

# Data Science CA2

Alireza Karimi 810101492

Mohammad Taha Majlesi 810101504

Mohammad Hossein Mazhari 810101520

## Guides for working with terminal and kafka.

```
### link for this commands:
```

```
https://kafka.apache.org/quickstart
```

```
# This initializes the log metadata directory for KRaft.
```

```
$ cd ~/Documents/install_file/kafka/kafka_2.13-4.0.0
```

```
# The address to my CA files:
```

```
# cd Documents/uni/term6/Data-Science/CA/2/DS-CA2/
```

```
# Generate a Cluster UUID
```

```
$ KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"
```

```
# Format Log Directories
```

```
$ bin/kafka-storage.sh format --standalone -t $KAFKA_CLUSTER_ID -c config/server.properties
```

```
# Start the Kafka Server
```

```
$ bin/kafka-server-start.sh config/server.properties
```

```
# Create a topic to store your events (In a new terminal)
```

```
$ bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server local
```

```
# Write some events into the topic  
$ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server lo
```

```
# Read the events(In a new terminal)  
$ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --l
```

```
# Terminate the Kafka environment(Stop the producer and consumer clients and  
$ rm -rf /tmp/kafka-logs /tmp/kraft-combined-logs
```

```
##### check that kafka is running  
$ netstat -tuln | grep 9092  
or this one:  
$ nc -zv localhost 9092
```

```
#To run the python code "darooghe_pulse":  
python darooghe_pulse.py
```

```
lsof -i :9092
```

```
#Stop the existing container using port 9000  
###Run this to find the container using port 9000:  
$ docker ps
```

```
###Look for something exposing 0.0.0.0:9000 and note the CONTAINER ID or NAME  
$ docker stop <container_id_or_name>
```

1. List all topics

```
bin/kafka-topics.sh \
--bootstrap-server localhost:9092 \
--list
```

This will print every topic name on your broker.

## 2. Describe a topic (partitions, replicas, offsets)

```
bin/kafka-topics.sh \
--bootstrap-server localhost:9092 \
--describe \
--topic <your_topic_name>
```

That shows you partition counts, leader/ISR, and log size (i.e. high-watermark offset).

## 3. Peek at the messages in a topic

```
bin/kafka-console-consumer.sh \
--bootstrap-server localhost:9092 \
--topic <your_topic_name> \
--from-beginning \
--max-messages 10
```

# Part1: Environment Setup

Input

```
(base) alireza@alirezas-MacBook-Air DS-CA2 % python darooghe_pulse.py
```

Output

```
2025-04-19 20:08:44,566 INFO Producing 20000 historical events...
2025-04-19 20:08:45,658 INFO Historical events production completed.
2025-04-19 20:08:45,658 INFO Starting continuous event production...
```

## input

```
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % bin/kafka-topics.sh \
--bootstrap-server localhost:9092 \
--list
```

## output

```
...a.Kafka config/server.properties ... ~ -- -zsh /tmp -- -zsh ...afka/kafka_2.13-4.0.0 -- -zsh ...afka/kafka_2.13-4.0.0 -- -zsh +
Last login: Mon Apr 28 09:16:17 on ttys022
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % bin/kafka-topics.sh --create --topic commission_by_type --bootstrap-server localhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore('_') could collide. To avoid issues it is best to use either, but not both.
Created topic commission_by_type.
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % bin/kafka-topics.sh --create --topic commission_ratio --bootstrap-server localhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore('_') could collide. To avoid issues it is best to use either, but not both.
Created topic commission_ratio.
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % bin/kafka-topics.sh --create --topic darooghe.top_merchants --bootstrap-server localhost:9092
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore('_') could collide. To avoid issues it is best to use either, but not both.
Created topic darooghe.top_merchants.
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % bin/kafka-topics.sh \
--bootstrap-server localhost:9092 \
--list
__consumer_offsets
commission_by_type
commission_ratio
darooghe.error_logs
darooghe.fraud_alerts
darooghe.insights
darooghe.top_merchants
darooghe.transactions
(base) alireza@alirezas-MacBook-Air kafka_2.13-4.0.0 % java -version
```

# Another part for creating kafdrop but unsuccessful

Ahh yep — that error tells us exactly what's up 🤔

🚫 Port 9000 is already being used on your Mac, probably by a previously launched Kafdrop container that didn't stop cleanly.

## Quick Fix Options

### 🔧 Option 1: Stop the running container using port 9000

Run this to find it:

```
bash  
CopyEdit  
docker ps
```

Look for the container using **port 9000** (check the "PORTS" column).

Then stop it:

```
bash  
CopyEdit  
docker stop <container_id>
```

Example:

```
bash  
CopyEdit  
docker stop aea9b8844959
```

Now try launching Kafdrop again.

## 🔧 Option 2: Run Kafdrop on a different port (e.g., 9001)

If you want to keep the other container running, just change the port:

```
bash
CopyEdit
docker run --platform linux/amd64 -d --rm -p 9001:9000 \
-e KAFKA_BROKERCONNECT=host.docker.internal:9092 \
obsidiandynamics/kafdrop
```

Now you can visit:

Kafka Cluster Overview

Bootstrap servers	host.docker.internal:9092
Total topics	0
Total partitions	0
Total preferred partition leader	0%
Total under-replicated partitions	0

Brokers

ID	Host	Port	Rack	Controller	Number of partitions (% of total)
1	host.docker.internal	9092	-	Yes	0 (0%)

Topics    ACLs

Name	Partitions	% Preferred	# Under-replicated	Custom Config
No topics available	(0)			

Absolutely — you can totally monitor the `darooghe.error_logs` topic straight from the command line! 🔐 Here's how to do it using **Kafka's CLI tools** that come with your Kafka installatio

## 💻 Option 2: On Your Mac Terminal (If Kafka CLI installed)

If you installed Kafka locally (e.g., via `brew`), just run this on your Mac terminal:

```
kafka-console-consumer \
--bootstrap-server localhost:9092 \
--topic darooghe.error_logs \
--from-beginning \
--property print.key=true \
--property print.value=true
```

This is the output, when I run `load_data.py`:

```
root
|-- amount: long (nullable = true)
|-- commission_amount: long (nullable = true)
|-- commission_type: string (nullable = true)
|-- customer_id: string (nullable = true)
|-- customer_type: string (nullable = true)
|-- device_info: struct (nullable = true)
|   |-- app_version: string (nullable = true)
|   |-- device_model: string (nullable = true)
|   |-- os: string (nullable = true)
|-- failure_reason: string (nullable = true)
|-- location: struct (nullable = true)
|   |-- lat: double (nullable = true)
|   |-- lng: double (nullable = true)
|-- merchant_category: string (nullable = true)
```

```

|-- merchant_id: string (nullable = true)
|-- payment_method: string (nullable = true)
|-- risk_level: long (nullable = true)
|-- status: string (nullable = true)
|-- timestamp: string (nullable = true)
|-- total_amount: long (nullable = true)
|-- transaction_id: string (nullable = true)
|-- vat_amount: long (nullable = true)

```

amount	commission_amount	commission_type	customer_id	customer_type
681670	136331	tiered	cust_498	CIP {2.4.1, Samsung G...}
1514059	30281	flat	cust_169	CIP  {NULL, NULL, NULL}
499205	9984	progressive	cust_879	business {1.9.5, Google Pi...
255756	5115	progressive	cust_28	individual {3.1.0, iPhone 15...}
1327604	26552	flat	cust_321	CIP {3.1.0, iPhone 15...}

only showing top 5 rows

merchant_category	total_commission	avg_commission	commission_to_amour
retail	33664610	20255.481347773766	0.01999950804145335
entertainment	33235002	20389.571779141104	0.01999952116690160
food_service	33755412	21031.409345794393	0.01999954244828
government	33785856	20702.117647058825	0.0199995276951645
transportation	34291043	20423.4919594997	0.0199995088400672

```

(spark-env) (base) alireza@alirezas-MacBook-Air DS-CA2 % python3 load_data.
25/04/21 18:29:45 WARN Utils: Your hostname, alirezas-MacBook-Air.local resolv
25/04/21 18:29:45 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
Setting default log level to "WARN".

```

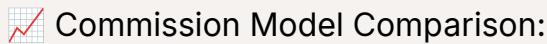
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel

```
25/04/21 18:29:46 WARN NativeCodeLoader: Unable to load native-hadoop libra
```



### Commission Efficiency by Merchant Category:

merchant_category	total_commission	avg_commission	commission_to_amour
retail	33664610	20255.481347773766	0.01999950804145335
entertainment	33235002	20389.571779141104	0.019999521166901603
food_service	33755412	21031.409345794393	0.019999542448285
government	33785856	20702.117647058825	0.019999527695164585
transportation	34291043	20423.4919594997	0.019999508840067227



### Commission Model Comparison:

merchant_category	avg_real	avg_flat	avg_progressive
retail	20255.481347773766	15000.0	20255.979602888085
entertainment	20389.571779141104	15000.0	20390.059950920244
food_service	21031.409345794393	15000.0	21031.890504672887
government	20702.117647058825	15000.0	20702.606544117636
transportation	20423.4919594997	15000.0	20423.993531864206



### Transactions per Hour:

hour	count
0	110
1	112
2	111
3	122
4	100
5	115
6	120
7	91
8	102
9	110

10  101
11  109
12  119
13  91
14  114
15  118
16  2325
17  3488
18  110
19  111
20  113
21  95
22  103
23  118

+-----+

#### 📅 Transactions per Day of Week:

day_of_week	count
1	779
2	7429

#### 👤 Customer Segmentation (Low/Medium/High frequency):

segment	count
Low	672
Medium	328

#### 🛒 Avg Amount and Volume by Merchant Category:

merchant_category	txn_count	avg_amount

```

|transportation|1679|1021199.6765932102|
|retail|1662|1012798.9801444043|
|government|1632|1035130.3272058824|
|entertainment|1630|1019502.9975460123|
|food_service|1605|1051594.5252336448|
+-----+-----+

```

 Transactions by Part of Day:

```

+-----+-----+
|part_of_day|count|
+-----+-----+
|Afternoon|6255|
|Evening|650|
|Morning|633|
|Night|670|
+-----+-----+

```

 Daily Spend Trends:

```

+-----+-----+-----+
|date|total_amount|txns_count|
+-----+-----+-----+
|2025-04-20|794293966|779|
|2025-04-21|7642503989|7429|
+-----+-----+-----+

```

After loading data to MongoDB using this simple load\_to\_mongo.py:

```

import json
from pymongo import MongoClient

# 1. Connect to MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["darooghe"] # Your database name
collection = db["transactions"] # Collection/table name

# 2. Open your JSONL file (line by line)

```

```

with open("transactions.jsonl", "r") as f:
    data = [json.loads(line) for line in f]

# 3. Insert into MongoDB
collection.insert_many(data)

print(f"✅ Inserted {len(data)} documents into MongoDB!")

```

more complex one:

```

import json
from pymongo import MongoClient
from datetime import datetime

# Connect to MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["darooghe"]
collection = db["transactions"]

# Optional: Clean old data (during development)
collection.delete_many({})
print("⚠️ Old data cleared.")

# Load and enrich JSONL file
with open("transactions.jsonl", "r") as f:
    data = [json.loads(line) for line in f]

# Add 'date' field from 'timestamp'
for tx in data:
    try:
        iso_ts = tx.get("timestamp")
        if iso_ts:
            dt_obj = datetime.fromisoformat(iso_ts.replace("Z", "+00:00"))
            tx["date"] = dt_obj.strftime("%Y-%m-%d") # Add date as YYYY-MM-DD
    except Exception as e:
        print(f"⚠️ Skipped bad timestamp in tx {tx.get('transaction_id')}: {e}")

```

```

# Insert into MongoDB
if data:
    collection.insert_many(data)
    print(f"✓ Inserted {len(data)} documents into MongoDB with 'date' field.")

# Add indexes for better performance
collection.create_index("date")
collection.create_index("merchant_id")
collection.create_index([("date", 1), ("merchant_id", 1)])
print("/Indexes on 'date', 'merchant_id', and ('date', 'merchant_id') created."
else:
    print("✗ No data found to insert.")

