



**BERLIN SCHOOL OF
BUSINESS & INNOVATION**

Essay / Assignment Title: Cloud-Based Big Data Analytics with Apache Spark and Hadoop Ecosystem

Programme title: MSc. DATA ANALYTICS

Name: SEFA ÜNVEREN

Year: 2025

CONTENTS

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters):

.....SEFA ÜNVEREN.....

Date:13..../...02.../...2025..

INTRODUCTION

In today's digital world, Big data has become a strategic asset for business, human life, and governments. With increasing digital usage, the amount of data generated has become too big and complex and impossible to process with traditional data processing methods. Therefore, big data analytics requires advanced technologies and methodologies to collect, store, process, and analyze data.

This assignment is on the practical implementation of Big Data technologies and the derivation of meaningful insights from large-scale data. As a dataset, I have chosen the Beauty and Personal Care review dataset from Amazon Reviews 2023. This dataset meets the requirements such as a public dataset and the size of the dataset is bigger than 10 GB.

This size of dataset was very hard to process and needed a long time to get meaningful insights.

CHAPTER ONE

TASK 1

In recent years, the world's understanding of trade and shopping has undergone a radical change. Shopping that used to be done face-to-face in stores is now largely done through e-commerce websites. And every day, more and more people shop on e-commerce websites. As a result, millions of products and people's information increase the dataset size to an incredible extent.

Traditional data analysis methods are insufficient for such large datasets. For this reason, different programs and methods are needed to analyze big data. By processing big datasets and extracting meaningful insights, we can shape the future of our e-commerce website, increase customer numbers, product variety, and quality, and reach higher business volume.

Based on big data analysis, e-commerce platforms can improve themselves and help their customer's decision-making process. For example, by analyzing customer shopping behavior, platforms can recommend relevant products.

By analyzing product reviews and ratings, platforms can gain meaningful insights and improve customer satisfaction. In addition to big data analysis, real-time analysis and using machine learning models provide great benefits to e-commerce platforms. Platforms like Netflix or Amazon use ML models for recommendation systems.

Some platforms that are focusing on better shopping experience use clustering techniques. With this technique, customers are divided into different clusters such as loyal customers, seasonal customers, or seeking discount customers.

For this project, I will use the beauty and personal care dataset from Amazon Reviews 2023 datasets. The dataset size is 10.3 GB. I chose this dataset on purpose because generally, women are shopping in this category and I expect that this dataset may include more information about customer interactions and product reviews.

CHAPTER TWO

TASK 2

I prefer to use Google Cloud because in our theoretical and practical lessons, we used it however Google Cloud has a smaller number of data centers compared to AWS and Azure, which are known for their high-performance private network infrastructure. Google Cloud Platform leads in data management and analytics, offering tools like BigQuery or DataProc that make it a good choice for enterprises scaling real-time big data processing, machine learning workflows, and data-driven decision-making.

To use Google Cloud we have to register on the website <https://console.cloud.google.com/> After registering with our personal information we have to register our credit card by clicking billing on the left side of the screen. After registering successfully, our first job must be adding APIs. In the search bar which is located on top of the screen, if we write DataProc and search, our screen will be as same as in Figure 1.

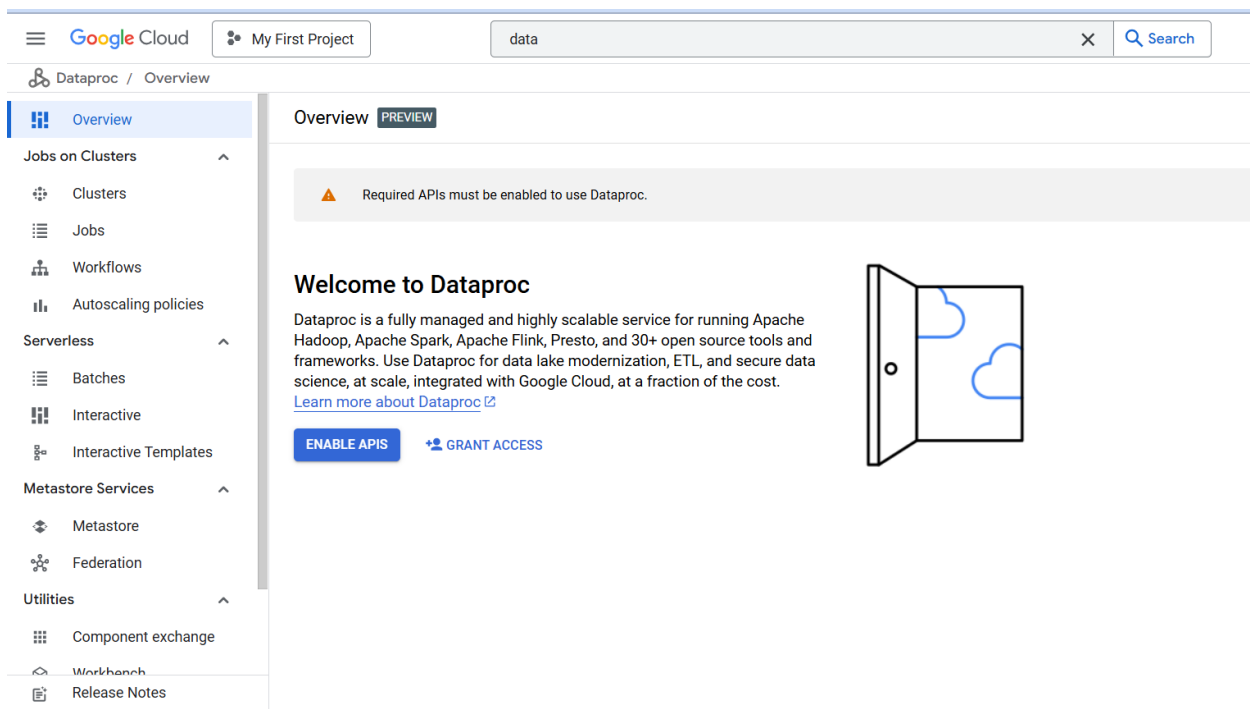


Figure 1 – Searching DataProc

DataProc is crucial for us because as seen on the screen Apache Hadoop and Apache Spark are accessible. After clicking on enable APIs, DataProc will be successfully installed. The next thing after the installation of DataProc, we have to create a cluster.

When you click on 'create cluster', the options will occur as shown in Figure 2.

