University for the Creative Arts

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.

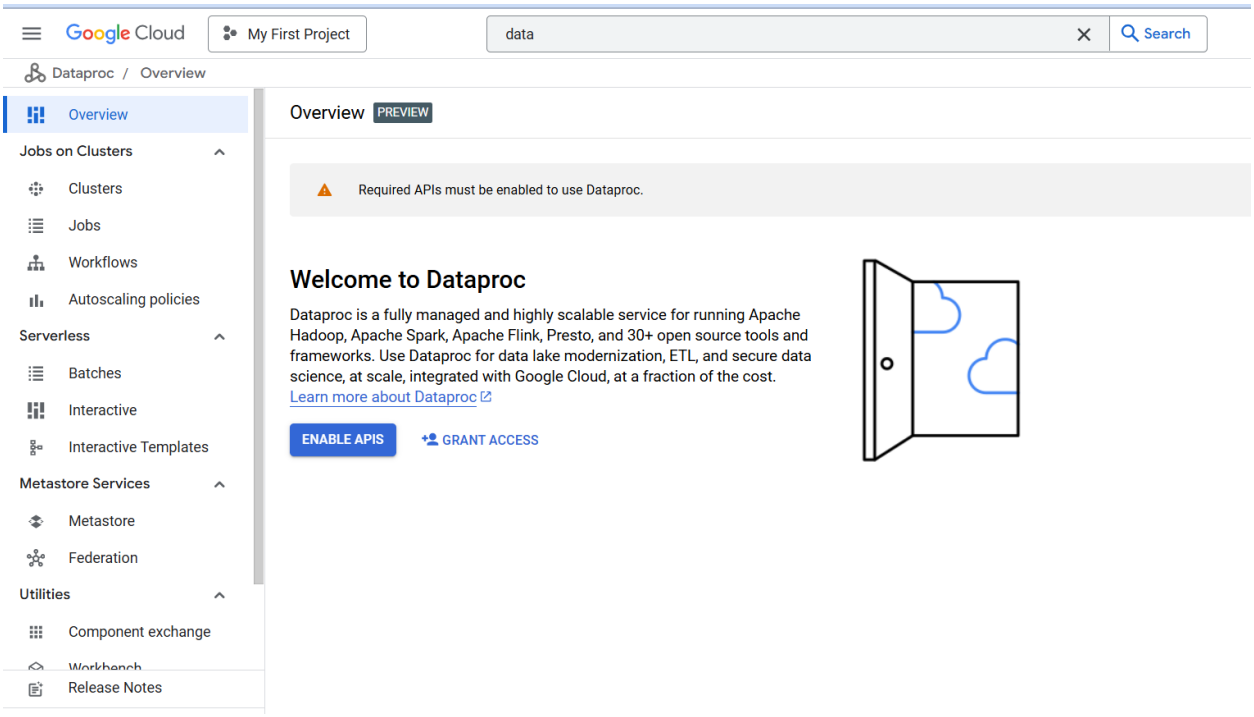


Figure 1 – Searching DataProc

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.