```

What happened in mongoDB:

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases: 'daroghe\_transactions' (selected), 'admin', 'config', 'daroghe', and 'local'. The main area displays the 'transactions' collection under the 'daroghe\_transactions' database. The 'Documents' tab shows 8.2K documents. A query builder at the top right allows for filtering, explaining queries, and performing find operations. Two documents are expanded to show their contents:

```

_id: ObjectId('68066142073b98e8a49d3245')
transaction_id: "87edfe0c-63ff-46e7-9fd9-ad75d13529f4"
timestamp: "2025-04-20T21:49:49.165116Z"
customer_id: "cust_498"
merchant_id: "merch_4"
merchant_category: "entertainment"
payment_method: "mobile"
amount: 681670
location: Object
device_info: Object
status: "approved"
commission_type: "tiered"
commission_amount: 13633
vat_amount: 61350
total_amount: 756653
customer_type: "CIP"
risk_level: 2
failure_reason: null

_id: ObjectId('68066142073b98e8a49d3246')
transaction_id: "dcb129ca-f8f9-4138-9fff-288e3c328288"
timestamp: "2025-04-21T09:28:03.397829Z"
customer_id: "cust_169"

```

**✓ 1. Check that documents have the `date` field**

## In Compass:

- Go to your `darooghe` database → `transactions` collection
- In the **filter bar**, paste this:

```
json
CopyEdit
{ "date": { "$exists": true } }
```

- Press **Enter** or click “**Find**”

This will show only the documents that **do** have a `date` field.

## 2. Query by a specific `date` and/or `merchant_id`

**Example:** Find all transactions from April 20th, 2025

```
json
CopyEdit
{ "date": "2025-04-20" }
```

**Example:** Find all transactions for a specific merchant

```
json
CopyEdit
{ "merchant_id": "merch_1578" }
```

**Example:** Combine both

```
json
CopyEdit
{
```

```
"date": "2025-04-20",
"merchant_id": "merch_1578"
}
```

👉 Tip: Use the **filter bar** in Compass for all of these — just paste and click **Find**.

### ✓ 3. Run a Summary Aggregation (e.g., total amount per merchant)

In MongoDB Compass:

Go to the **Aggregation tab** → (add a new stage and) paste this pipeline:

```
json
CopyEdit
[
  {
    "$group": {
      "_id": "$merchant_id",
      "total_amount": { "$sum": "$amount" },
      "transaction_count": { "$sum": 1 }
    },
    { "$sort": { "total_amount": -1 } }
  ]
]
```

This gives you:

- Total transaction amount per merchant
- Number of transactions
- Sorted by top spending

### ✓ 4. Optional: Show top merchants on a specific date

```
json
CopyEdit
[
  { "$match": { "date": "2025-04-20" } },
  {
    "$group": {
      "_id": "$merchant_id",
      "total_amount": { "$sum": "$amount" }
    }
  },
  { "$sort": { "total_amount": -1 } }
]
```



## Extra Query: Find All Transactions With High Risk

```
json
CopyEdit
{ "risk_level": { "$gte": 4 } }
```

## generate pipeline

The screenshot shows the Compass MongoDB interface. On the left, the connection tree displays 'Darooghe\_transactions' with 'transactions' selected. The main area shows an aggregation pipeline:

```

1 [ ]
2   {
3     $group: {
4       _id: "$merchant_id",
5       total_amount: {
6         $sum: "$amount"
7       },
8       transaction_count: {
9         $sum: 1
10      }
11    }
12  },
13  {
14    $sort: {
15      transaction_count: -1,
16      total_amount: -1
17    }
18  }
19 ]

```

The 'PIPELINE OUTPUT' section shows sample documents from the aggregation:

- \_id: "merch\_36" total\_amount: 193510673 transaction\_count: 182
- \_id: "merch\_20" total\_amount: 179895207 transaction\_count: 182
- \_id: "merch\_27" total\_amount: 190923852 transaction\_count: 179
- \_id: "merch\_17" total\_amount: 182897503 transaction\_count: 179
- \_id: "merch\_6" total\_amount: 182255691 transaction\_count: 177

# generate query

The screenshot shows the Compass MongoDB interface. On the left, the connection tree displays 'Darooghe\_transactions' with 'transactions' selected. The main area shows a search query in the search bar:

```
[{"date": "2025-04-21", "merchant_id": "merch_50"}]
```

The results show two documents:

```

_id: ObjectId('6806730ee86138fce2496bd5')
transaction_id: "23734aaaf-78bb-41be-87e7-af4cc7bae515"
timestamp: "2025-04-21T09:16:58.300551Z"
customer_id: "cust_217"
merchant_id: "merch_50"
merchant_category: "transportation"
payment_method: "online"
amount: 106988
location: Object
device_info: Object
status: "approved"
commission_type: "progressive"
commission_amount: 2139
vat_amount: 9628
total_amount: 118747
customer_type: "business"
risk_level: 1
failure_reason: null
date: "2025-04-21"

_id: ObjectId('6806730ee86138fce2496bd6')
transaction_id: "f95be45e-2cb0-48df-8307-62902a4f1fb"
timestamp: "2025-04-21T07:08:01.741900Z"
customer_id: "cust_83"
merchant_id: "merch_50"
merchant_category: "retail"
payment_method: "online"
amount: 253295
location: Object
device_info: Object
status: "declined"

```

## Daily summery pipeline:

```
[  
 {  
 $group: {  
 _id: {  
 merchant_id: "$merchant_id",  
 date: "$date"  
 },  
 total_amount: { $sum: "$amount" },  
 txn_count: { $sum: 1 }  
 }  
 },  
 { $sort: { "_id.date": 1 } }  
 ]
```

## Commission report pipeline:

```
[  
 {  
 $group: {  
 _id: {  
 merchant_category: "$merchant_category",  
 date: "$date"  
 },  
 total_commission: {  
 $sum: "$commission_amount"  
 },  
 avg_commission: {  
 $avg: "$commission_amount"  
 }  
 },  
 ]
```

```
{ $sort: { "_id.date": 1 } }
```

```
]
```

## Monthly Summary:

```
[  
 {  
   $group: {  
     _id: {  
       merchant_id: "$merchant_id",  
       month: { $substr: ["$date", 0, 7] } // gives "YYYY-MM"  
     },  
     total_amount: { $sum: "$amount" },  
     txn_count: { $sum: 1 }  
   }  
 }  
 ]
```

## Customer\_pipeline

```
[  
 {  
   "$group": {  
     "_id": "$customer_id",  
     "transaction_count": { "$sum": 1 },  
     "total_amount": { "$sum": "$amount" }  
   }  
 },  
 {  
   "$addFields": {  
     "segment": {  
       "$switch": {
```

```

        "branches": [
            { "case": { "$gte": ["$transaction_count", 15] }, "then": "High" },
            { "case": { "$gte": ["$transaction_count", 10] }, "then": "Medium" }
        ],
        "default": "Low"
    }
}
},
{ "$sort": { "transaction_count": -1 } } // Sort the results by date (ascending)
]

```

**We need to change the version of java to 17 or 11 in order to make it compatible with spark**

**java version change:**

```
(venv) (base) alireza@alirezas-MacBook-Air DS-CA2 % java -version
openjdk version "23.0.2" 2025-01-21
OpenJDK Runtime Environment Homebrew (build 23.0.2)
OpenJDK 64-Bit Server VM Homebrew (build 23.0.2, mixed mode, sharing)
(venv) (base) alireza@alirezas-MacBook-Air DS-CA2 % /usr/libexec/java_home -
```

Matching Java Virtual Machines (4):

```
23.0.2 (arm64) "Homebrew" - "OpenJDK 23.0.2" /opt/homebrew/Cellar/openj
17.0.14 (arm64) "Homebrew" - "OpenJDK 17.0.14" /opt/homebrew/Cellar/openj
11.0.23 (arm64) "Amazon.com Inc." - "Amazon Corretto 11" /Library/Java/JavaV
1.8.441.07 (arm64) "Oracle Corporation" - "Java" /Library/Internet Plug-Ins/Jav
/opt/homebrew/Cellar/openjdk/23.0.2/libexec/openjdk.jdk/Contents/Home
```

```
(venv) (base) alireza@alirezas-MacBook-Air DS-CA2 % export JAVA_HOME="/Li
export PATH="$JAVA_HOME/bin:$PATH"
```

```
(venv) (base) alireza@alirezas-MacBook-Air DS-CA2 % java -version  
openjdk version "11.0.23" 2024-04-16 LTS  
OpenJDK Runtime Environment Corretto-11.0.23.9.1 (build 11.0.23+9-LTS)  
OpenJDK 64-Bit Server VM Corretto-11.0.23.9.1 (build 11.0.23+9-LTS, mixed mod
```

Always change the java version to 11 when running pyspark. and you don't need any anacondo diactivated.

For changing to java 17, you can do this

```
(venv) (base) alireza@alirezas-MacBook-Air DS-CA2 % export JAVA_HOME="/opt/jdk-17.0.2_12"  
export PATH="$JAVA_HOME/bin:$PATH"
```

## Forcing to change java version in all terminals

As you can see, I added those lines in the .zshrc file at the end(comment java 11 and now I am using version 17 and if comment both version 11 and 17, it will be java version 23

```

#!/bin/zsh
# _ZSH_THEME="simple"
# >>> conda initialize >>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/Users/alireza/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/Users/alireza/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/Users/alireza/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/Users/alireza/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<

export PATH="/opt/homebrew/Cellar/antlr/4.13.2/bin:$PATH"

__MY_VMOPTIONS_SHELL_FILE="${HOME}/.jetbrains.vmoptions.sh"; if [ -f "${__MY_VMOPTIONS_SHELL_FILE}" ]; then . "${__MY_VMOPTIONS_SHELL_FILE}"; fi
export PATH="/opt/homebrew/opt/openjdk@17/bin:$PATH"
export ARTEMIS_HOME=/opt/artemis
export PATH="$ARTEMIS_HOME/bin:$PATH"
export JAVA_HOME=$(/usr/libexec/java_home -v 23)
export PATH="$JAVA_HOME/bin:$PATH"

# export JAVA_HOME="/Library/Java/JavaVirtualMachines/amazon-corretto-11.jdk/Contents/Home"
# export PATH="$JAVA_HOME/bin:$PATH"
export JAVA_HOME="/opt/homebrew/Cellar/openjdk@17/17.0.14/libexec/openjdk.jdk/Contents/Home"
export PATH="$JAVA_HOME/bin:$PATH"

```

# force to java 11:

To ensure that Java 11 is the default version instead of Java 23, follow these steps:

## 1. Find the Path to Java 11

From your list, you have Java 11 installed at:

## 2. Update `.zshrc` to Set Java 11 as Default

You need to modify your shell configuration file (`~/.zshrc`) to point to Java 11 as the default version. Open the file in a text editor:

```

bash
CopyEdit
nano ~/.zshrc

```

Then, add the following lines to the file, which will set Java 11 as the default when you open a new terminal:

```
bash
CopyEdit
export JAVA_HOME="/Library/Java/JavaVirtualMachines/amazon-corretto-11.j
dk/Contents/Home"
export PATH="$JAVA_HOME/bin:$PATH"
```

### 3. Apply Changes to `.zshrc`

After saving the changes, apply them by either restarting your terminal or using:

```
bash
CopyEdit
source ~/.zshrc
```

### 4. Verify the Java Version

Once you've updated and sourced your `.zshrc`, verify that Java 11 is now the default by running:

```
bash
CopyEdit
java -version
```

This should show Java 11 as the active version.

