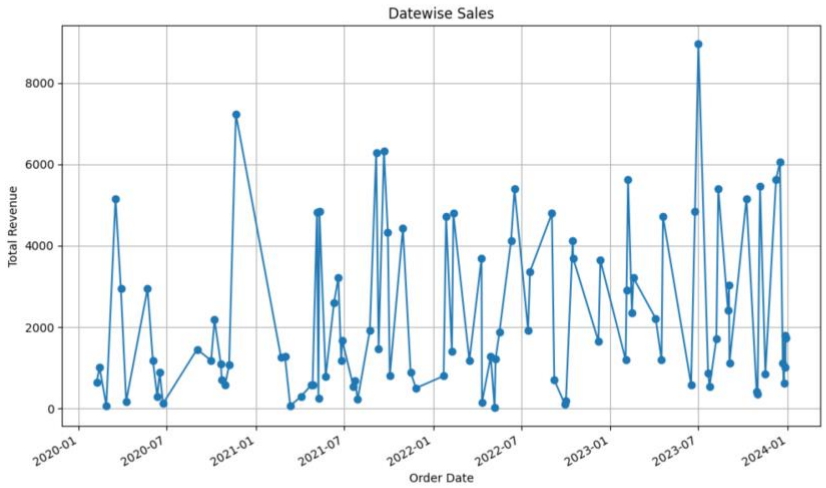


Sales by Country

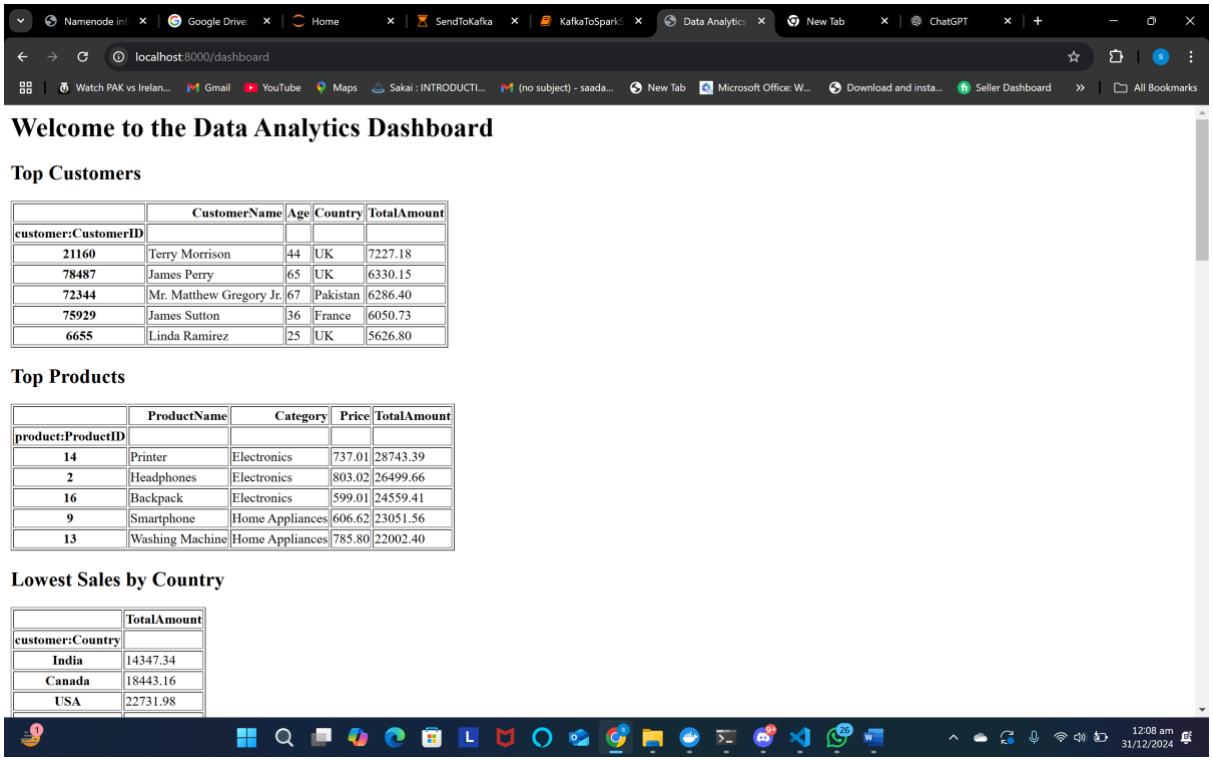


6235	Veronica Marshall	64	India	1

Datewise Sales



Sales by Category



Welcome to the Data Analytics Dashboard

Top Customers

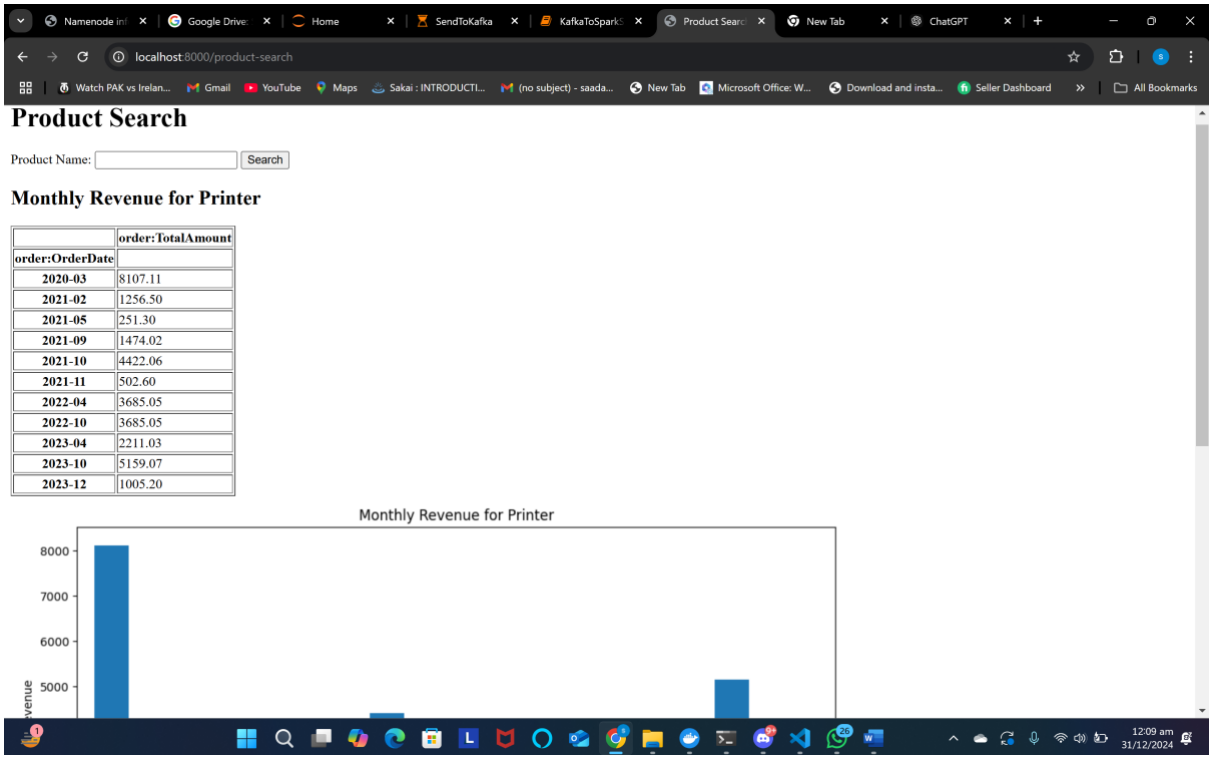
	CustomerName	Age	Country	TotalAmount
customer:CustomerID				
21160	Terry Morrison	44	UK	7227.18
78487	James Perry	65	UK	6330.15
72344	Mr. Matthew Gregory Jr.	67	Pakistan	6286.40
75929	James Sutton	36	France	6050.73
6655	Linda Ramirez	25	UK	5626.80

Top Products

	ProductName	Category	Price	TotalAmount
product:ProductID				
14	Printer	Electronics	737.01	28743.39
2	Headphones	Electronics	803.02	26499.66
16	Backpack	Electronics	599.01	24559.41