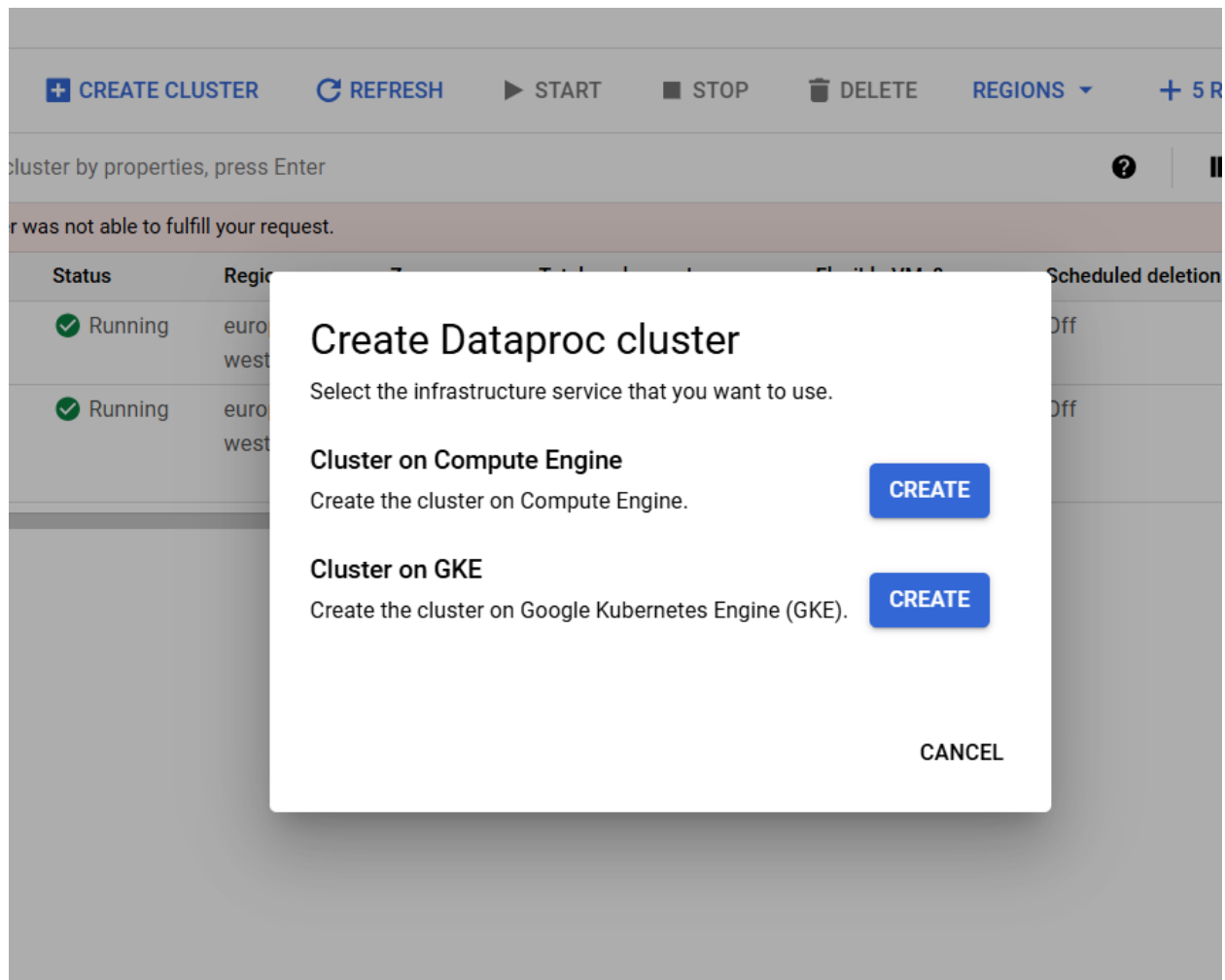


Figure 2 – Creating Cluster

By clicking 'create cluster on compute engine', we are redirected to another web page which is shown in Figure 3.

← Create a Dataproc cluster on Compute Engine

- **Set up cluster**
Begin by providing basic information.
- **Configure nodes (optional)**
Change node compute and storage capabilities.
- **Customize cluster (optional)**
Add cluster properties, features, and actions.
- **Manage security (optional)**
Change access, encryption, and security settings.

CREATE **CANCEL**

EQUIVALENT COMMAND LINE ▾

Name

Cluster Name *
cluster-7446 ?

Location

Region *
europe-west8 ▼ ?

Zone *
europe-west8-a ▼ ?

Cluster type

☒ **Standard (1 master, N workers)**

☐ **Single Node (1 master, 0 workers)**
Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing

☐ **High Availability (3 masters, N workers)**
Hadoop High Availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Versioning

Use a custom image to load pre-installed packages. [Learn more](#)

Image Type and Version
2.1-ubuntu20

Release Date

Figure 3 – Creating Cluster

As shown in Figure 3, the cluster name is given automatically, but I have to choose a location or region near our location or region. In the middle part, we must specify the cluster type according to our jobs. If we have a large-scale process, we can choose standard or high availability. The single node option is for small-scale processing.

The versioning differences apply to Linux-based operating systems like Ubuntu, Debian, or RockyLinux. There are small differences between these systems for example Ubuntu makes frequent updates on the system but Debian has a more conservative mentality and stability is its first priority. That's why Ubuntu offers 5 years of support for their product but Debian supports 3 years for their stable products.

In the versioning part, we must choose the proper system for our work. I had chosen the 2.1 Ubuntu 20.04 version as shown in figure 4.

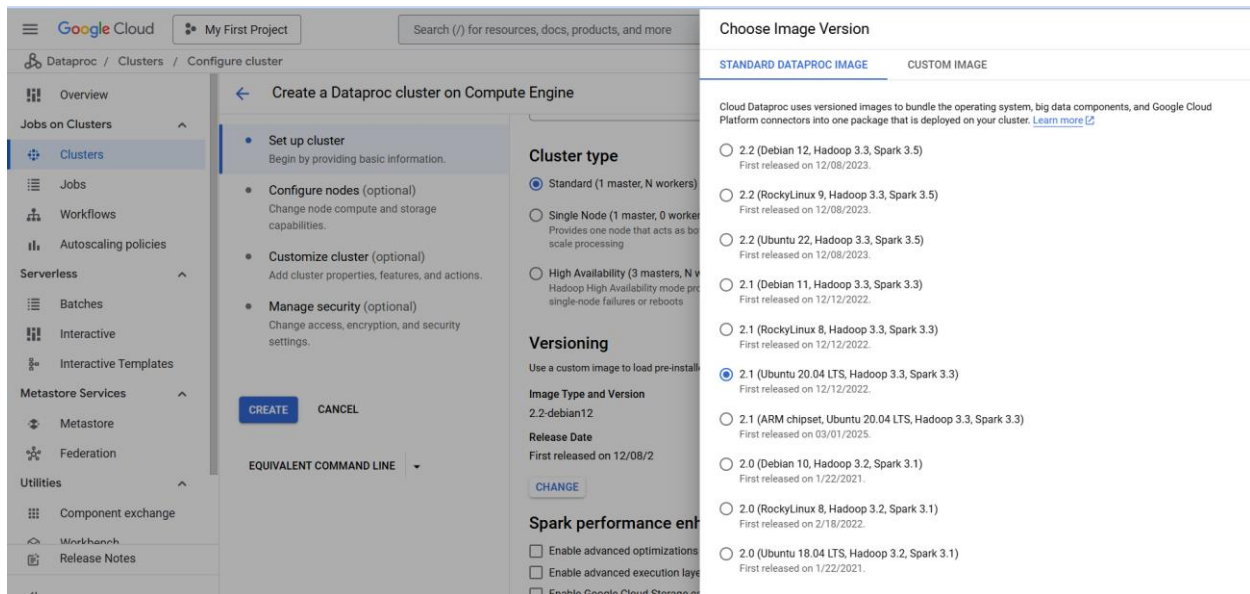


Figure 4 – Creating Cluster

Now we have to make other settings in configure nodes menu.

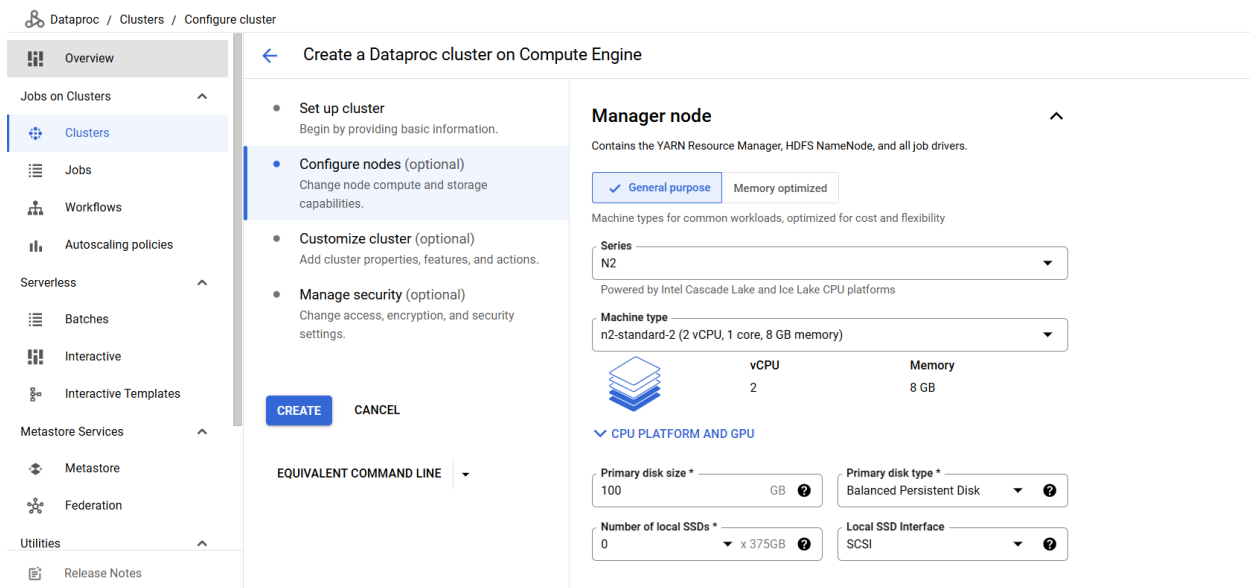


Figure 5 – Cluster Settings

The point we need to pay attention to in this setting is that manager node settings and worker node settings must be the same.