#### Optional: Check the Java Versions Installed

If you'd like to see all installed Java versions and ensure that the `java` command points to Java 11, you can run:

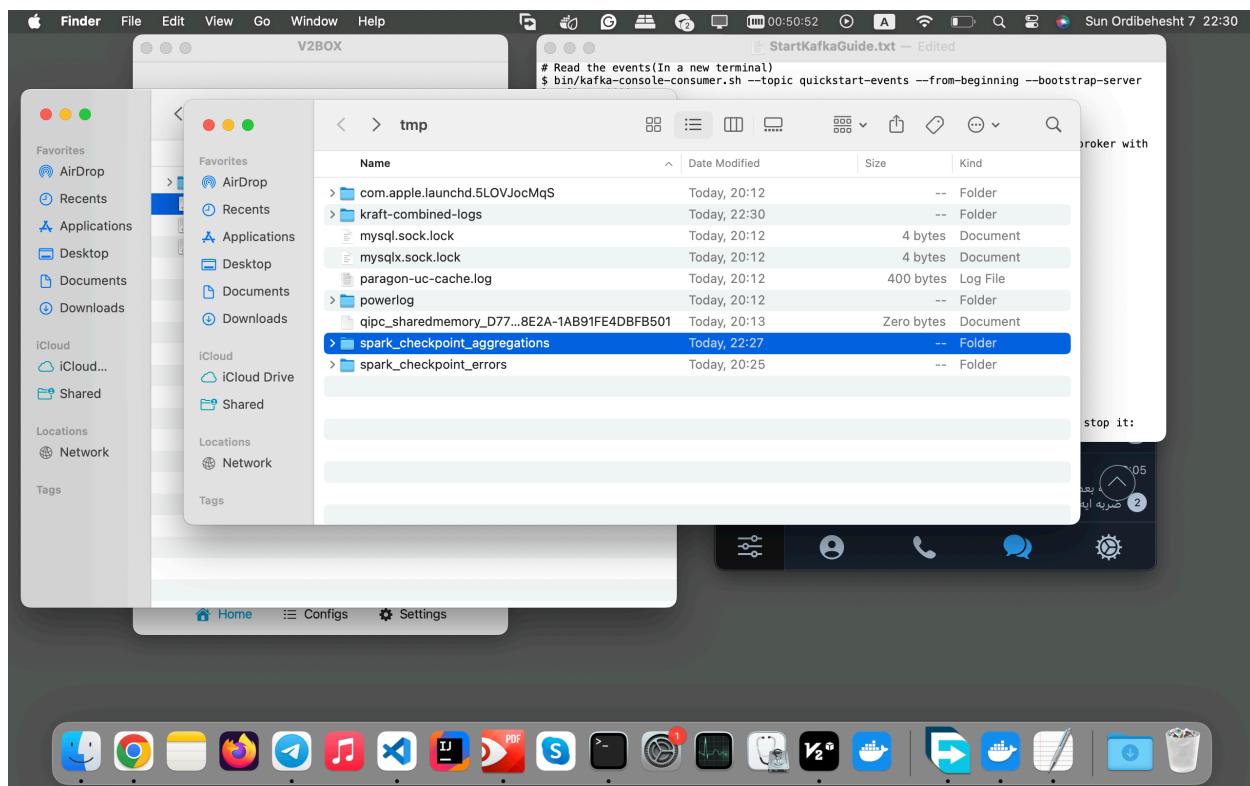
```
bash
CopyEdit
```

```
/usr/libexec/java_home -V
```

This will list all Java versions, and you can confirm that the `JAVA_HOME` path is correct.

Let me know if you encounter any issues!

## Remove the `/tmp/spark_checkpoints` files created at real time processing part

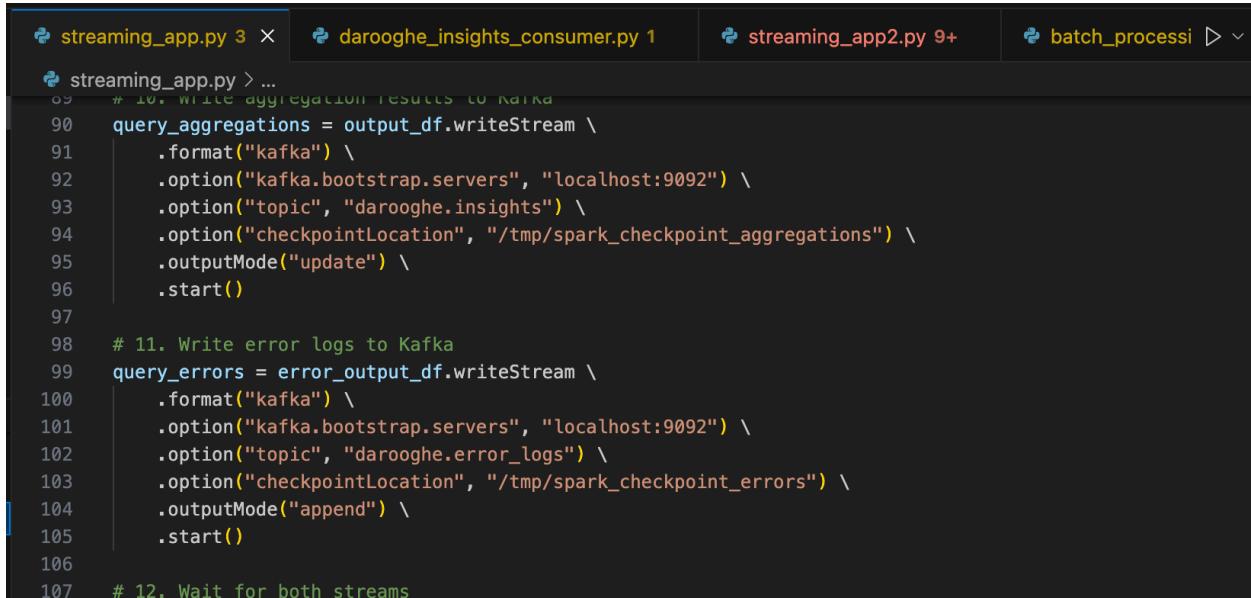


## Darooghe.insights for the output of the first part in real time processing layer

Spark Streaming Application

- I. Implement a Spark Streaming application that connects to Kafka Consumer.
- II. Process data in small time intervals (micro-batches), using time-based windows to aggregate and analyze data over specific periods (e.g., 1 minute), with the window sliding at regular intervals (e.g., every 20 seconds), to ensure timely and efficient data processing in real time. Use the processing to gain some informative insights. Write detected insights to an (or multiple) arbitrary topic (topics).
- III. Implement a checkpoint mechanism for fault tolerance.

**if you run `stream_app.py`, it will create output into `transactions.insight` and**



```

streaming_app.py 3 ×      derooghe_insights_consumer.py 1      streaming_app2.py 9+      batch_processi ▶ ▾
streaming_app.py > ...
  # 10. Write aggregation results to Kafka
90  query_aggregations = output_df.writeStream \
91    .format("kafka") \
92    .option("kafka.bootstrap.servers", "localhost:9092") \
93    .option("topic", "darooghe.insights") \
94    .option("checkpointLocation", "/tmp/spark_checkpoint_aggregations") \
95    .outputMode("update") \
96    .start()
97
98 # 11. Write error logs to Kafka
99 query_errors = error_output_df.writeStream \
100   .format("kafka") \
101   .option("kafka.bootstrap.servers", "localhost:9092") \
102   .option("topic", "darooghe.error_logs") \
103   .option("checkpointLocation", "/tmp/spark_checkpoint_errors") \
104   .outputMode("append") \
105   .start()
106
107 # 12. Wait for both streams

```

**And this is the result of its consumer:**

The screenshot shows a Jupyter Notebook interface with several code cells and a terminal output pane.

**Code Cells:**

- streaming\_app.py**: Contains code for creating a Kafka consumer and defining a function to poll and print messages from the 'darooghe.insights' topic.
- darooghe\_insights\_consumer.py**: Contains code for creating a Kafka consumer, defining a function to poll and print messages, and a main loop that prints received messages.
- streaming\_app2.py**: Contains code for creating a Kafka consumer and defining a function to poll and print messages.
- batch\_processi**: Contains code for creating a Kafka consumer and defining a function to poll and print messages.

**Terminal Output:**

```

action_count': 8, 'total_amount_sum': 10215489.0}
Received: {'window_start': '2025-04-27T20:55:40.000+03:30', 'window_end': '2025-04-27T20:56:40.000+03:30', 'merchant_category': 'entertainment', 'trans
action_count': 26, 'total_amount_sum': 32948957.0}
Received: {'window_start': '2025-04-27T20:56:00.000+03:30', 'window_end': '2025-04-27T20:57:00.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 11, 'total_amount_sum': 16091557.0}
Received: {'window_start': '2025-04-27T20:55:40.000+03:30', 'window_end': '2025-04-27T20:56:40.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 31, 'total_amount_sum': 36328313.0}
Received: {'window_start': '2025-04-27T20:55:20.000+03:30', 'window_end': '2025-04-27T20:56:20.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 48, 'total_amount_sum': 53819237.0}
Received: {'window_start': '2025-04-27T20:55:40.000+03:30', 'window_end': '2025-04-27T20:56:40.000+03:30', 'merchant_category': 'food_service', 'transa
ction_count': 28, 'total_amount_sum': 23739395.0}
Received: {'window_start': '2025-04-27T20:55:20.000+03:30', 'window_end': '2025-04-27T20:56:20.000+03:30', 'merchant_category': 'food_service', 'transa
ction_count': 43, 'total_amount_sum': 52607524.0}
Received: {'window_start': '2025-04-27T20:56:00.000+03:30', 'window_end': '2025-04-27T20:57:00.000+03:30', 'merchant_category': 'food_service', 'transa
ction_count': 7, 'total_amount_sum': 10578032.0}
Received: {'window_start': '2025-04-27T22:27:28.000+03:30', 'window_end': '2025-04-27T22:28:20.000+03:30', 'merchant_category': 'food_service', 'transa
ction_count': 3, 'total_amount_sum': 3181055.0}
Received: {'window_start': '2025-04-27T22:27:40.000+03:30', 'window_end': '2025-04-27T22:28:40.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 6, 'total_amount_sum': 88925932.0}
Received: {'window_start': '2025-04-27T22:27:00.000+03:30', 'window_end': '2025-04-27T22:28:00.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 6, 'total_amount_sum': 8892592.0}
Received: {'window_start': '2025-04-27T22:27:20.000+03:30', 'window_end': '2025-04-27T22:28:20.000+03:30', 'merchant_category': 'transportation', 'tran
saction_count': 6, 'total_amount_sum': 8892592.0}

```

**Bottom Status Bar:**

Ln 9, Col 29 Spaces: 4 UTF-8 LF {} Python 3.13.2 64-bit Go Live

for fraud\_Detection part, the “fraud\_detection.py” create the output into topic “darooghe.fraud\_alerts” and the consumer file is “darooghe\_fraud\_detection\_consumer.py”

The screenshot shows a Jupyter Notebook interface with several code cells and a terminal output window.

- EXPLORER:** Shows files like streaming\_app.py, batch\_processing.py, real\_time.ipynb, fraud\_detection.py, derooghe\_pulse.py, and derooghe.fraud\_alerts.ipynb.
- OPEN EDITORS:** Shows fraud\_detection.py, which contains Python code for fraud detection based on geographical impossibility.
- TERMINAL:** Shows a list of fraud detections with transaction IDs and alert types.
- PROBLEMS:** Shows a list of errors and warnings.
- OUTPUT:** Shows the terminal output of the fraud detection logic.
- DEBUG CONSOLE:** Shows a list of available kernels: python, python3.11, zsh, zsh, python3.11, and python3.11.
- PORTS:** Shows port numbers 4, 2, and 42.

```

9+ # 6. Fraud Rule 2: Geographical Impossibility
89
90
91     # Join the dataframes with itself on customer_id within 5 minutes
92     geo_join = df_parsed.alias("a").join(
93         df_parsed.alias("b"),
94         (col("a.customer_id") == col("b.customer_id")) &
95         (col("a.timestamp") < col("b.timestamp")) &
96         (unix_timestamp(col("b.timestamp")) - unix_timestamp(col("a.timestamp")) <= 300)
97     )
98
99     geo_alerts = geo_join \
100     .withColumn("distance", haversine_udf(
101         col("a.location.lat"), col("a.location.lng"),
102         col("b.location.lat"), col("b.location.lng")
103     )) \
104     .filter(col("distance") > 5) \
105     .select(
106
107     Fraud Detected: {'customer_id': 'cust_928', 'transaction_id': '7ccb11f4-88d0-433f-a3e1-b2d915216349', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
108     Fraud Detected: {'customer_id': 'cust_928', 'transaction_id': '41b643e-31e3-405c-bb42-6e864f64a3d', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
109     Fraud Detected: {'customer_id': 'cust_101', 'transaction_id': '3b56e80-6cf2-4d15-a2bf-33dd1e17042a4', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
110     Fraud Detected: {'customer_id': 'cust_101', 'transaction_id': '0b935bae-793b-4809-b166-27e902c9286', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
111     Fraud Detected: {'customer_id': 'cust_101', 'transaction_id': '3b56e80-6cf2-4d15-a2bf-33dd1e17042a4', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
112     Fraud Detected: {'customer_id': 'cust_281', 'transaction_id': '2d889a99-c3e0-4676-a48d-26f5895c77bd', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
113     Fraud Detected: {'customer_id': 'cust_387', 'transaction_id': 'e1bde23db-5c5f-4102-b70b-395987172bf', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
114     Fraud Detected: {'customer_id': 'cust_387', 'transaction_id': '79605535-1a33-44d0-b5b4-a142d7e850', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
115     Fraud Detected: {'customer_id': 'cust_928', 'transaction_id': '41b643e-31e3-405c-bb42-6e864f64a3d', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
116     Fraud Detected: {'customer_id': 'cust_223', 'transaction_id': 'd13274df-026e-4363-b850-42e2df42cea', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
117     Fraud Detected: {'customer_id': 'cust_529', 'transaction_id': '5a1ce19f-5c16-493b-b8dc-39b9506f505', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
118     Fraud Detected: {'customer_id': 'cust_299', 'transaction_id': '42ed7ed9-b666-43c6-be85-adc02d224fa1', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
119     Fraud Detected: {'customer_id': 'cust_299', 'transaction_id': 'fce2655b-35f-45a7-a988-3bac7b9e1e53', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
120     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': '12f6094b-a2b4-49a9-a35a-78369a1a190', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
121     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': 'de328f94-b8b4-497b-aefcb-130e09ef7a7b', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
122     Fraud Detected: {'customer_id': 'cust_299', 'transaction_id': '4f74626f-7d39-4e33-9974-ce7864222525', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
123     Fraud Detected: {'customer_id': 'cust_261', 'transaction_id': '59fec02c-946d-4984-a3b0-9ae78f254e9', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
124     Fraud Detected: {'customer_id': 'cust_928', 'transaction_id': '08721cc7-6b47-456c-be8b-980c1b504321', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
125     Fraud Detected: {'customer_id': 'cust_261', 'transaction_id': '2b1b3789-9473-44af-a167-2bb21d1ead', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
126     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': '10373789-4d6e-4491-9d49-f692216867f', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
127     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': '831b63a7-fa55-43c6-8fcfa-a67eef2ab19', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
128     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': '9768f554-c790-41a4-a632-a70fbce952a5', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}
129     Fraud Detected: {'customer_id': 'cust_207', 'transaction_id': '831b63a7-fa55-43c6-8fcfa-a67eef2ab19', 'alert_type': 'GEO_IMPOSSIBLE_ALERT'}