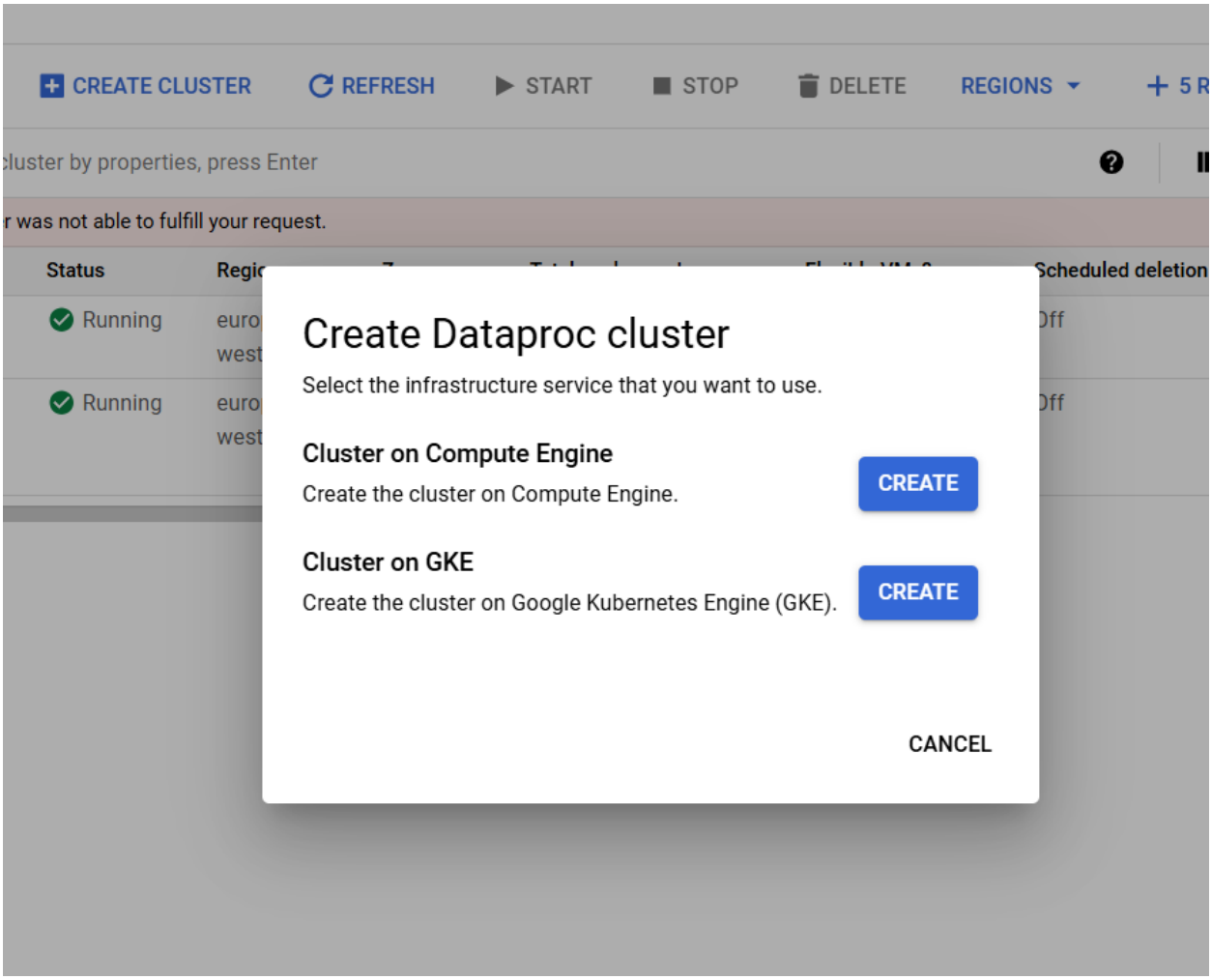


Figure 2 – Creating Cluster

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