Figure 6 – Cluster settings

When all settings are done, we can click the create option to create a cluster. Unfortunately due to a technical problem, I received an error as shown below in Figure 7.

Error

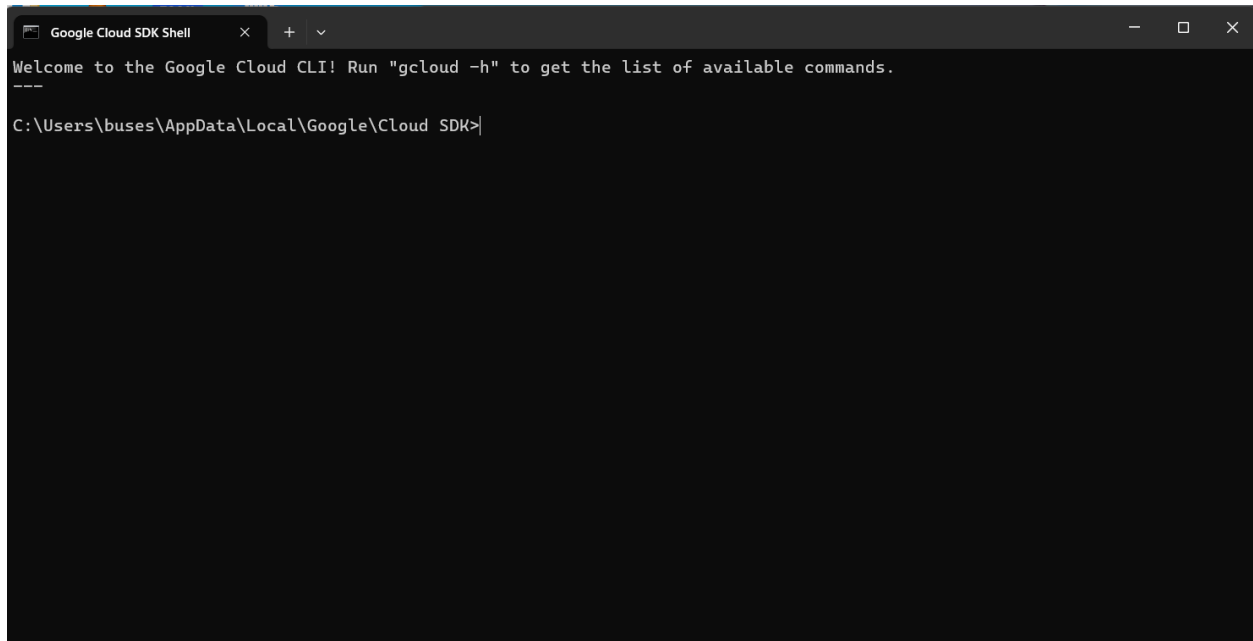
Multiple validation errors: - Insufficient 'CPUS_ALL_REGIONS' quota. Requested 6.0, available 0.0. Your resource request exceeds your available quota. See <https://cloud.google.com/compute/resource-usage>. Use https://cloud.google.com/docs/quotas/view-manage#requesting_higher_quota to request additional quota. - Insufficient 'SSD_TOTAL_GB' quota. Requested 300.0, available 250.0. Your resource request exceeds your available quota. See <https://cloud.google.com/compute/resource-usage>. Use https://cloud.google.com/docs/quotas/view-manage#requesting_higher_quota to request additional quota. - This request exceeds CPU quota. Some things to try: request fewer workers (a minimum of 2 is required), use smaller master and/or worker machine types (such as n1-standard-2).

Request ID: 17659404521831402219

SEND FEEDBACK
CLOSE

Figure 7 – Error message while creating the cluster

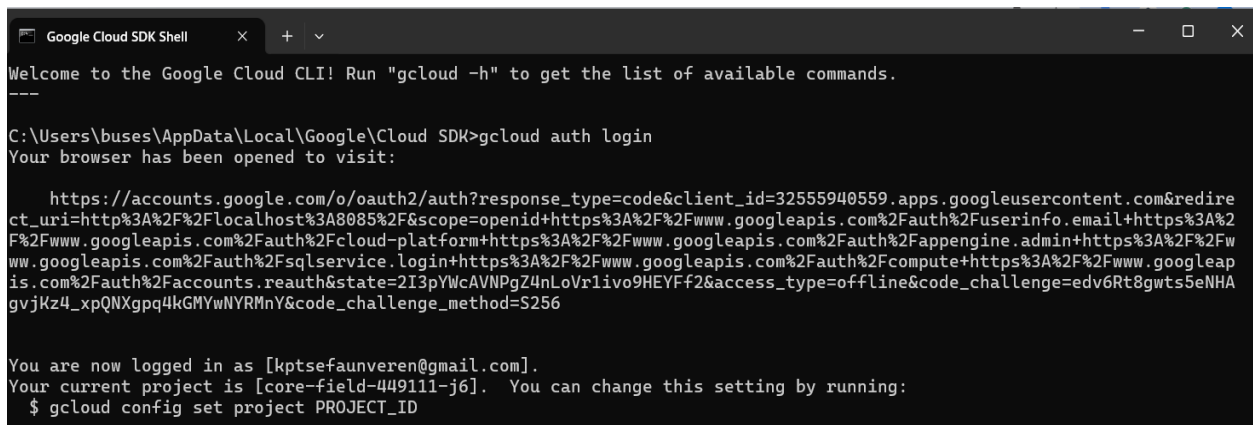
Due to receiving errors many times, I want to try with Google Cloud Shell by typing cluster codes. After downloading and installing SDK on my computer, the opening screen is shown in Figure 8.

A screenshot of a terminal window titled "Google Cloud SDK Shell". The window has a dark background with white text. The text inside the terminal reads: "Welcome to the Google Cloud CLI! Run 'gcloud -h' to get the list of available commands." followed by a separator line "----". Below this, the command prompt shows the current directory: "C:\Users\buses\AppData\Local\Google\Cloud SDK>".

```
Google Cloud SDK Shell
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
----
C:\Users\buses\AppData\Local\Google\Cloud SDK>
```

Figure 8 – Google Cloud SDK

Firstly I wrote 'gcloud auth login' to log in with my Google account. Then my internet browser opened and showed my Google account.

A screenshot of a terminal window titled "Google Cloud SDK Shell". The window has a dark background with white text. The text inside the terminal reads: "Welcome to the Google Cloud CLI! Run 'gcloud -h' to get the list of available commands." followed by a separator line "----". Below this, the command prompt shows the user has entered 'gcloud auth login'. The terminal then displays a long URL for the user to visit in their browser. After the URL, it says "You are now logged in as [kptsefaunveren@gmail.com]. Your current project is [core-field-449111-j6]. You can change this setting by running: \$ gcloud config set project PROJECT_ID".

```
Google Cloud SDK Shell
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
----
C:\Users\buses\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=2I3pYWcAVNPgZ4nLoVr1ivo9HEYFf2&access_type=offline&code_challenge=edv6Rt8gwt5eNHAgvjKz4_xpQNXgpp4kGMYYwNYRMnY&code_challenge_method=S256

You are now logged in as [kptsefaunveren@gmail.com].
Your current project is [core-field-449111-j6]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
```

Figure 9 - Google Cloud SDK

After successfully logging in, I typed my cluster settings by reducing disk size.