```

In the commission\_analytics.py, I created other three topics said in the :

## Real-Time Commission Analytics

- Calculate and write to an (or multiple) arbitrary topic (topics), real-time metrics for commissions:
  - Total commission by type per minute.
  - Commission ratio (commission/transaction amount) by merchant category.
  - Highest commission-generating merchants in 5-minute windows.

for each i created a consumer:

```

commission_by_type_consumer.py > ...
1  from kafka import KafkaConsumer
2  import json
3
4  consumer = KafkaConsumer(
5      'darooqhe.commission_by_type',
6      bootstrap_servers='localhost:9092',
7      value_deserializer=lambda x: json.loads(x.decode('utf-8')),
8      auto_offset_reset='latest',
9      enable_auto_commit=True
10 )
11
12 print("Listening to darooqhe.commission_by_type...")
13 for message in consumer:
14     print(message.value)
15

```

PROBLEMS 66 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

```

{'commission_type': 'flat', 'total_commission': 1775785.0, 'window_start': '2025-04-28T14:03:00.000+03:30', 'window_end': '2025-04-28T14:04:00.000+03:30'}
{'commission_type': 'flat', 'total_commission': 500356.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'tiered', 'total_commission': 384069.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'progressive', 'total_commission': 494534.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'flat', 'total_commission': 913347.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'tiered', 'total_commission': 735050.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'progressive', 'total_commission': 875751.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'flat', 'total_commission': 1104541.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'tiered', 'total_commission': 889138.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'progressive', 'total_commission': 1113346.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'commission_type': 'flat', 'total_commission': 1431853.0, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
[]

Ln 5, Col 15 (18 selected) Spaces: 4 UTF-8 LF {} Python ⚙ 3.13.2 64-bit ⚙ Go Live ⚙

```

```

commission_ratio_consumer.py > ...
1  from kafka import KafkaConsumer
2  import json
3
4  consumer = KafkaConsumer(
5      'darooqhe.commission_ratio',
6      bootstrap_servers='localhost:9092',
7      value_deserializer=lambda x: json.loads(x.decode('utf-8')),
8      auto_offset_reset='latest',
9      enable_auto_commit=True
10 )
11
12 print("Listening to darooqhe.commission_ratio...")
13 for message in consumer:
14     print(message.value)
15

```

```

{'merchant_category': 'entertainment', 'commission_ratio': 0.01999951120217547, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'transportation', 'commission_ratio': 0.0199995117400873, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'food_service', 'commission_ratio': 0.019999507374402873, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'government', 'commission_ratio': 0.019999450937539073, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'retail', 'commission_ratio': 0.01999949047437268, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'entertainment', 'commission_ratio': 0.019999474207044017, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'transportation', 'commission_ratio': 0.019999560459679572, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'food_service', 'commission_ratio': 0.019999505640344372, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'government', 'commission_ratio': 0.019999410987289705, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'retail', 'commission_ratio': 0.019999497229140042, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
{'merchant_category': 'entertainment', 'commission_ratio': 0.019999464539102032, 'window_start': '2025-04-28T14:04:00.000+03:30', 'window_end': '2025-04-28T14:05:00.000+03:30'}
[]

Ln 15, Col 1 Spaces: 4 UTF-8 LF {} Python ⚙ 3.13.2 64-bit ⚙ Go Live ⚙

```

The screenshot shows a code editor interface with multiple tabs open. The tabs include:

- commission\_by\_type\_consumer.py 1
- commission\_ratio\_consumer.py 1
- top\_merchants\_consumer.py 1
- Generate
- Simulate

The left sidebar shows a file tree with several Python files and notebooks:

- fraud\_detection.py
- commission\_analytics.py
- all\_consumers.ipynb
- commission\_by\_type\_cons... 1
- commission\_ratio\_consum... 1
- DS-CA2
- real\_time.ipynb
- streaming\_app.py
- streaming\_app2.py
- top\_merchants\_consumer.py 1
- transactions.json

The main editor area displays the content of the top\_merchants\_consumer.py script:

```
from kafka import KafkaConsumer
import json

consumer = KafkaConsumer(
    'darooghe.top_merchants',
    bootstrap_servers='localhost:9092',
    value_deserializer=lambda x: json.loads(x.decode('utf-8')),
    auto_offset_reset='latest',
    enable_auto_commit=True
)

print("Listening to derooghe.top_merchants...")
for message in consumer:
    print(message.value)
```

The bottom status bar shows:

- PROBLEMS 66
- OUTPUT
- DEBUG CONSOLE
- TERMINAL
- PORTS
- JUPYTER
- Ln 15, Col 1
- Spaces: 4
- UTF-8 LF
- Python
- 3.13.2 64-bit
- Go Live

# Visualization

Figure 1

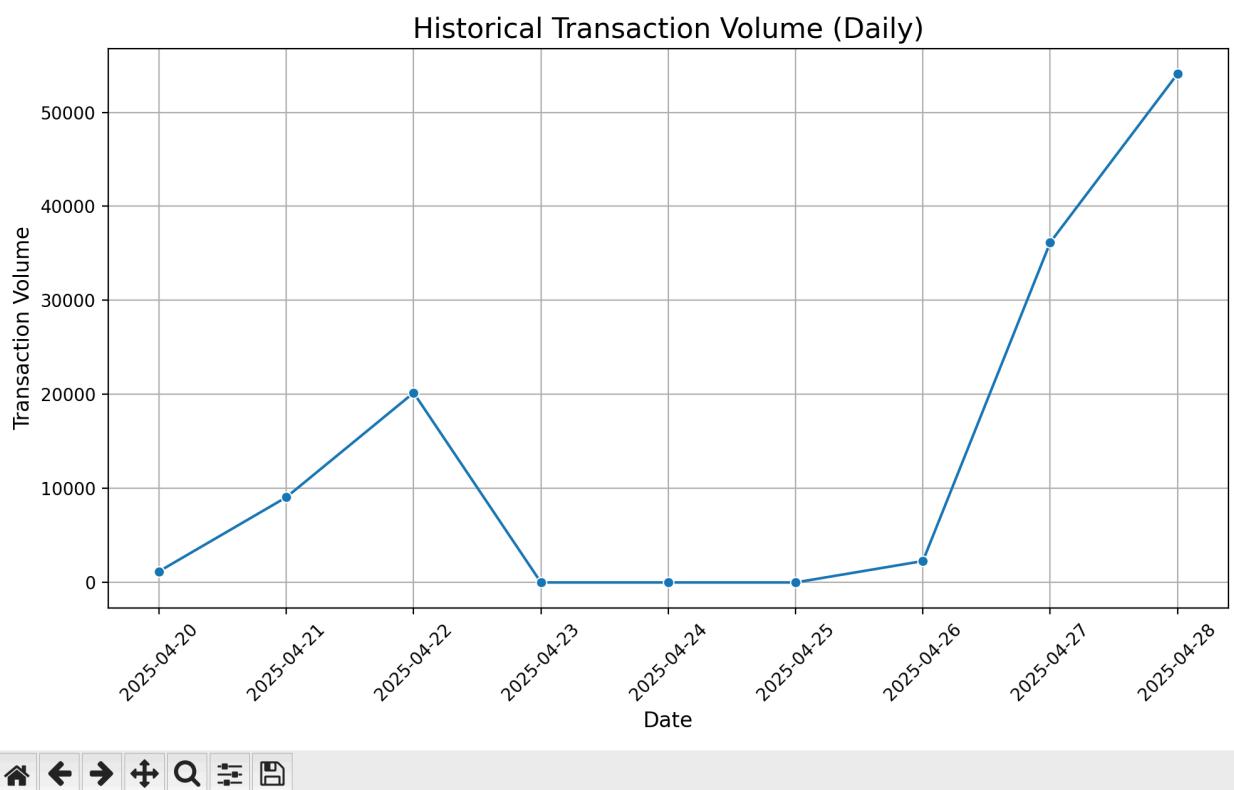
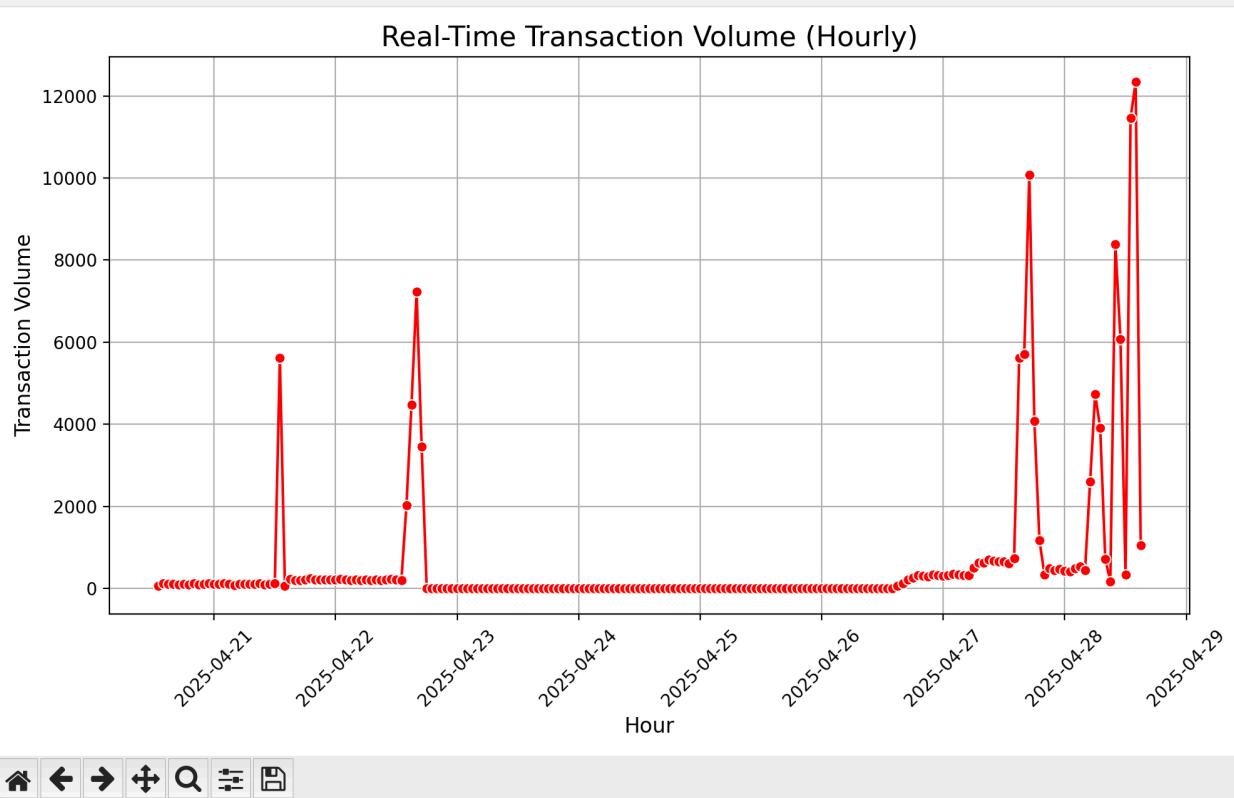