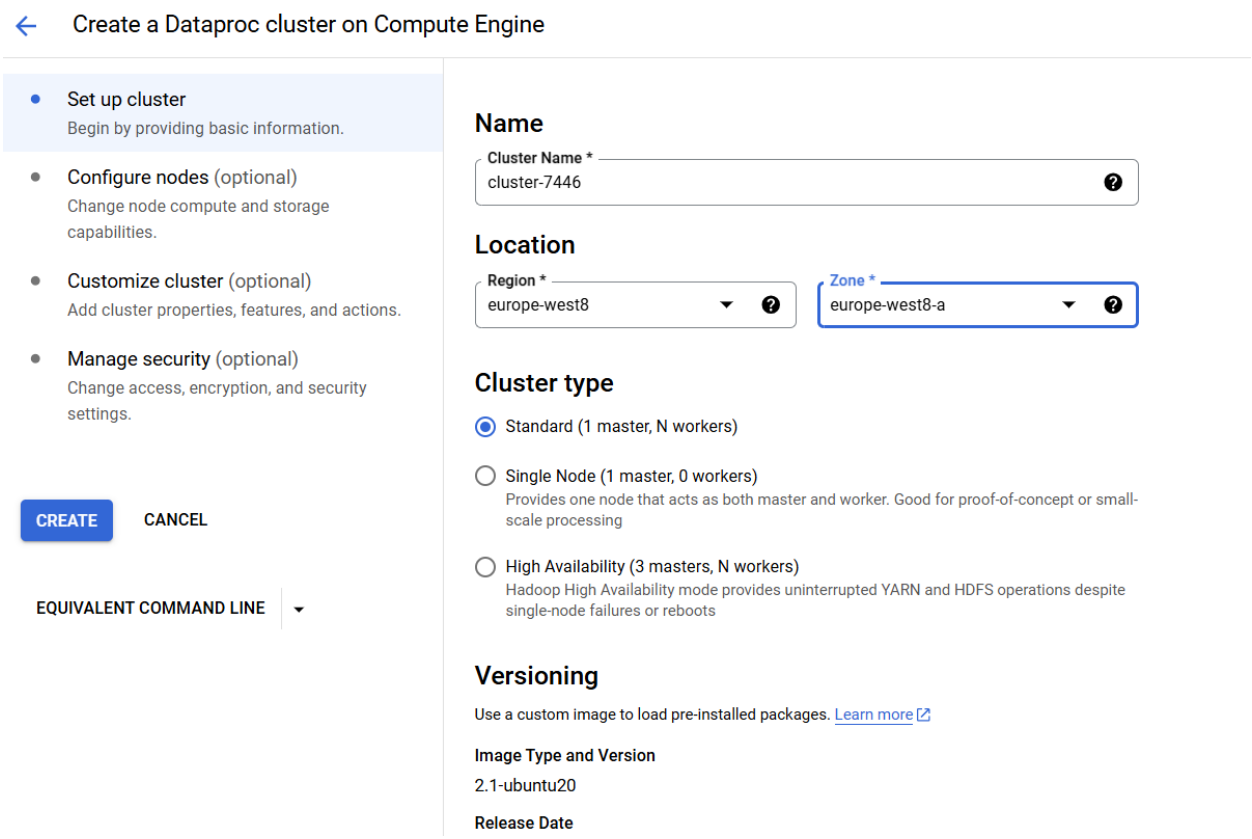


Figure 3 – Creating Cluster

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.