The code is: `gcloud dataproc clusters create my-cluster --region=europe-west6 --zone=europe-west6-a --master-machine-type=n1-standard-2 --master-boot-disk-size=20GB --worker-machine-type=n1-standard-2 --num-workers=2 --worker-boot-disk-size=200GB --image-version=2.1-ubuntu20`

But again, I received an error as shown in Figure 10.

```
C:\Users\buses\AppData\Local\Google\Cloud SDK>gcloud dataproc clusters create my-cluster --region=europe-west6 --zone=europe-west6-a --master-machine-type=n1-standard-2 --master-boot-disk-size=20GB --worker-machine-type=n1-standard-2 --num-workers=2 --worker-boot-disk-size=20GB --image-version=2.1-ubuntu20
ERROR: (gcloud.dataproc.clusters.create) INVALID_ARGUMENT: Multiple validation errors:
- Insufficient 'CPUS_ALL_REGIONS' quota. Requested 6.0, available 0.0. Your resource request exceeds your available quota. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quota/view-manage#requesting_higher_quota to request additional quota.
- Insufficient 'IN_USE_ADDRESSES' quota. Requested 3.0, available 1.0. Your resource request exceeds your available quota. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quota/view-manage#requesting_higher_quota to request additional quota.
- Requested image requires minimum boot disk size of 30 GB; requested 20 GB
- This request exceeds CPU quota. Some things to try: request fewer workers (a minimum of 2 is required), use smaller master and/or worker machine types (such as n1-standard-2).
```

Figure 10 - Google Cloud SDK Error

Then I decided to change the region and disk size which mentioned in the error minimum disk size must be 30GB. I had to change the region because I reached the quota limit while trying to create a cluster.

The code is: `gcloud dataproc clusters create my-cluster --region=europe-west4 --zone=europe-west4-a --master-machine-type=n1-standard-2 --master-boot-disk-size=30GB --worker-machine-type=n1-standard-2 --num-workers=2 --worker-boot-disk-size=30GB --image-version=2.1-ubuntu2`

Then after this code, as you can see in Figure 11 my cluster is created.

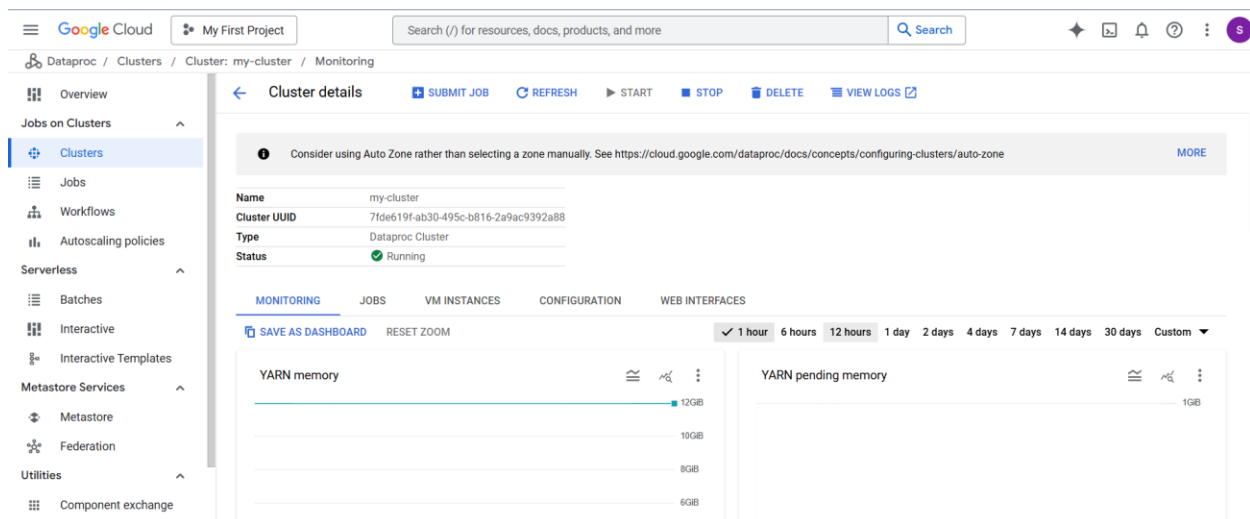


Figure 11 – My Cluster

The next step is creating a bucket. A bucket is a logical container in cloud storage systems, used to organize and store datasets.

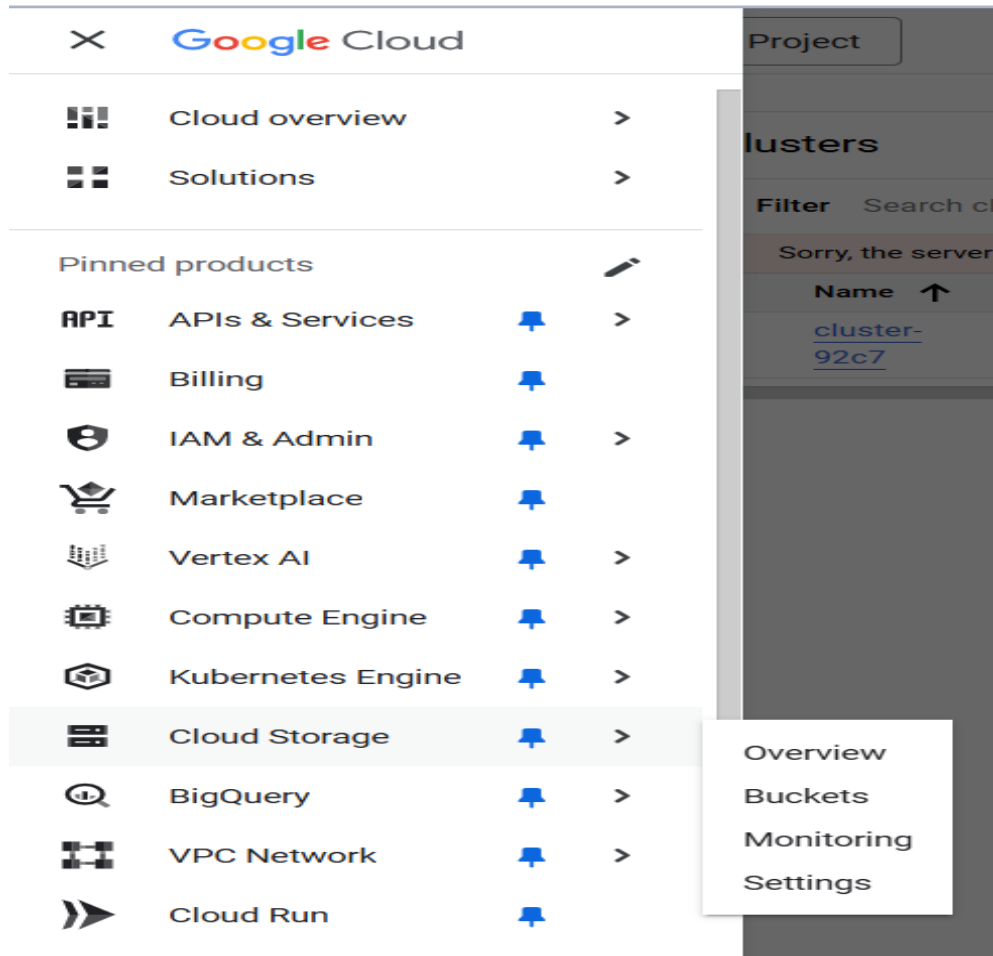


Figure 12 – Create Buckets

On the left menu, choose 'Cloud Storage' and click on the buckets menu as shown in Figure 12. On the new shown in figure 13.

Google Cloud My First Project Search (/) for resources, docs, products, and more Search

Cloud Storage Create a bucket

Overview Buckets Monitoring Settings

Marketplace Release Notes

Get Started
Pick a globally unique, permanent name. [Naming guidelines](#)
Ex. 'example', 'example_bucket-1', or 'example.com'
Tip: Don't include any sensitive information

Optimize storage for data-intensive workloads

Labels (optional)

CONTINUE

Choose where to store your data
Location: us (multiple regions in United States)
Location type: Multi-region

Choose a storage class for your data
Default storage class: Standard

Choose how to control access to objects
Public access prevention: On
Access control: Uniform

Good to know
Location pricing
Storage rates vary depending on the storage class of your data and location of your bucket. [Pricing details](#)
Current configuration: Multi-region / Standard

Item	Cost
us (multiple regions in United States)	\$0.026 per GB-month
With default replication	\$0.020 per GB written

ESTIMATE YOUR MONTHLY COST

Figure 13 – Creating Bucket

In this screen, we must make settings of buckets such as giving the name and selecting the region where to store our data. When we start typing the name, the system scans it and warns us if the name is taken. As a region, we can select one of the multi-region, dual-region, and single-region options. Only pricing is changing between options. I choose multiple regions in the European Union and after clicking on the create button, in a few seconds, our buckets will be created.