Figure 1



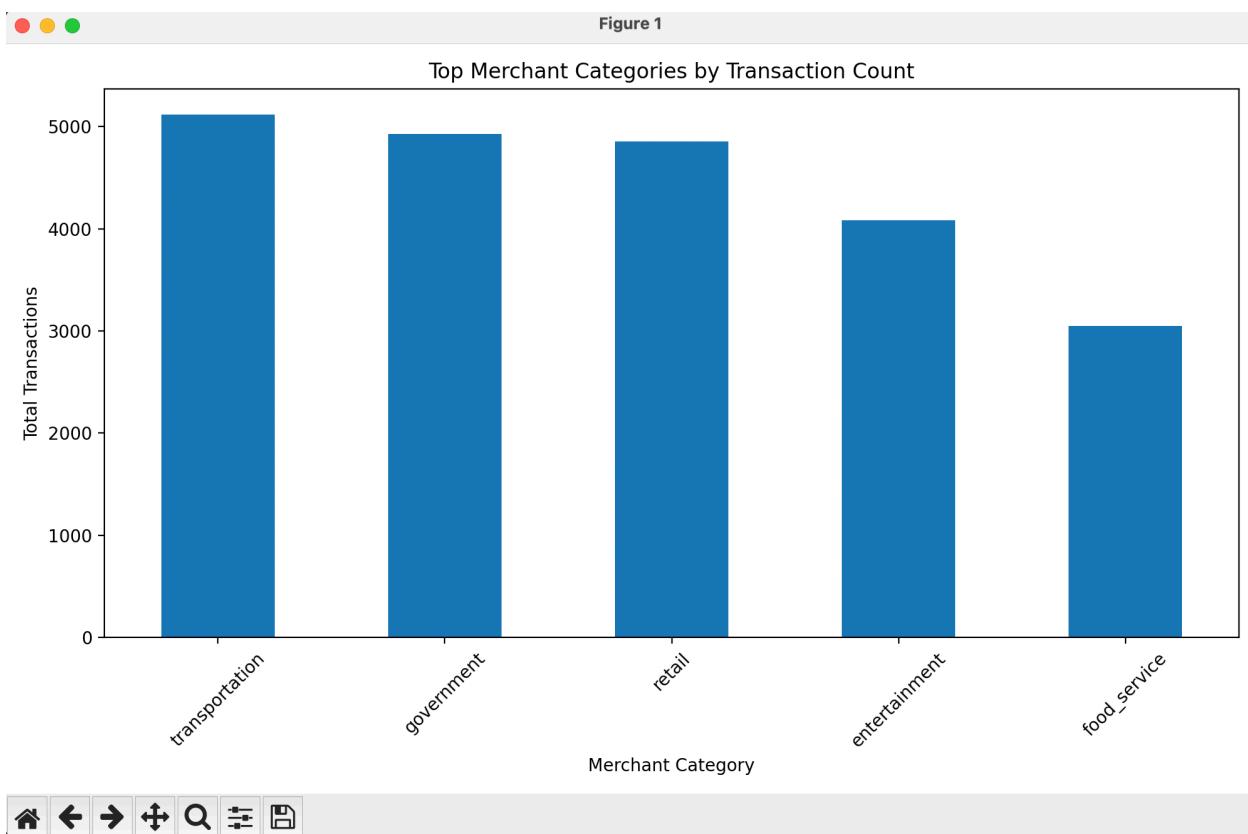
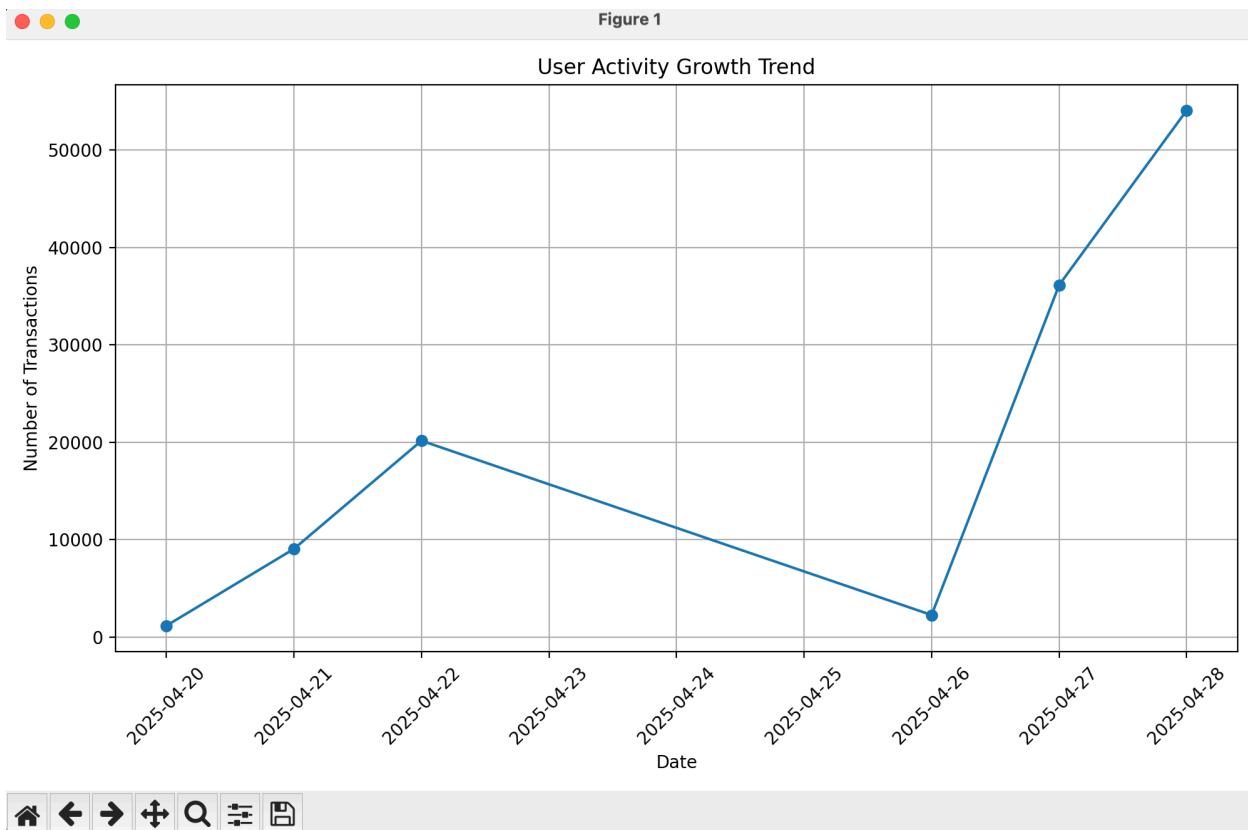