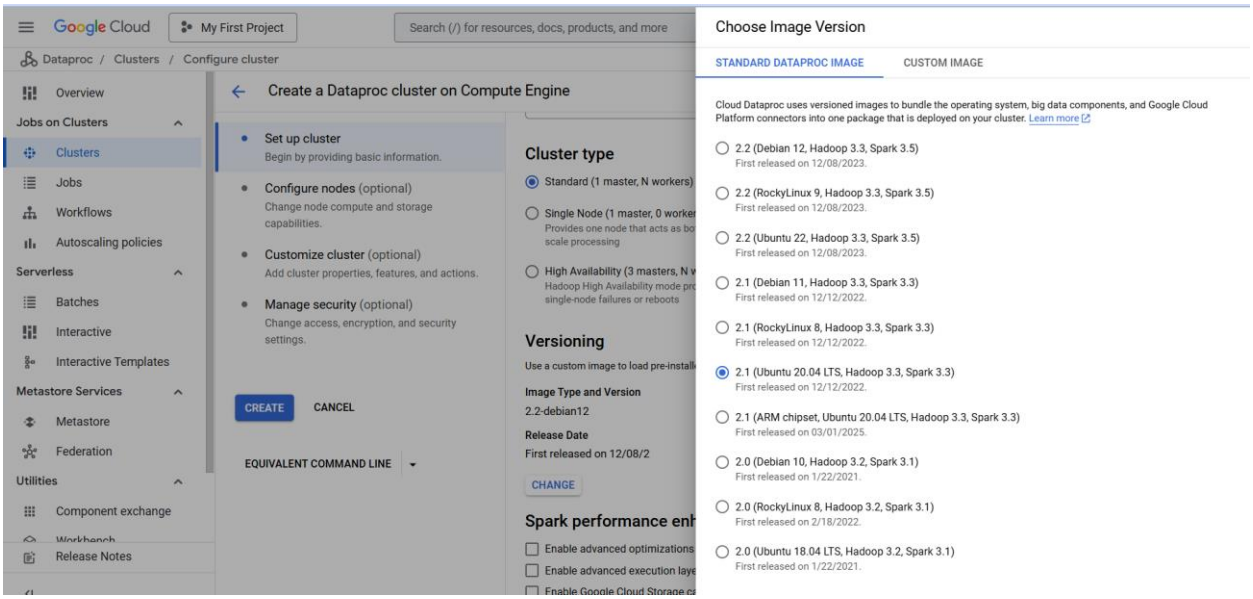


Figure 4 – Creating Cluster

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

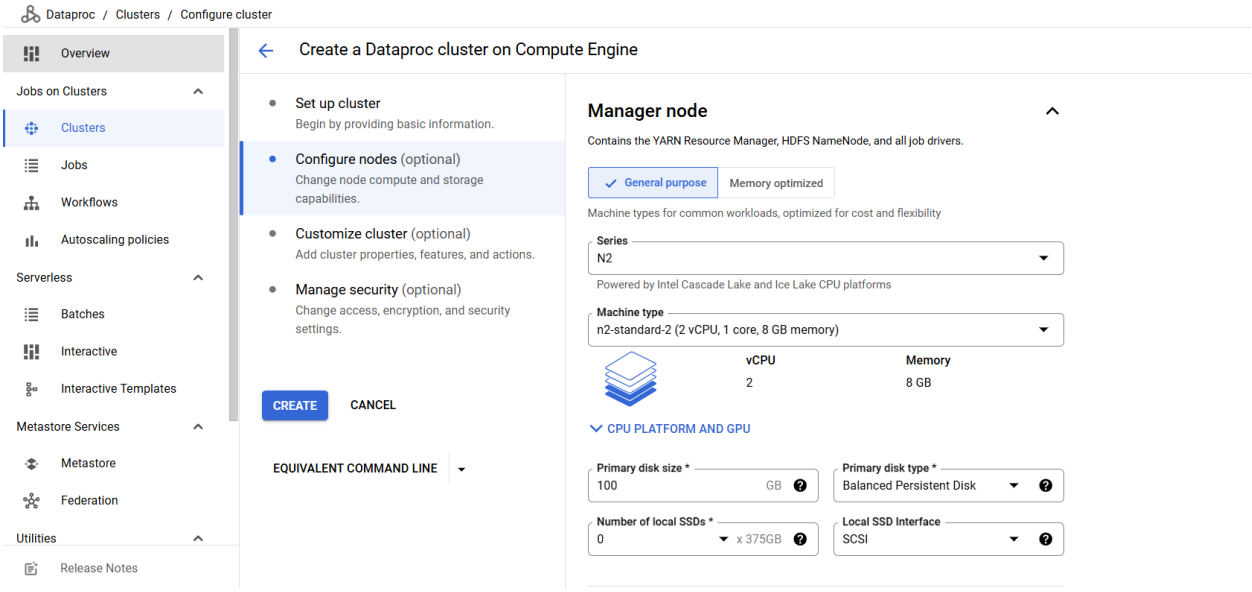


Figure 5 – Cluster Settings

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