Google Cloud My First Project Search (/) for resources, docs, products, and more Search

Cloud Storage Create a bucket

Overview Buckets Monitoring Settings

Marketplace Release Notes

Get Started
Pick a globally unique, permanent name. [Naming guidelines](#)
assignment_86
Tip: Don't include any sensitive information

Optimize storage for data-intensive workloads

Labels (optional)

CONTINUE

Choose where to store your data
This choice defines the geographic placement of your data and affects cost, performance, and availability. Cannot be changed later. [Learn more](#)
Location type
☒ Multi-region
Highest availability across largest area
eu (multiple regions in European Union)

☐ Add cross-bucket replication via Storage Transfer Service
As data is added or changed, replicate it to another bucket, enabling you to store

Good to know
Location pricing
Storage rates vary depending on the storage class of your data and location of your bucket. [Pricing details](#)
Current configuration: Multi-region / Standard

Item	Cost
eu (multiple regions in European Union)	\$0.026 per GB-month
With default replication	\$0.020 per GB written

ESTIMATE YOUR MONTHLY COST

Figure 14 – Creating Bucket

As a result, our bucket will shown on the bucket list screen as shown in Figure 15.

Name	Created	Location type	Location	Default storage class	Last modified	Public access
assignment86	Feb 7, 2025, 11:28:31 PM	Multi-region	eu	Standard	Feb 7, 2025, 11:28:31 PM	Not public
dataprocc-staging-europe-west4-92621608	Feb 12, 2025, 6:51:06 PM	Region	europe-west4	Standard	Feb 12, 2025, 6:51:06 PM	Subject to object ACL
dataprocc-staging-europe-west6-92621608	Feb 7, 2025, 9:40:29 PM	Region	europe-west6	Standard	Feb 7, 2025, 9:40:29 PM	Subject to object ACL
dataprocc-temp-europe-west4-92621608	Feb 12, 2025, 6:51:06 PM	Region	europe-west4	Standard	Feb 12, 2025, 6:51:06 PM	Subject to object ACL
dataprocc-temp-europe-west6-92621608	Feb 7, 2025, 9:40:29 PM	Region	europe-west6	Standard	Feb 7, 2025, 9:40:29 PM	Subject to object ACL

Figure 15 – Bucket List

Now we can upload our dataset in these buckets.

Bucket details for **assignment86**

Location: eu (multiple regions in European Union) | Storage class: Standard | Public access: Not public | Protection: Soft Delete

OBJECTS | CONFIGURATION | PERMISSIONS | PROTECTION | LIFECYCLE | OBSERVABILITY | INVENTORY REPORTS | OPERATIONS

Folder browser: **assignment86**

CREATE FOLDER | **UPLOAD** | TRANSFER DATA | OTHER SERVICES

Filter by name prefix only | Upload files | Filter objects and folders | Show Live objects only

Name	Size	Type	Created	Storage class
------	------	------	---------	---------------

Figure 16 – Upload Files

When we click on the bucket name where we want to upload the dataset, The screen will be as same as Figure 16. By clicking on the upload files, the new Windows screen will appear, and by selecting the files which we want to upload.

File Explorer: Desktop > SEFA > BIG DATA ANALYTICS

Files: AmazonReviews-2023-main, galiba, Gigantic (-14 GB)

Google Cloud Console: assignment86 bucket details

UPLOAD | TRANSFER DATA | OTHER SERVICES

Filter by name prefix only | Filter objects and folders | Show Live objects only

Name	Size	Type	Created	Storage class
Amazon-Reviews-2023.py	38.7 KB	application/octet-stream	Feb 7, 2025, 11:42:55 PM	Standard
Beauty_and_Personal_Care.jsonl.gz	10.3 GB	application/x-gzip	Feb 12, 2025, 8:41:44 PM	Standard
Home_and_Kitchen.jsonl.gz	29.3 GB	application/x-gzip	Feb 12, 2025, 10:47:35 PM	Standard
all_categories.txt	585 B	text/plain	Feb 13, 2025, 6:00:45 PM	Standard
asin2category.json	1.2 GB	application/json	Feb 7, 2025, 11:50:11 PM	Standard

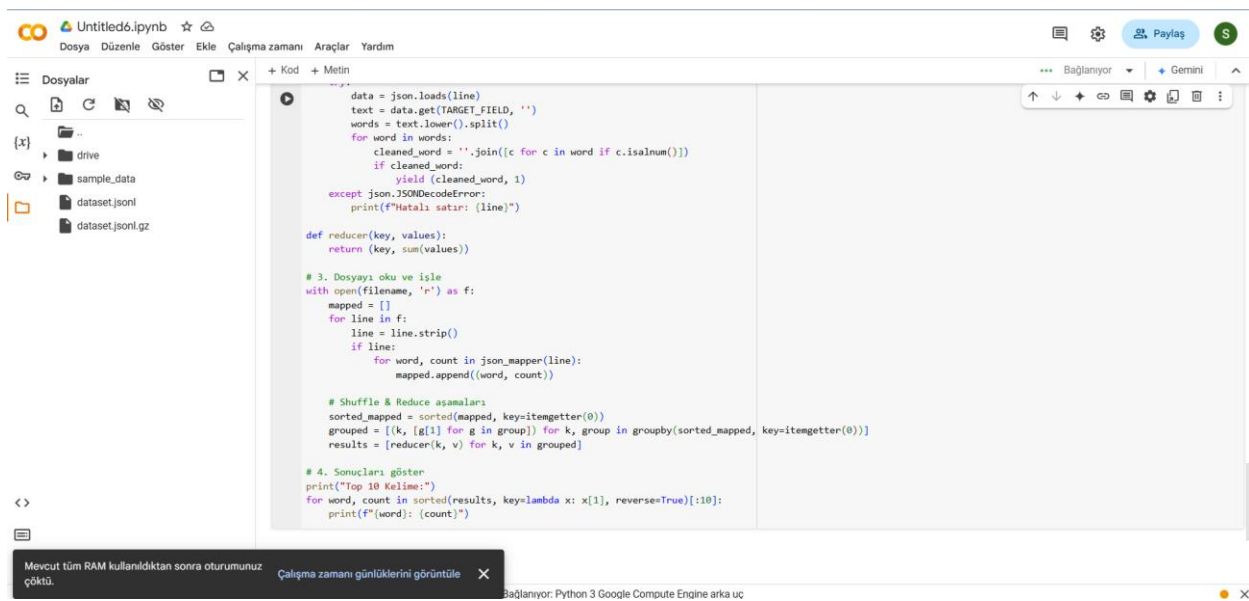
Figure 17 – Upload Files

CHAPTER THREE

TASK 3

Big data analytics requires powerful technologies to efficiently process datasets. In this context, Hadoop MapReduce and Apache Spark play a crucial role in handling and analyzing big data.