Figure 1

### Commission Distribution by Type

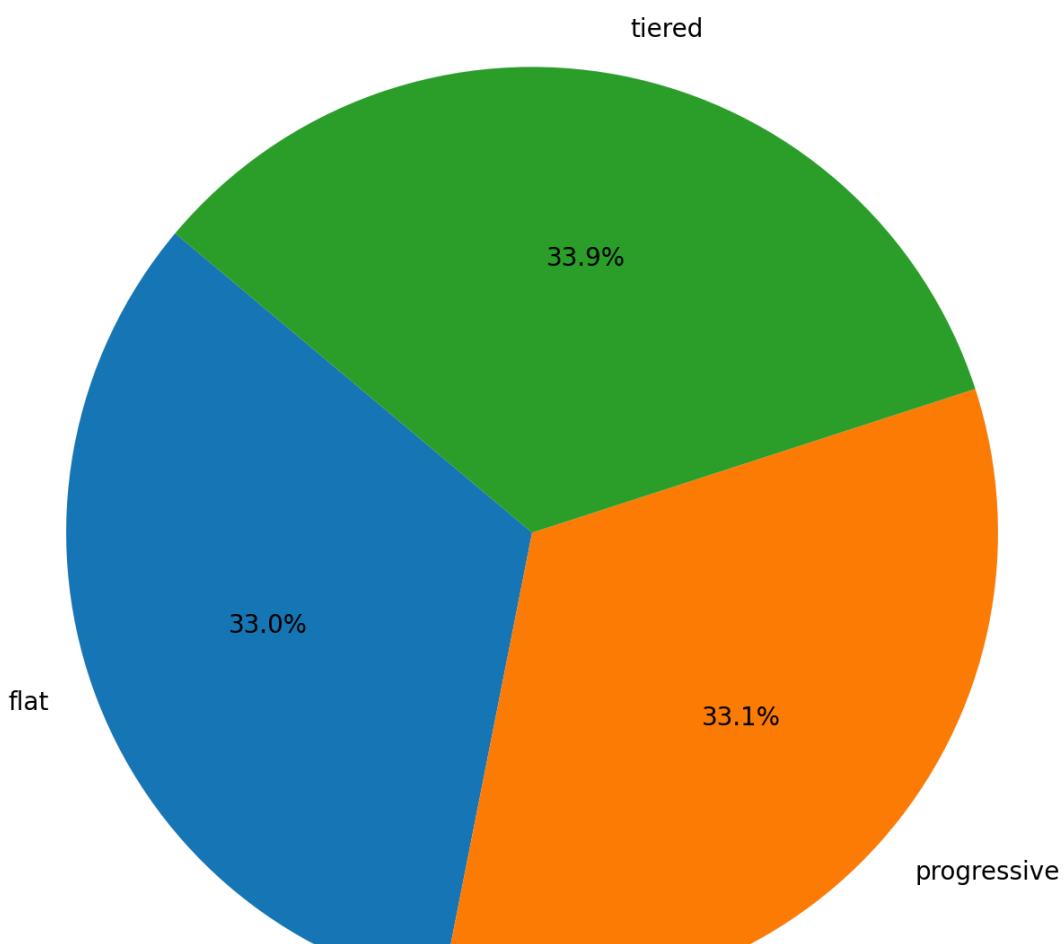




Figure 1

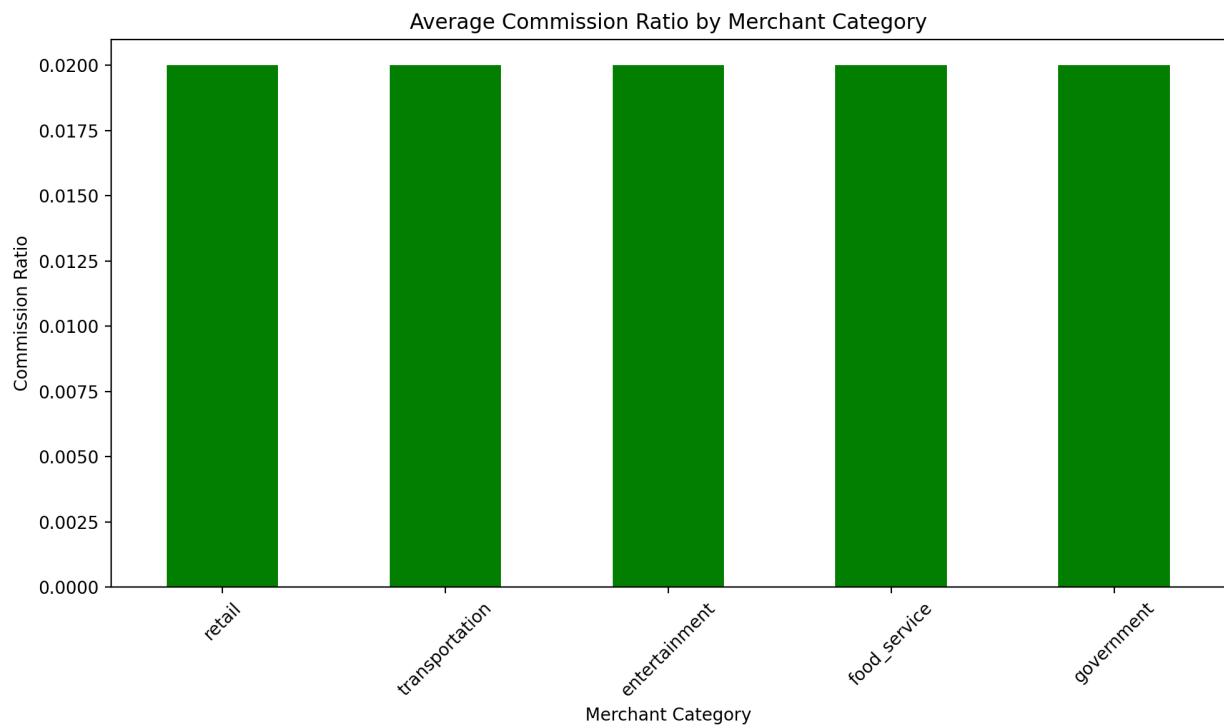
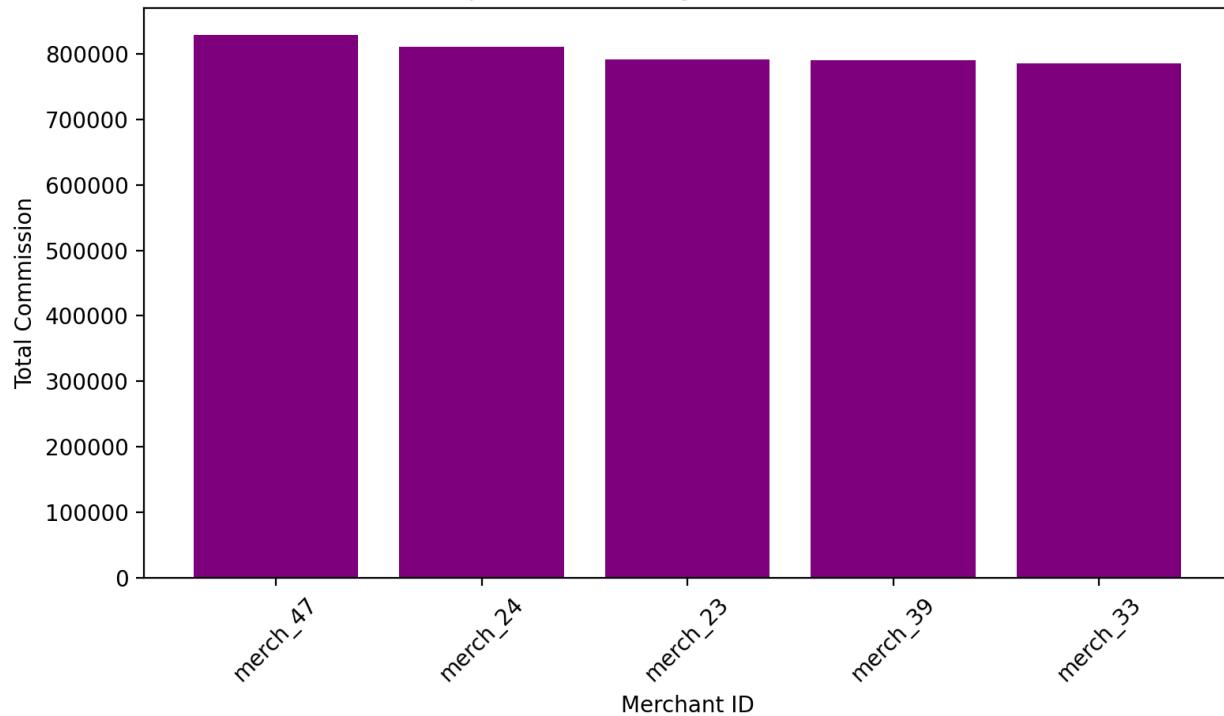
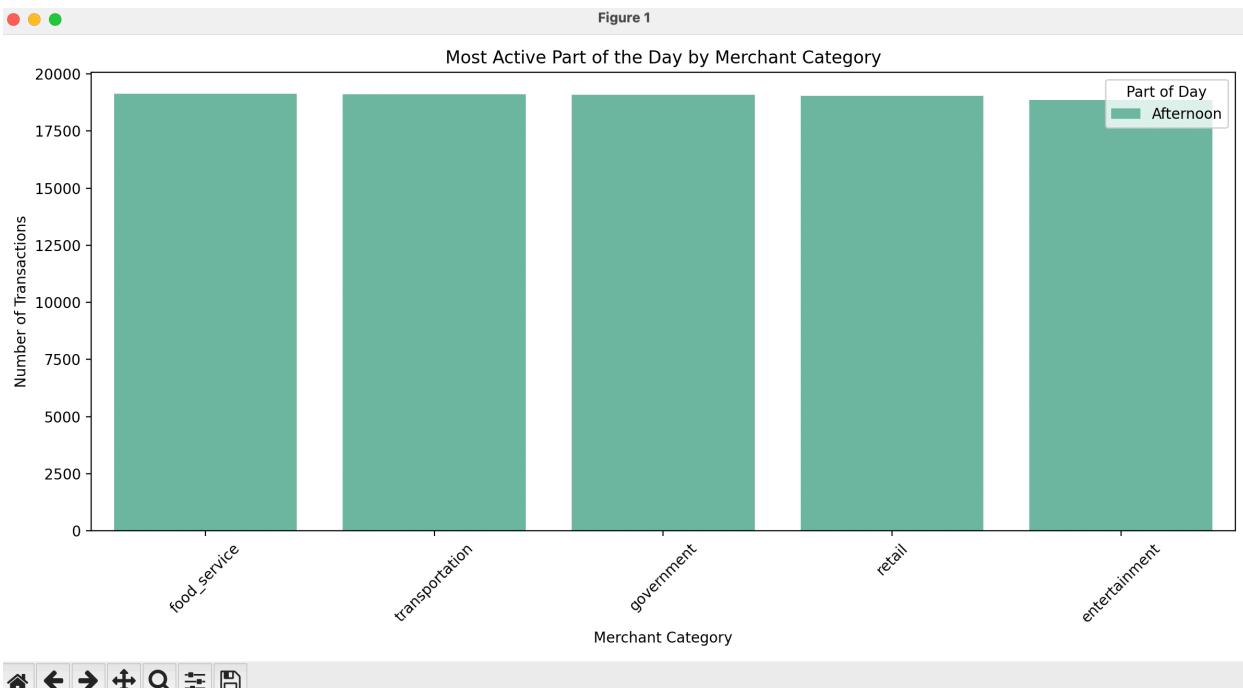
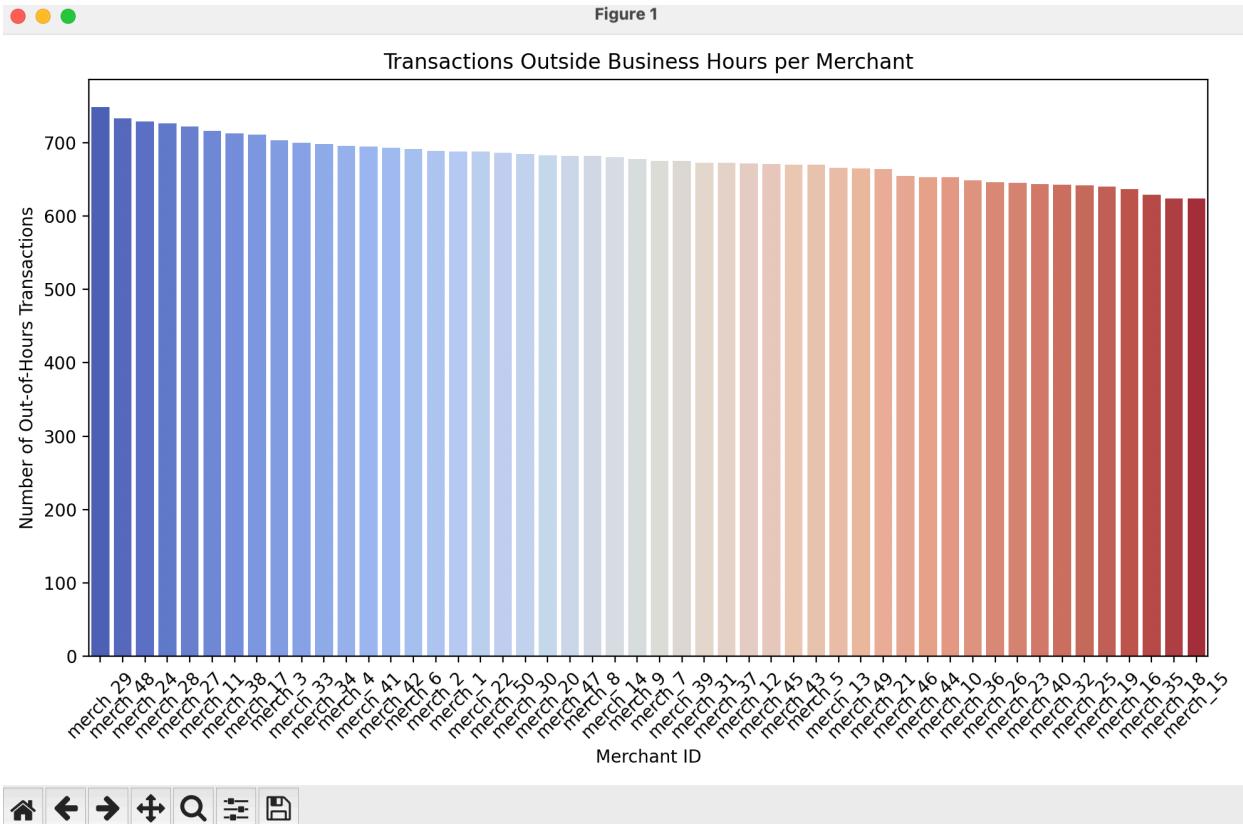


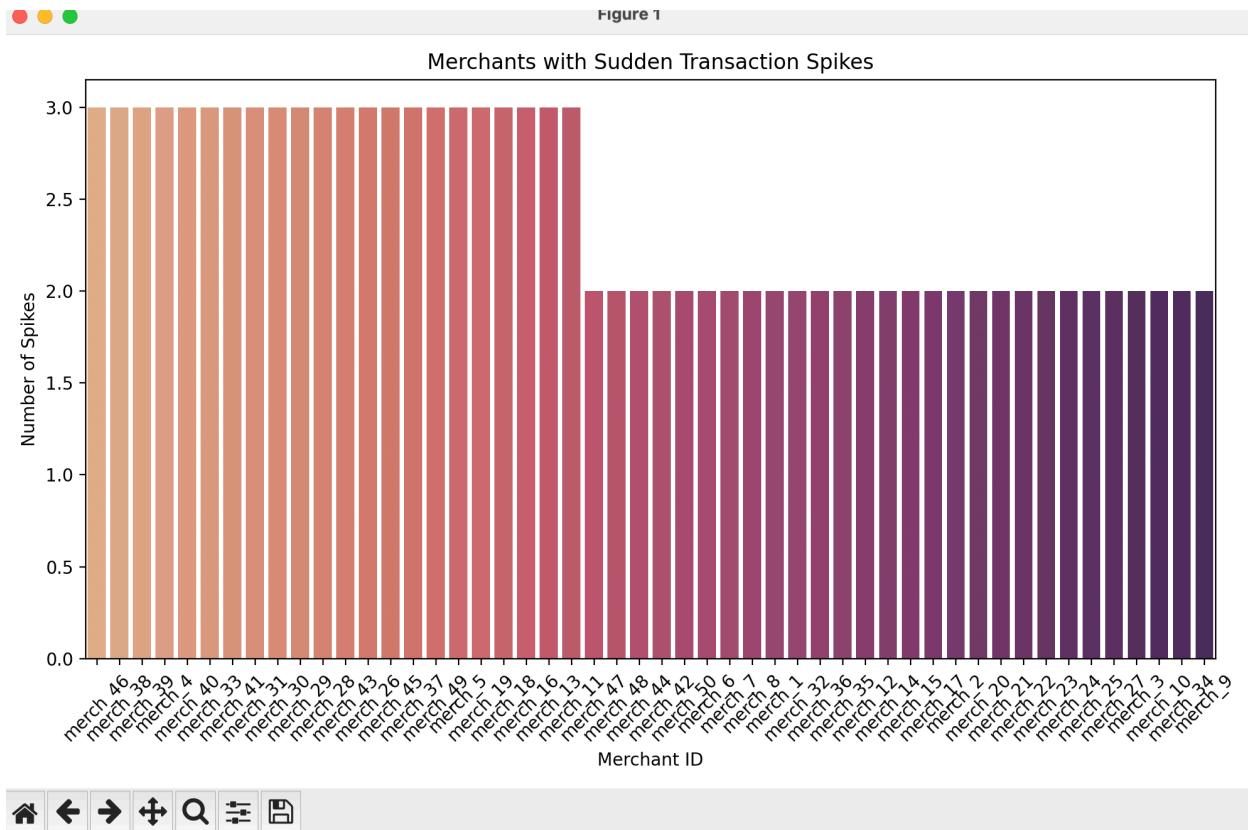


Figure 1

### Top 5 Merchants by Total Commission







## Commission Audit System (5 pts)

### Dynamic Pricing Simulator

I. Create PySpark UDF to recommend optimal commission type per transaction. (3 pts)

II. Validate against historical profitability data. (2 pts)

here is the code :

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import udf, col
from pyspark.sql.types import StringType

# Initialize Spark session
```

```

spark = SparkSession.builder.appName('CommissionSimulation').getOrCreate()