9	Smartphone	Home Appliances	606.62	23051.56
13	Washing Machine	Home Appliances	785.80	22002.40

Lowest Sales by Country

	TotalAmount
customer:Country	
India	14347.34
Canada	18443.16
USA	22731.98



Product Search

Product Name: Search

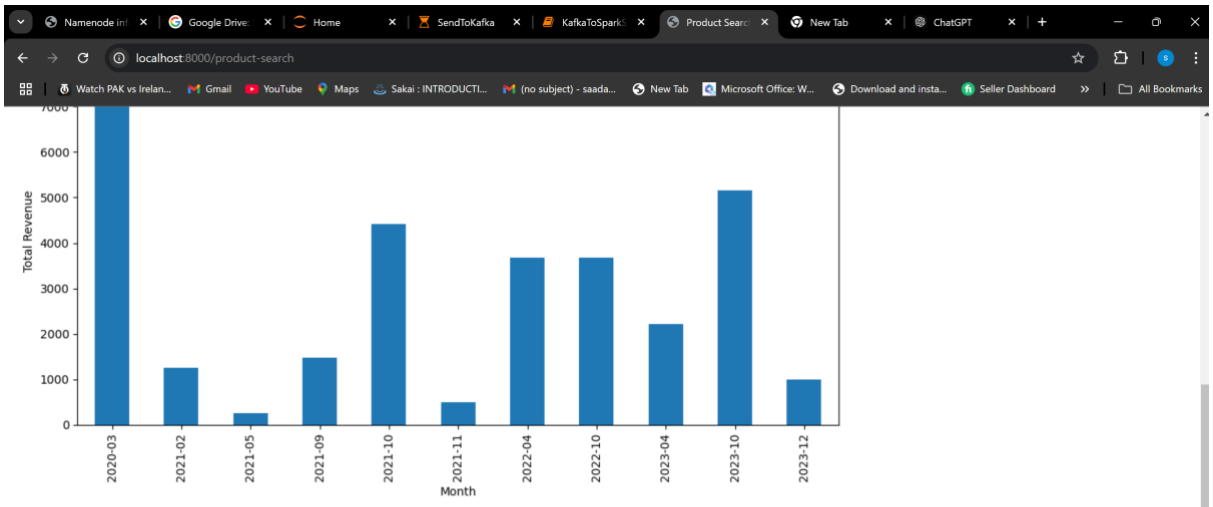
Monthly Revenue for Printer

	order:TotalAmount
order:OrderDate	
2020-03	8107.11
2021-02	1256.50
2021-05	251.30
2021-09	1474.02
2021-10	4422.06
2021-11	502.60
2022-04	3685.05
2022-10	3685.05
2023-04	2211.03
2023-10	5159.07
2023-12	1005.20

2023-10	5159.07
2023-12	1005.20

Top 5 Countries that Order Printer

	order:TotalAmount



Top 5 Countries that Order Printer

	order:TotalAmount
customer:Country	
France	9581.13
Germany	5896.08
Australia	5159.07
Canada	4941.55
UK	2948.04