Firstly we will use Hadoop MapReduce to perform basic data processing and cleaning tasks such as counting word frequencies.



```
data = json.loads(line)
text = data.get(TARGET_FIELD, '')
words = text.lower().split()
for word in words:
    cleaned_word = ''.join([c for c in word if c.isalnum()])
    if cleaned_word:
        yield (cleaned_word, 1)
except json.JSONDecodeError:
    print(f"Hatalı satır: {line}")

def reducer(key, values):
    return (key, sum(values))

# 3. Dosyayı oku ve işle
with open(filename, 'r') as f:
    mapped = []
    for line in f:
        line = line.strip()
        if line:
            for word, count in json_mapper(line):
                mapped.append((word, count))

# Shuffle & Reduce aşamaları
sorted_mapped = sorted(mapped, key=itemgetter(0))
grouped = [(k, [g[1] for g in group]) for k, group in groupby(sorted_mapped, key=itemgetter(0))]
results = [reducer(k, v) for k, v in grouped]

# 4. Sonuçları göster
print("Top 10 Kelime:")
for word, count in sorted(results, key=lambda x: x[1], reverse=True)[:10]:
    print(f"{word}: {count}")
```

Figure 18 – MapReduce Error

When I typed MapReduce codes for the counting words, I got an error which means 'Because of using all RAM, your session is collapsed'. Due to this error I changed my codes and try to make more simple than before and as a result I succeeded to analyze.



```
import json
from collections import defaultdict
from google.colab import drive

FILE_PATH = '/content/dataset.jsonl'
TARGET_FIELD = 'text'
TOP_N = 10

def process_line(line, counter):
    """Satırı işleyip counter'ı günceller"""
    try:
        data = json.loads(line)
        text = data.get(TARGET_FIELD, '')
        words = text.lower().split()
        for word in words:
            cleaned_word = ''.join([c for c in word if c.isalnum()])
            if cleaned_word:
                counter[cleaned_word] += 1
    except Exception as e:
        pass

def main():
    word_counts = defaultdict(int)

    with open(FILE_PATH, 'r') as f:
        for i, line in enumerate(f):
            process_line(line.strip(), word_counts)

    if i % 100000 == 0:
        print(f"İşlenen satır: {i+1} | Bellek kullanımı: {len(word_counts)} kelime")
```

Figure 19 – Python Codes for MapReduce



```
if i % 100000 == 0:
    print(f"İşlenen satır: {i+1} | Bellek kullanımı: {len(word_counts)} kelime")

sorted_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse=True)
print(f"Top {TOP_N} Kelime:")
for word, count in sorted_counts[:TOP_N]:
    print(f"{word}: {count}")

if __name__ == "__main__":
    main()
```

İşlenen satır: 1 | Bellek kullanımı: 128 kelime
İşlenen satır: 100001 | Bellek kullanımı: 67522 kelime
İşlenen satır: 200001 | Bellek kullanımı: 95404 kelime
İşlenen satır: 300001 | Bellek kullanımı: 116168 kelime
İşlenen satır: 400001 | Bellek kullanımı: 135539 kelime
İşlenen satır: 500001 | Bellek kullanımı: 152275 kelime
İşlenen satır: 600001 | Bellek kullanımı: 168872 kelime
İşlenen satır: 700001 | Bellek kullanımı: 182849 kelime
İşlenen satır: 800001 | Bellek kullanımı: 197963 kelime
İşlenen satır: 900001 | Bellek kullanımı: 212176 kelime
İşlenen satır: 1000001 | Bellek kullanımı: 225824 kelime
İşlenen satır: 1100001 | Bellek kullanımı: 237767 kelime
İşlenen satır: 1200001 | Bellek kullanımı: 249159 kelime
İşlenen satır: 1300001 | Bellek kullanımı: 261754 kelime
İşlenen satır: 1400001 | Bellek kullanımı: 272705 kelime
İşlenen satır: 1500001 | Bellek kullanımı: 282593 kelime
İşlenen satır: 1600001 | Bellek kullanımı: 292729 kelime
İşlenen satır: 1700001 | Bellek kullanımı: 304823 kelime
İşlenen satır: 1800001 | Bellek kullanımı: 314267 kelime
İşlenen satır: 1900001 | Bellek kullanımı: 326120 kelime
İşlenen satır: 2000001 | Bellek kullanımı: 336295 kelime
İşlenen satır: 2100001 | Bellek kullanımı: 346184 kelime
İşlenen satır: 2200001 | Bellek kullanımı: 356128 kelime
İşlenen satır: 2300001 | Bellek kullanımı: 366617 kelime
İşlenen satır: 2400001 | Bellek kullanımı: 376787 kelime
İşlenen satır: 2500001 | Bellek kullanımı: 385673 kelime
İşlenen satır: 2600001 | Bellek kullanımı: 395378 kelime

Figure 20 - Python Codes for MapReduce

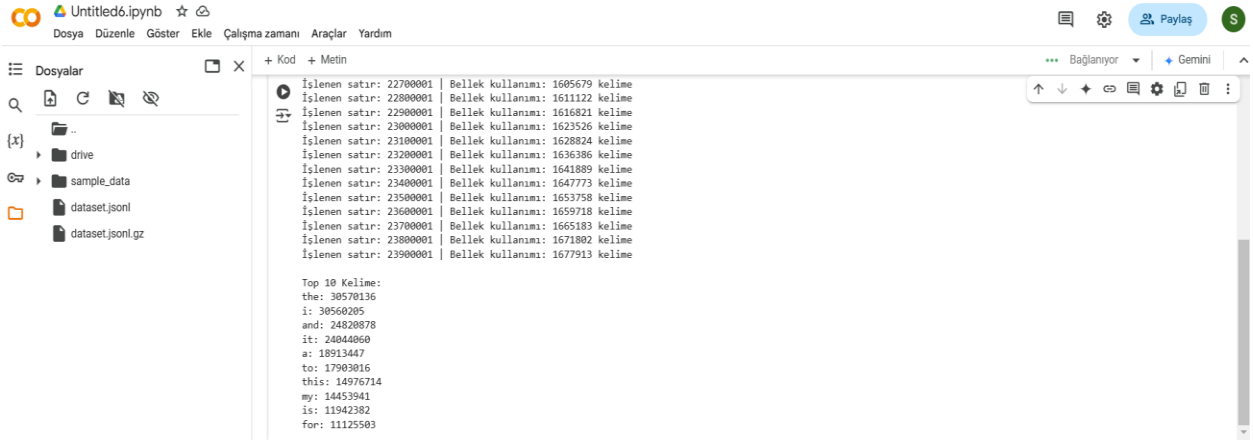


Figure 21 - Python Codes for MapReduce

As a result of analyze top 10 words as shown in Figure 21. The most used word is ‘the’ and it is used 30 570 136 times.

Then we will use Apache Spark to be utilized to conduct advanced data transformations and analyses, including data filtering, aggregation, and exploratory data analysis (EDA). These processes will help make large datasets more meaningful, ultimately supporting data-driven business decisions.

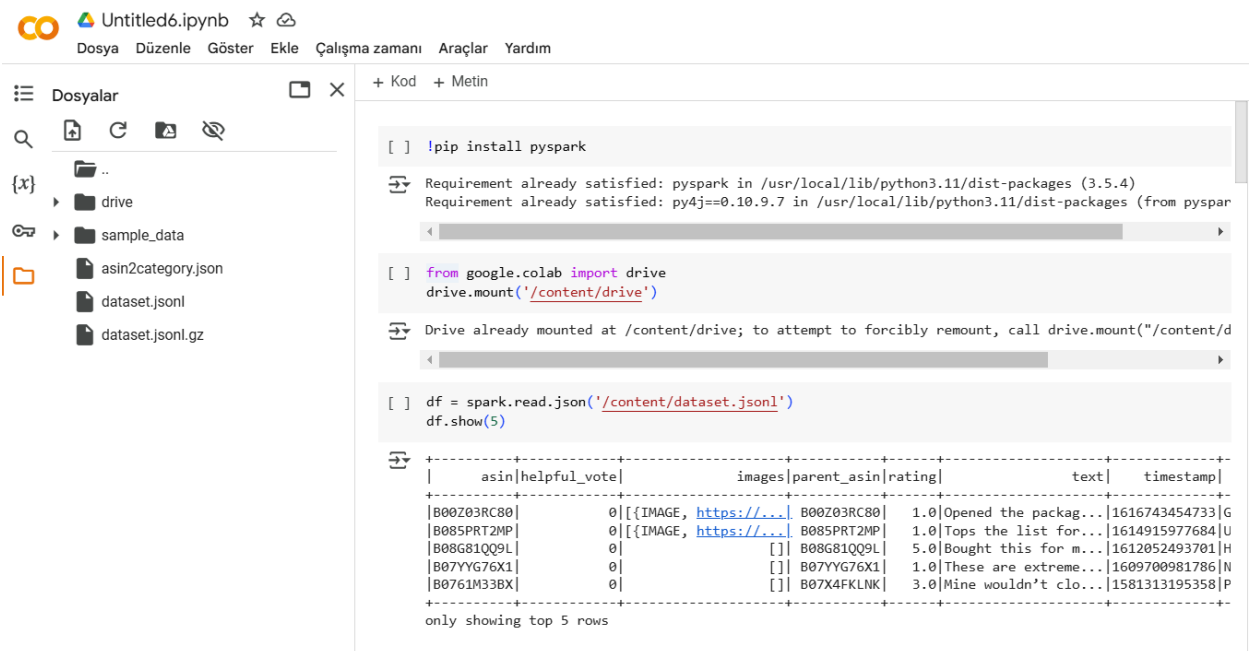


Figure 22 – Python codes for Spark

For the Spark job via Python, we have to load pyspark on Google Colab. I already uploaded jsonl dataset on my drive and the code reflects this connection. 'from Google.Colab import drive / drive.mount('/content/drive')'