# Sample transaction data (replace with your actual DataFrame)
# data = [
#     {"transaction_id": "e64a5870-2d82-4d68-8be0-6850afa10d67", "timestamp": "2023-01-01T08:00:00Z", "amount": 1000000, "category": "food_service", "is_valid": true},
#     {"transaction_id": "a652709f-426a-4bee-b216-ead917d71118", "timestamp": "2023-01-01T09:00:00Z", "amount": 500000, "category": "retail", "is_valid": false},
#     # Add more transaction data...
# ]

# Create Spark DataFrame from the data (replace this step if using existing df_transactions)
df_transactions = spark.createDataFrame(df_transactions)

# Define UDF to recommend the commission type
def recommend_commission_type(amount, category):
    if amount > 1000000:
        if category in ['food_service', 'retail']:
            return 'tiered'
        else:
            return 'progressive'
    else:
        return 'flat'

# df_out_of_hours
# Register the UDF
recommend_commission_type_udf = udf(recommend_commission_type, StringType())

# Apply UDF to the DataFrame to recommend commission type
df_spark = df_transactions.withColumn('recommended_commission_type',
                                       recommend_commission_type_udf('amount', 'merchant_category'))

# Show DataFrame with recommended commission type
df_spark.show()

# Validate the recommended commission type by comparing with actual commission type
df_spark = df_spark.withColumn('is_valid',
                               (col('recommended_commission_type') == col('commission_type')) & df_spark['is_valid'])

```

```
# Show validation results (checking if the recommendation matches the actual commission type)
df_spark.select('transaction_id', 'recommended_commission_type', 'commission_amount').show()

# Optionally, calculate profitability difference (recommended commission vs. actual commission)
df_spark = df_spark.withColumn('profitability_diff',
                                col('commission_amount') - col('recommended_commission_amount'))

# Show profitability comparison
df_spark.show()
```

## this is the output:

transaction_id	customer_id	merchant_id	merchant_category	payment_method	amount	location	device_info	status	commission_type	commission_amount	vat_amount	total_amount	customer_type	risk_level	failure_reason	transaction_hour	part_of_day	recommended_commission_type
87edfe0c-63ff-46e...	cust_498	merch_4	entertainment	mobile	681670	{lng > 51.353940...}	{device_model > ...}	approved	tiered									
ed13633	61350	756653	CIP	2	NULL	21	Evening		flat									
dcbb129ca-f8f9-413...	cust_169	merch_42	entertainment	pos	5154059	{lng > 51.334607...}			{} approved	progressive								
at 30281	136265	1689605	CIP	1	NULL	9	Morning		flat									
77db852e-bce6-446...	cust_879	merch_12	food_service	online	499205	{lng > 51.337542...}	{device_model > ...}	approved	progressive									
ve 9984	44928	554117	business	1	NULL	17	Afternoon		flat									
acc8d68a-ab8f-48c...	cust_28	merch_26	retail	mobile	257576	{lng > 51.379014...}	{device_model > ...}	approved	progressive									
ve 5151	23018	283889	individual	3	NULL	16	Afternoon		flat									
055c9c6b-8863-4a9...	cust_321	merch_49	entertainment	online	1327604	{lng > 51.357866...}	{device_model > ...}	approved	flat									
at 26552	119484	473640	CIP	2	NULL	11	Morning		progressive									
8cce7e3-6b72-40b...	cust_14	merch_43	food_service	online	1785324	{lng > 51.353605...}	{device_model > ...}	approved	tiered									
ed 35706	160679	1981709	individual	3	NULL	9	Morning		tiered									
d2f897da-c94b-47d...	cust_250	merch_30	entertainment	online	1071625	{lng > 51.345109...}	{device_model > ...}	approved	tier									
ed 21432	96446	1189503	individual	1	NULL	6	Morning		progressive									
25f98c1f-d090-4ee...	cust_601	merch_21	government	online	143323	{lng > 51.310085...}	{device_model > ...}	approved	flat									
at 2866	12899	159088	individual	2	NULL	9	Morning		flat									
4002c85b-ed55-4d4...	cust_594	merch_2	retail	nfc	427319	{lng > 51.379845...}			{} approved	tier								
ed 8546	38458	474323	CIP	1	NULL	19	Evening		flat									
eedad34af-637d-45b...	cust_55	merch_36	food_service	pos	1199535	{lng > 51.379554...}			{} approved	tier								
ed 23990	107958	1331483	business	3	NULL	20	Evening		tiered									
df14c80b-36c6-47c...	cust_245	merch_28	retail	pos	487344	{lng > 51.309118...}			{} approved	tier								
ed 9746	43860	549950	CIP	5	NULL	14	Afternoon		flat									
710acb3c-05bf-4f6...	cust_358	merch_10	entertainment	pos	1266571	{lng > 51.331444...}			{} approved	progressive								
ve 25331	113991	1405893	individual	1	NULL	5	Night		progressive									
e065a8f2-278e-473...	cust_485	merch_31	government	nfc	312219	{lng > 51.337480...}			{} approved	progressive								
ve 6244	28099	346562	business	3	NULL	21	Night		flat									
4f6258fa-1be3-464...	cust_961	merch_3	retail	pos	1204533	{lng > 51.335957...}			{} approved	tier								
ed 23996	188407	1332301	CIP	3	NULL	22	Night		tiered									
2883aa5f-df66-4c3...	cust_958	merch_46	transportation	online	461440	{lng > 51.355907...}	{device_model > ...}	approved	tier									
ed 9228	41528	512197	CIP	2	NULL	8	Morning		flat									
32f9ffdb-0b74-4f2...	cust_131	merch_33	retail	nfc	1938942	{lng > 51.319503...}			{} declined	tier								
ed 3878	174504	2152224	business	2 fraud_prevented	5	Night			tiered									
35dfffdcd-2fd4-47f...	cust_263	merch_29	entertainment	pos	871155	{lng > 51.352838...}			{} approved	tier								
ed 17423	78403	966981	individual	1	NULL	0	Night		flat									
728ec00a-e6d4-4fe...	cust_945	merch_47	food_service	pos	1212730	{lng > 51.365379...}			{} approved	tier								

The screenshot shows a Jupyter Notebook interface within VS Code. The top bar displays the title "DS-CA2". The left sidebar shows "OPEN EDITORS" with "commission\_analytics.py" and "all\_consumers.ipynb" open. The main area has tabs for "commission\_ratio\_consumer.py", "consumertopandas.py", "top\_merchants\_consumer.py", and "consumertopandas.py". The "TERMINAL" tab is active, displaying the following command-line output:

```

546 def recommend_commission_type(amount, category):
547     if amount > 1000000:
548         if category in ['food service', 'retail']:
549             ...

```

Below the terminal, there is a large table with columns: transaction\_id, recommended\_commission\_type, commission\_type, and is\_valid. The table contains approximately 20 rows of data. The right side of the interface shows a sidebar with several "zsh" and "python3.11" entries.

**Note: always remove the /tmp/checkpoints file after creating them in the realtime part**

I created different consumer for different parts of realtime

this is a function for creating consumer who waits until a specific time and gets specific number of messages. then turn them to dataframe for working in visualization part

```

def read_topic_to_dataframe(topic_name, max_messages=1000, timeout_ms=50
    """

```

Reads messages from a Kafka topic into a Pandas DataFrame.

Args:

topic\_name (str): Name of the Kafka topic  
max\_messages (int): Maximum number of messages to read  
timeout\_ms (int): How long to wait for messages before timing out (in milliseconds)

Returns:

pd.DataFrame: DataFrame containing the topic data

"""

```
consumer = KafkaConsumer(  
    topic_name,  
    bootstrap_servers='localhost:9092',  
    auto_offset_reset='earliest',  
    enable_auto_commit=True,  
    value_deserializer=lambda x: json.loads(x.decode('utf-8')),  
    consumer_timeout_ms=timeout_ms  
)
```

```
print(f"Consuming from topic: {topic_name}...")
```

```
data = []  
for message in consumer:  
    record = message.value  
    data.append(record)
```

```
if len(data) >= max_messages:  
    break
```

```
consumer.close()
```

```
if data:  
    df = pd.DataFrame(data)  
    print(f"Read {len(df)} records from {topic_name}")  
    return df  
else:
```

```
print(f"No data found in topic {topic_name}.")  
return pd.DataFrame() # Return empty DataFrame
```

**load\_data.py creates the transactions.jsonl and load\_to\_mongo.py is for loading data to mongoDB.**

## Last part: pipeline Optimization

Resource Monitoring

I. Implement Prometheus metrics for Kafka consumer lag, Spark executor CPU/MEM, and JVM garbage collection time.



**1. `prometheus.yml` file (put this in your project folder)**

```
yaml  
CopyEdit  
global:  
  scrape_interval: 5s # scrape every 5 seconds  
  
scrape_configs:  
  - job_name: 'spark'  
    static_configs:  
      - targets: ['localhost:4040'] # Spark UI  
  
      - job_name: 'kafka_consumer'  
        static_configs:
```

```
- targets: ['localhost:7071'] # Your custom Python consumer exporter
```

✓ This tells Prometheus to:

- Scrape Spark metrics from localhost:4040
- Scrape Kafka consumer lag metrics from localhost:7071

## 2. `kafka_consumer_monitor.py` (your Kafka Consumer Lag exporter)

```
python
CopyEdit
from kafka import KafkaConsumer
from prometheus_client import start_http_server, Gauge
import json
import time

# Create a Prometheus Gauge
consumer_lag_gauge = Gauge('kafka_consumer_lag', 'Kafka Consumer Lag fo
r derooghe.commission_by_type')

# Start Prometheus metrics server
start_http_server(7071) # Exposes metrics at http://localhost:7071

# Create Kafka consumer
consumer = KafkaConsumer(
    'darooghe.commission_by_type',
    bootstrap_servers='localhost:9092',
    auto_offset_reset='earliest',
    enable_auto_commit=True,
    group_id='monitoring-group', # Important: Set group_id to track committed
    offsets
    value_deserializer=lambda x: json.loads(x.decode('utf-8'))
```

```

)
print("Kafka consumer started. Exposing lag metrics...")

# Assign topic partitions manually
consumer.poll(timeout_ms=1000) # Poll to join group and get assignments

def get_consumer_lag(consumer):
    for tp in consumer.assignment():
        committed = consumer.committed(tp)
        end_offset = consumer.end_offsets([tp])[tp]
        if committed is not None and end_offset is not None:
            lag = end_offset - committed
            consumer_lag_gauge.set(lag)

# Infinite loop
try:
    while True:
        get_consumer_lag(consumer)
        time.sleep(5) # Update every 5 seconds
except KeyboardInterrupt:
    consumer.close()
    print("Consumer closed.")

```

 This script:

- Connects to your Kafka topic
- Measures **consumer lag** live
- Exposes it at <http://localhost:7071/metrics>
- Prometheus scrapes it every 5 seconds



## 3. Commands to Run Everything

Step	Command
Start Spark streaming application	(your <code>commission_analytics.py</code> )
Start Prometheus	<code>prometheus --config.file=prometheus.yml</code>
Start Kafka consumer monitor	<code>python kafka_consumer_monitor.py</code>

✓ Now visit:

- Prometheus UI: <http://localhost:9090>
- Targets: <http://localhost:9090/targets> (you should see Spark + Kafka consumer green ✓)

✓ You can **query metrics** like:

- `kafka_consumer_lag`
- `jvm_gc_collection_seconds_count` (if you configure JMX properly)

✓ You can **see CPU/MEM stats** from Spark UI itself ( <http://localhost:4040> ).

---



## Example folder structure:

```

CopyEdit
your_project/
|
└── commission_analytics.py
└── kafka_consumer_monitor.py
└── prometheus.yml
└── spark_checkpoint_commission_by_type/
└── spark_checkpoint_commission_ratio/
└── spark_checkpoint_top_merchants/
└── (other files)

```



## Summary

Monitor	How You Do It
Kafka Consumer Lag	<code>kafka_consumer_monitor.py</code> exposes via Prometheus
Spark Executor CPU/MEM	Scraped from <code>localhost:4040</code>
JVM Garbage Collection	From Spark metrics and Executors tab

As you can see, I run `commission_analytics.py` for this

```

! prometheus.yml
commission_analytics.py > ...
commission_analytics.py 2
.
.
.
# -----
# 6. Await termination
spark.streams.awaitAnyTermination(30)

```