Then I made the connection between the spark and the path of my dataset. Df.show(5) codes show the dataset's only 5 rows.



```
import gzip
import json
import os

input_gz_file = "/content/dataset.jsonl.gz"
output_jsonl_file = "/content/dataset.jsonl"

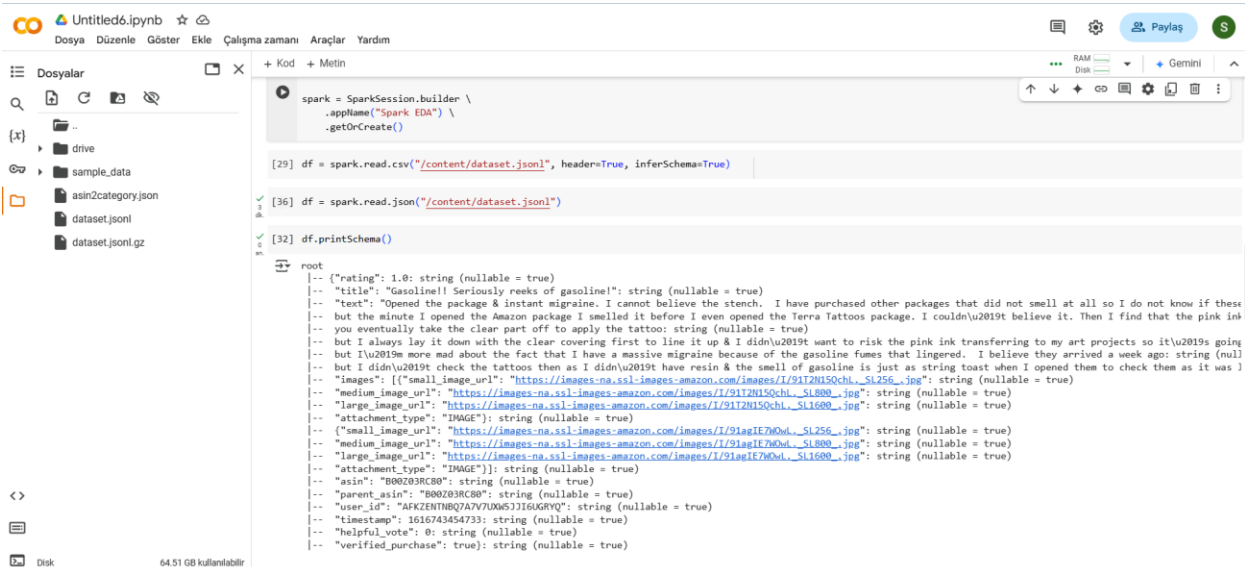
with gzip.open(input_gz_file, 'rt', encoding='utf-8') as f_in:
    with open(output_jsonl_file, 'w', encoding='utf-8') as f_out:
        for line in f_in:
            f_out.write(line)

print("JSONL dosyası başarıyla açıldı ve kaydedildi:", output_jsonl_file)
```

JSONL dosyası başarıyla açıldı ve kaydedildi: /content/dataset.jsonl

Figure 23 – Python codes for Spark

My dataset was jsonl.gz files. This file is just for the Java file and Python can not read and process this file that's why firstly I have to extract this file as only jsonl file. Print kod: print("JSONL dosyası başarıyla açıldı ve kaydedildi:", output_jsonl_file) writing as a Turkish and it's means JSONL files extracted successfully and saved.



```
spark = SparkSession.builder \
    .appName("Spark EDA") \
    .getOrCreate()

[29] df = spark.read.csv("/content/dataset.jsonl", header=True, inferSchema=True)

[36] df = spark.read.json("/content/dataset.jsonl")

[32] df.printSchema()

root
  |-- {"rating": 1.0: string (nullable = true)
  |-- "title": "Gasoline! Seriously reeks of gasoline!": string (nullable = true)
  |-- "text": "Opened the package & instant migraine. I cannot believe the stench. I have purchased other packages that did not smell at all so I do not know if these
  |-- but the minute I opened the Amazon package I smelled it before I even opened the Terra Tattoos package. I couldn't believe it. Then I find that the pink ink
  |-- you eventually take the clear part off to apply the tattoo: string (nullable = true)
  |-- but I always lay it down with the clear covering first to line it up & I didn't want to risk the pink ink transferring to my art projects so it's going
  |-- but I always lay it down with the clear covering first to line it up & I didn't want to risk the pink ink transferring to my art projects so it's going
  |-- but I didn't check the tattoos then as I didn't have resin & the smell of gasoline is just as strong as the smell of the gasoline fumes that lingered. I believe they arrived a week ago: string (null
  |-- "images": [{"small_image_url": "https://images-na.ssl-images-amazon.com/images/I/91T2H15Qchl_Sl256_.jpg": string (nullable = true)
  |-- "medium_image_url": "https://images-na.ssl-images-amazon.com/images/I/91T2H15Qchl_Sl800_.jpg": string (nullable = true)
  |-- "large_image_url": "https://images-na.ssl-images-amazon.com/images/I/91T2H15Qchl_Sl1600_.jpg": string (nullable = true)
  |-- "attachment_type": "IMAGE": string (nullable = true)
  |-- {"small_image_url": "https://images-na.ssl-images-amazon.com/images/I/91agIE7W0dL_Sl256_.jpg": string (nullable = true)
  |-- "medium_image_url": "https://images-na.ssl-images-amazon.com/images/I/91agIE7W0dL_Sl800_.jpg": string (nullable = true)
  |-- "large_image_url": "https://images-na.ssl-images-amazon.com/images/I/91agIE7W0dL_Sl1600_.jpg": string (nullable = true)
  |-- "attachment_type": "IMAGE": string (nullable = true)
  |-- "asin": "B00Z03RCB0": string (nullable = true)
  |-- "parent_asin": "B00Z03RCB0": string (nullable = true)
  |-- "user_id": "AFKZENTNBQ7A7V7U065J3IGUGRQV": string (nullable = true)
  |-- "timestamp": 1616743454733: string (nullable = true)
  |-- "helpful_vote": 0: string (nullable = true)
  |-- "verified_purchase": true: string (nullable = true)
```

Figure 24 - Python codes for Spark

With df.printschema() code, I printed the schema so that I could see the context of the dataset.

```

[32] -- "helpful_vote": 0: string (nullable = true)
      -- "verified_purchase": true: string (nullable = true)

filtered_df = df.filter(df["rating"] > 1)

filtered_df.show()