```

:: retrieving :: org.apache.spark#spark-submit-parent-831750cc-9ca2-4f53-a088-789ea00af71d
conf: [default]
0 artifacts copied, 11 already retrieved (0kB/9ms)
25/04/28 23:22:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/04/28 23:22:46 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
25/04/28 23:22:46 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
25/04/28 23:22:46 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Datasets and will be disabled.
25/04/28 23:22:46 WARN AdminClientConfig: These configurations '[key.deserializer, value.deserializer, enable.auto.commit, max.poll.records, auto.offset.reset]' were supplied but are not used yet.
25/04/28 23:22:46 WARN AdminClientConfig: These configurations '[key.deserializer, value.deserializer, enable.auto.commit, max.poll.records, auto.offset.reset]' were supplied but are not used yet.
25/04/28 23:22:46 WARN AdminClientConfig: These configurations '[key.deserializer, value.deserializer, enable.auto.commit, max.poll.records, auto.offset.reset]' were supplied but are not used yet.
[Stage 1:>(115 + 8) / 200][Stage 3:> (0 + 0) / 200][Stage 6:> (0 + 0) / 100]

```

These are the outputs I got after running the yml file:

The screenshot shows a macOS desktop environment with a terminal window and a code editor.

**Terminal Window:**

```

(base) alireza@alirezas-MacBook-Air DS_C2_810101492_810101504_810101520 % prometheus --config.file=prometheus.yml
time=2025-04-28T19:55:31.439Z level=INFO source=main.go:640 msg="No time or size retention was set so using the default time retention" duration=15d
time=2025-04-28T19:55:31.439Z level=INFO source=main.go:687 msg="Starting Prometheus Server" mode=server version="(version=3.3.0, branch=no-n-git, revision=no-n-git)"
time=2025-04-28T19:55:31.439Z level=INFO source=main.go:764 msg="Leaving GOMAXPROCS=8: CPU quota undefined" component=autamaxprocs
time=2025-04-28T19:55:31.440Z level=INFO source=main.go:764 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2025-04-28T19:55:31.440Z level=INFO source=main.go:764 msg="Starting TSDB ..."
time=2025-04-28T19:55:31.443Z level=INFO source=tls_config.go:347 msg="Listening on" component=web address=[::]:9090
time=2025-04-28T19:55:31.443Z level=INFO source=tls_config.go:350 msg="TLS is disabled." component=web http2=false address=[::]:9090
time=2025-04-28T19:55:31.446Z level=INFO source=main.go:638 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2025-04-28T19:55:31.446Z level=INFO source=head.go:725 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=875ns
time=2025-04-28T19:55:31.446Z level=INFO source=head.go:733 msg="Replaying WAL, this may take a while" component=tsdb
time=2025-04-28T19:55:31.447Z level=INFO source=head.go:1228 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=0
time=2025-04-28T19:55:31.447Z level=INFO source=head.go:842 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=66.166μs wal_replay_duration=346.958μs wbl_replay_duration=83ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=875ns total_replay_duration=429.875μs
time=2025-04-28T19:55:31.448Z level=INFO source=main.go:1242 msg="filesystem information" fs_type=1a
time=2025-04-28T19:55:31.448Z level=INFO source=main.go:1252 msg="TSDB started"
time=2025-04-28T19:55:31.448Z level=INFO source=main.go:1437 msg="Loading configuration file" filename=prometheus.yml
time=2025-04-28T19:55:31.458Z level=INFO source=main.go:1476 msg="updated GOC" old=100 new=75
time=2025-04-28T19:55:31.458Z level=INFO source=main.go:1486 msg="Completed loading of configuration file" db_storage=625ns remote_storage=917ns web_handler=417ms query_engine=625ns scrape=20.008749583s scrape_sd=824.917μs notify=1.917μs rules=56.417μs tracing=100.125μs filename=prometheus.yml totalDuration=20.010315459s
time=2025-04-28T19:55:51.458Z level=INFO source=main.go:1213 msg="Server is ready to receive web requests."
time=2025-04-28T19:55:51.458Z level=INFO source=manager.go:175 msg="Starting rule manager..." component=rule manager

```

**Code Editor:**

- OPEN EDITORS:
  - ! prometheus.yml
  - ! kafka\_consumer\_monitor.py 2
  - commission\_analytics.py 9+
- PROBLEMS: 16
- OUTPUT
- DEBUG CONSOLE
- TERMINAL
- PORTS

The screenshot shows a macOS desktop environment with a terminal window and a code editor.

**Terminal Window:**

```

0240, hard-unlimited" vm_limits="(soft=unlimited, hard=unlimited)"
time=2025-04-28T19:58:34.952Z level=INFO source=main.go:768 msg="Leaving GOMAXPROCS=8: CPU quota undefined" component=autamaxprocs
time=2025-04-28T19:58:34.960Z level=INFO source=web.go:654 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2025-04-28T19:58:34.964Z level=INFO source=main.go:1228 msg="Starting TSDB ..."
time=2025-04-28T19:58:34.967Z level=INFO source=tls_config.go:347 msg="Listening on" component=web address=[::]:9090
time=2025-04-28T19:58:34.967Z level=INFO source=tls_config.go:350 msg="TLS is disabled." component=web http2=false address=[::]:9090
time=2025-04-28T19:58:34.971Z level=INFO source=head.go:638 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2025-04-28T19:58:34.972Z level=INFO source=head.go:725 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=92ns
time=2025-04-28T19:58:34.975Z level=INFO source=head.go:733 msg="Replaying WAL, this may take a while" component=tsdb
time=2025-04-28T19:58:34.981Z level=INFO source=head.go:805 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=1
time=2025-04-28T19:58:34.981Z level=INFO source=head.go:805 msg="WAL segment loaded" component=tsdb segment=1 maxSegment=1
time=2025-04-28T19:58:34.981Z level=INFO source=head.go:805 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=772.167μs wal_replay_duration=5.616167ms wbl_replay_duration=84ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=792ns total_replay_duration=6.40725ms
time=2025-04-28T19:58:34.983Z level=INFO source=main.go:1249 msg="filesystem information" fs_type=1a
time=2025-04-28T19:58:34.984Z level=INFO source=main.go:1252 msg="TSDB started"
time=2025-04-28T19:58:34.984Z level=INFO source=main.go:1437 msg="Loading configuration file" filename=prometheus.yml
time=2025-04-28T19:58:55.002Z level=INFO source=main.go:1476 msg="updated GOC" old=100 new=75
time=2025-04-28T19:58:55.002Z level=INFO source=main.go:1486 msg="Completed loading of configuration file" db_storage=917ns remote_storage=1.209ns web_handler=1.458ms query_engine=834ms scrape=20.008755833s scrape_sd=1.272625ms notify=22.834μs notify_sd=1.208μs rules=637.667μs tracing=1.980333ms filename=prometheus.yml totalDuration=20.017103291s
time=2025-04-28T19:58:55.002Z level=INFO source=main.go:1213 msg="Server is ready to receive web requests."
time=2025-04-28T19:58:55.002Z level=INFO source=manager.go:175 msg="Starting rule manager..." component=rule manager
time=2025-04-28T19:59:04.310Z level=ERROR source=scrape_manager.go:1089 msg="Failed to determine correct type of scrape target." component=scrape_manager scrape_pool=spark targets=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type 'text/html;charset=utf-8' and no fallback_scrape_protocol specified for target"
time=2025-04-28T19:59:08.783Z level=ERROR source=scrape_manager.go:1600 msg="Failed to determine correct type of scrape target." component=scrape_manager scrape_pool=spark targets=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type 'text/html;charset=utf-8' and no fallback_scrape_protocol specified for target"
time=2025-04-28T19:59:13.775Z level=ERROR source=scrape.go:1608 msg="Failed to determine correct type of scrape target." component=scrape_manager scrape_pool=spark targets=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type 'text/html;charset=utf-8' and no fallback_scrape_protocol specified for target"

```

**Code Editor:**

- OPEN EDITORS:
  - ! prometheus.yml
  - ! kafka\_consumer\_monitor.py 2
  - commission\_analytics.py 9+
- PROBLEMS: 16
- OUTPUT
- DEBUG CONSOLE
- TERMINAL
- PORTS

The screenshot shows a macOS desktop environment. In the foreground, a terminal window is open with the following command:

```
! prometheus.yml
# kafka_consumer_monitor.py 2
commission_analytics.py 9+
```

The log output window displays the following log entries:

```
time=2025-04-28T19:58:34.083Z level=INFO source=main.go:1249 msg="filesystem information" fs_type=1a
time=2025-04-28T19:58:34.084Z level=INFO source=main.go:1252 msg="TSDB started"
time=2025-04-28T19:58:34.084Z level=INFO source=main.go:1437 msg="Loading configuration file" filename=prometheus.yml
time=2025-04-28T19:58:35.000Z level=INFO source=main.go:1476 msg="updated GOCV old=100 new=75"
time=2025-04-28T19:58:35.000Z level=INFO source=main.go:1486 msg="Completed loading of configuration file" db_storage=917ns remote_storage=1.290us web_handler=1.458us query_engine=834ns scrape=20.00875583s scrape_sd=1.727625ms notify=22.834us notify_sd=1.208us rules=637.667us tracing=1.980333ms
time=2025-04-28T19:58:35.000Z level=INFO source=main.go:1213 msg="Server is ready to receive web requests."
time=2025-04-28T19:58:35.000Z level=INFO source=manager.go:175 msg="Starting rule manager..." component="rule manager"
time=2025-04-28T19:58:35.000Z level=INFO source=manager.go:175 msg="Starting rule manager..." component="scrape manager"
time=2025-04-28T19:58:35.000Z level=INFO source=scrape.go:1609 msg="Failed to determine correct type of scrape target." component="scrape manager" scrape_pool=spart target=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type \"text/html;charset=utf-8\" and no fallback_scrape_protocol specified for target"
time=2025-04-28T19:59:08.783Z level=ERROR source=scrape.go:1609 msg="Failed to determine correct type of scrape target." component="scrape manager" scrape_pool=spart target=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type \"text/html;charset=utf-8\" and no fallback_scrape_protocol specified for target"
time=2025-04-28T19:59:13.775Z level=ERROR source=scrape.go:1609 msg="Failed to determine correct type of scrape target." component="scrape manager" scrape_pool=spart target=http://localhost:4040/metrics content_type="text/html;charset=utf-8" fallback_media_type="" err="received unsupported Content-Type \"text/html;charset=utf-8\" and no fallback_scrape_protocol specified for target"
^Ctime=2025-04-28T20:00:31.358Z level=WARN source=main.go:1015 msg="Received an OS signal, exiting gracefully..." signal=interrupt
time=2025-04-28T20:00:31.362Z level=INFO source=main.go:1040 msg="Stopping scrape discovery manager..."
time=2025-04-28T20:00:31.362Z level=INFO source=main.go:1054 msg="Stopping notify discovery manager..."
time=2025-04-28T20:00:31.362Z level=INFO source=main.go:1036 msg="Scrape discovery manager stopped"
time=2025-04-28T20:00:31.362Z level=INFO source=main.go:1050 msg="Notify discovery manager stopped"
time=2025-04-28T20:00:31.364Z level=INFO source=manager.go:189 msg="Stopping rule manager..." component="rule manager"
time=2025-04-28T20:00:31.365Z level=INFO source=manager.go:205 msg="Rule manager stopped" component="rule manager"
time=2025-04-28T20:00:31.366Z level=INFO source=main.go:1091 msg="Stopping scrape manager..."
time=2025-04-28T20:00:31.369Z level=INFO source=main.go:1083 msg="Scrape manager stopped"
time=2025-04-28T20:00:31.409Z level=INFO source=notifier.go:16 msg="Stopping notification manager..." component=notifier
time=2025-04-28T20:00:31.413Z level=INFO source=notifier.go:408 msg="Draining any remaining notifications..." component=notifier
time=2025-04-28T20:00:31.413Z level=INFO source=notifier.go:414 msg="Remaining notifications drained" component=notifier
time=2025-04-28T20:00:31.415Z level=INFO source=notifier.go:344 msg="Notification manager stopped" component=notifier
time=2025-04-28T20:00:31.415Z level=INFO source=main.go:1361 msg="Notifier manager stopped"
time=2025-04-28T20:00:31.415Z level=INFO source=main.go:1375 msg="See you next time!"
```

And this is output in <http://localhost:9090/>

The screenshot shows the Prometheus UI interface. At the top, there are two tabs: "Table" (selected) and "Graph". Below the tabs, there is a search bar containing the query `_ kafka_consumer_lag`. The results show one result series:

```
kafka_consumer_lag{instance="localhost:7071", job="kafka_consumer"} 0
```

Below this, another tab "Table" is selected, and the search bar contains the query `_ jvm_gc_collection_seconds_count`. The results show one result series:

```
jvm_gc_collection_seconds_count{job="jvm_gc_collection_seconds_count"} 0
```

At the bottom left, there is a button labeled "+ Add query".