```

asin	helpful_vote	images	parent_asin	rating	text	timestamp	title	user_id	verified_purchase
B08G81QQ9L	0	[[]]	B08G81QQ9L	5.0	Bought this for m...	1612052493701	Hailey loves unic...	AFKZENTNBQ7A7V7UX...	true
B0761M33BX	0	[[]]	B07X4FKLNK	3.0	Mine wouldn't clo...	1581313195358	Pretty, but didn'...	AFKZENTNBQ7A7V7UX...	true
B00YAZBWZ1	0	[[]]	B00YAZBWZ1	5.0	This stuff smells...	1458095420000	Five Stars	AGKASBHYZPGTEPO6L...	true
B007Z2R1S1	0	[[]]	B007Z2R1S1	3.0	I have used PFB v...	1458094710000	OK	AGKASBHYZPGTEPO6L...	true
B007IIUJ5Q	0	[[]]	B007IIUJ5Q	4.0	Nice set at a rea...	1454675735000	Neutral set flatt...	AGKASBHYZPGTEPO6L...	true
B00PA7VMD2	0	[[]]	B00PA7VMD2	3.0	These are very cu...	1452648690000	The clips vary...	AGKASBHYZPGTEPO6L...	true
B00V6R3R35	1	[[]]	B00V6R3R35	5.0	Beautiful palette...	1452647102000	Beautiful palette...	AGKASBHYZPGTEPO6L...	true
B00CS4HTAC	4	[[]]	B00CS4HTAC	5.0	The scent is fres...	1671844437231	My favorite color...	AG2L7H23R5LLKXLB...	true
B000XEBX08	1	[[]]	B000XEBX08	5.0	I used to work fo...	1593933886664	PSA TO ALL CRANIN...	AG2L7H23R5LLKXLB...	false
B0871S95M5	1	[IMAGE, https://...]	B0871S95M5	5.0	With 3 girls, we ...	1637522137428	Great size and qu...	AGC17FAH4GL5FI6SH...	true
B010TN80W	0	[[]]	B010TN80W	5.0	super useful	1456772214000	Five Stars	AGC17FAH4GL5FI6SH...	true
B000VT54QA	0	[[]]	B000VT54QA	3.0	I have thick hair...	1408993548000	Doesn't work for ...	AGC17FAH4GL5FI6SH...	true
B001ET76EY	1	[[]]	B0C43HKQM	5.0	cetaphil is like ...	1384912580000	good	AGC17FAH4GL5FI6SH...	true
B000M2HGQ4	0	[[]]	B000M2HGQ4	5.0	great product. d...	1384912377000	good	AGC17FAH4GL5FI6SH...	true
B019QLR74	0	[[]]	B083BSOV9F	5.0	Bought this for m...	1504010693432	Great purchase! L...	AGXVB1UFLFGW1ATY...	true
B00YQJHRA2	0	[[]]	B00YQJHRA2	3.0	Not as easy to ap...	1485870262000	Three Stars	AGXVB1UFLFGW1ATY...	true
B0069ET6CA	1	[[]]	B08A2F06Z2	5.0	Been using this p...	1445968681000	Five Stars	AGXVB1UFLFGW1ATY...	true
B071J0W4T9	0	[[]]	B0877ZL2VP	5.0	Smells so good bu...	1588687594456	What a nice scrub...	AGKHLEW2SONHMFQI...	true
B006BL0W64	2	[[]]	B08HPTZ54G	5.0	It smells Heaven...	1588615673460	This stuff is a l...	AGKHLEW2SONHMFQI...	true
B00IO334BN	0	[[]]	B076QMF18	3.0	I love this brand...	1558395642698	this smells weird	AGKHLEW2SONHMFQI...	true

only showing top 20 rows

Figure 25 - Python codes for Spark

As seen in Figure 21, I filtered the dataset and showed ratings > 1.

For the Data Aggregation, I prefer to calculate the average rating of specific titles. The results are shown in Figure 22.

```

from pyspark.sql import functions as F

df_grouped = df.groupBy("title").agg(
    F.avg("rating").alias("average_rating")
)

df_grouped.show()

```

title	average_rating
Most amazing eyel...	5.0
Love It !!	4.968421052631579
Wig holder frames	5.0
wide tooth combs	5.0
clear disposable ...	5.0
Other products on...	3.0
For those with Ps...	5.0
Wonderfully fresh...	5.0
works great! Vac...	4.0
Nice Toner!	5.0
Love Love Love!	4.976119402985074
Did not smell good	1.5555555555555556
Amazing!	4.973438696651117
Nice smelling soap	4.478260869565218
Great basic condi...	5.0
Moisturizes and s...	5.0
It works!!!!	4.939008042895442
RIP OFF	1.1581920903954803
The gold shimmer ...	2.0
Works Wonders	4.945244956772334

only showing top 20 rows

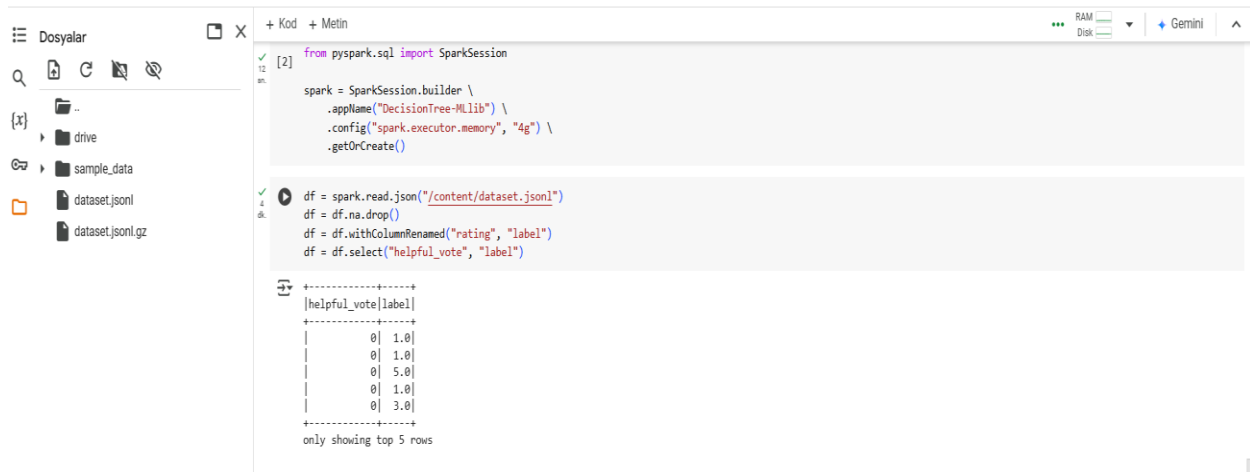
Figure 26 - Python codes for Spark

If I evaluate the performance of MapReduce and Spark, Spark's performance was faster than MapReduce as expected. If we compare based on the Python codes, Spark requires more simple codes than MapReduce because MapReduce needs a map and a reduce function and these requirements affect both speed and simplicity.

CHAPTER FOUR

TASK 4

For machine learning algorithm, I choose decision tree model because our dataset has numeric and categoric data. Decision tree model is fits for both data type.



The screenshot shows a Jupyter Notebook environment. On the left, a file explorer shows a directory structure with files like 'dataset.jsonl' and 'dataset.jsonl.gz'. The main area contains two code cells. The first cell imports SparkSession and configures it. The second cell reads a JSON dataset, drops nulls, renames a column, and selects specific columns. Below the code, a preview of the data is shown as a table with two columns: 'helpful_vote' and 'label'.

```
[2] from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("DecisionTree-MLlib") \
    .config("spark.executor.memory", "4g") \
    .getOrCreate()

df = spark.read.json("/content/dataset.jsonl")
df = df.na.drop()
df = df.withColumnRenamed("rating", "label")
df = df.select("helpful_vote", "label")

+-----+-----+
|helpful_vote|label|
+-----+-----+
|          0| 1.0|
|          0| 1.0|
|          0| 5.0|
|          0| 1.0|
|          0| 3.0|
+-----+-----+
only showing top 5 rows
```

Figure 27 – Python code for machine learning

CONCLUDING REMARKS

Big data analytics is critically important for businesses and even governments in today's data-driven world. The analyses conducted in this study provide significant benefits to businesses. With big data technologies, customer behaviors can be analyzed more effectively, operational efficiency can be increased, and future predictions can be made more accurately.

The analyses performed using the selected big data set have provided valuable insights. Among the big data processing methods, Spark and MapReduce have different approaches to big data analytics. MapReduce processes data by breaking it into chunks in a batch-processing manner, whereas Spark enables faster and more efficient analyses through in-memory computing capabilities. While Spark is preferred for real-time data processing and interactive analytics, MapReduce provides the advantage of processing large-scale datasets with lower memory requirements.

In conclusion, when used with the right tools and methodologies, big data analytics has great potential to improve decision-making processes and create strategic advantages. Technologies such as Spark and MapReduce enhance the effectiveness of big data processing by offering solutions suitable for different scenarios.

BIBLIOGRAPHY



APPENDIX

- ✓ https://mcauleylab.ucsd.edu/public_datasets/data/amazon_2023/raw/review_categories/Beauty_and_Personal_Care.jsonl.gz
- ✓ <https://colab.research.google.com/drive/1VCacooTP8bweI5I77qEylWdm8MQP6TPh#scrollTo=ElarG1PMSgOH>