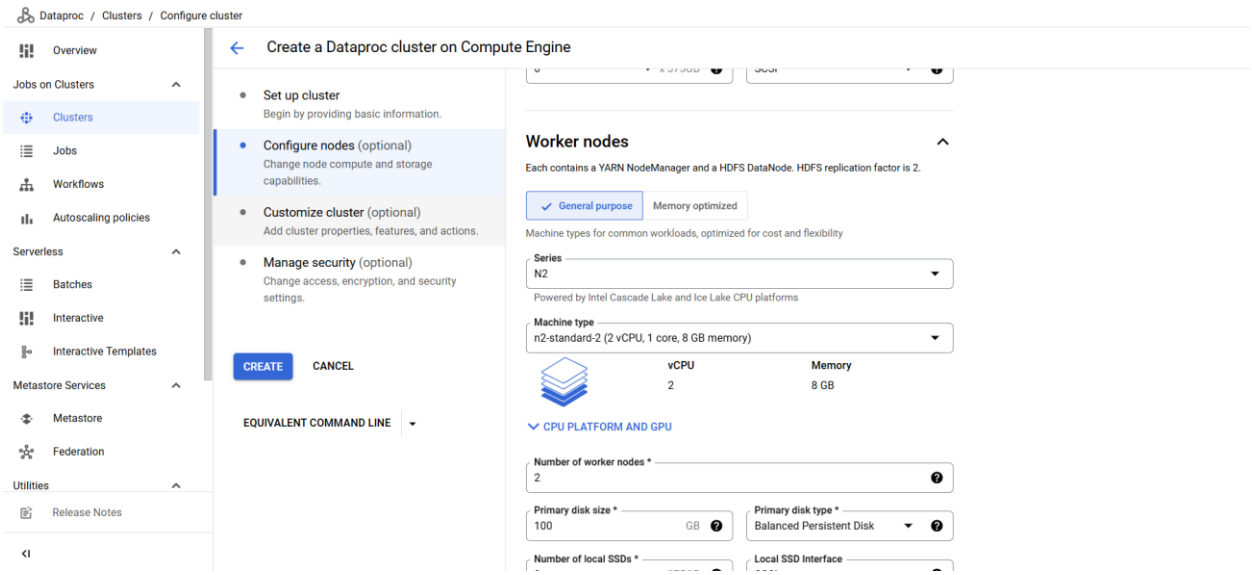


Figure 6 – Cluster settings

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.

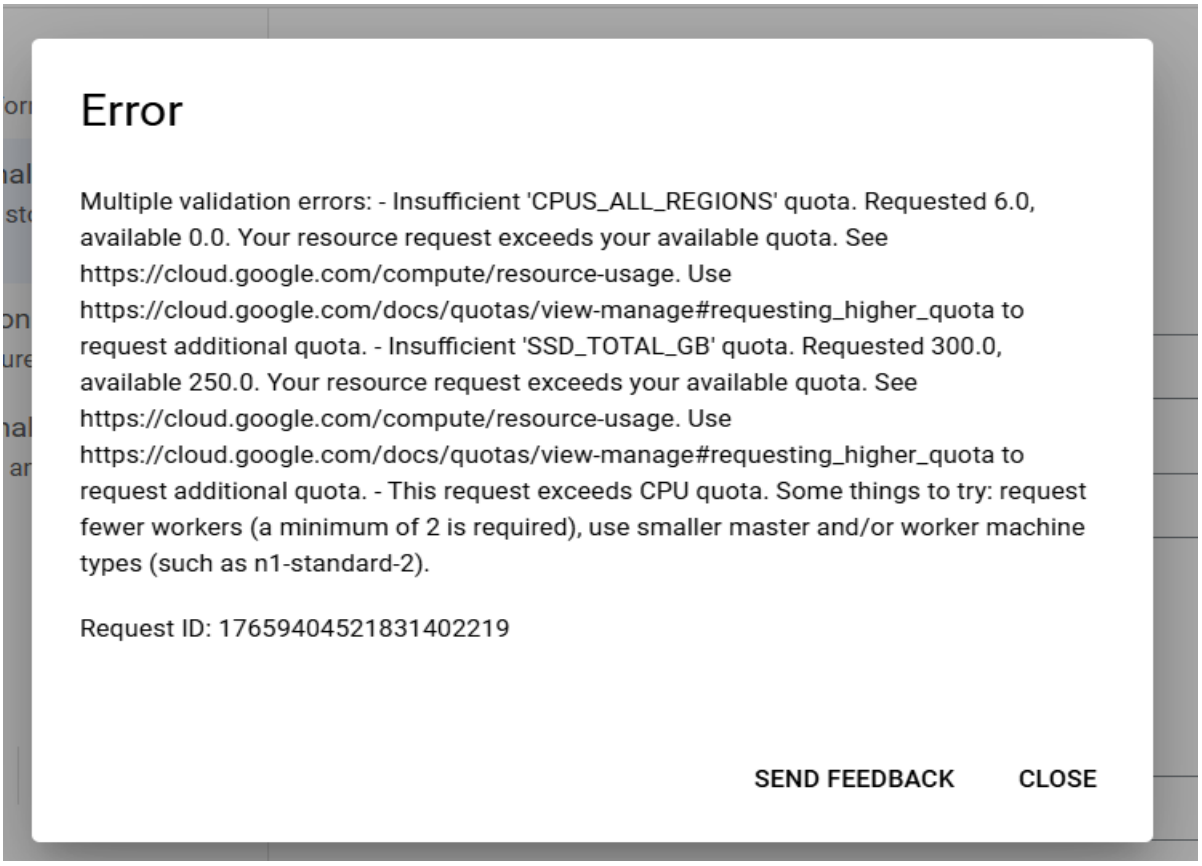


Figure 7 – Error message while creating the cluster

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.

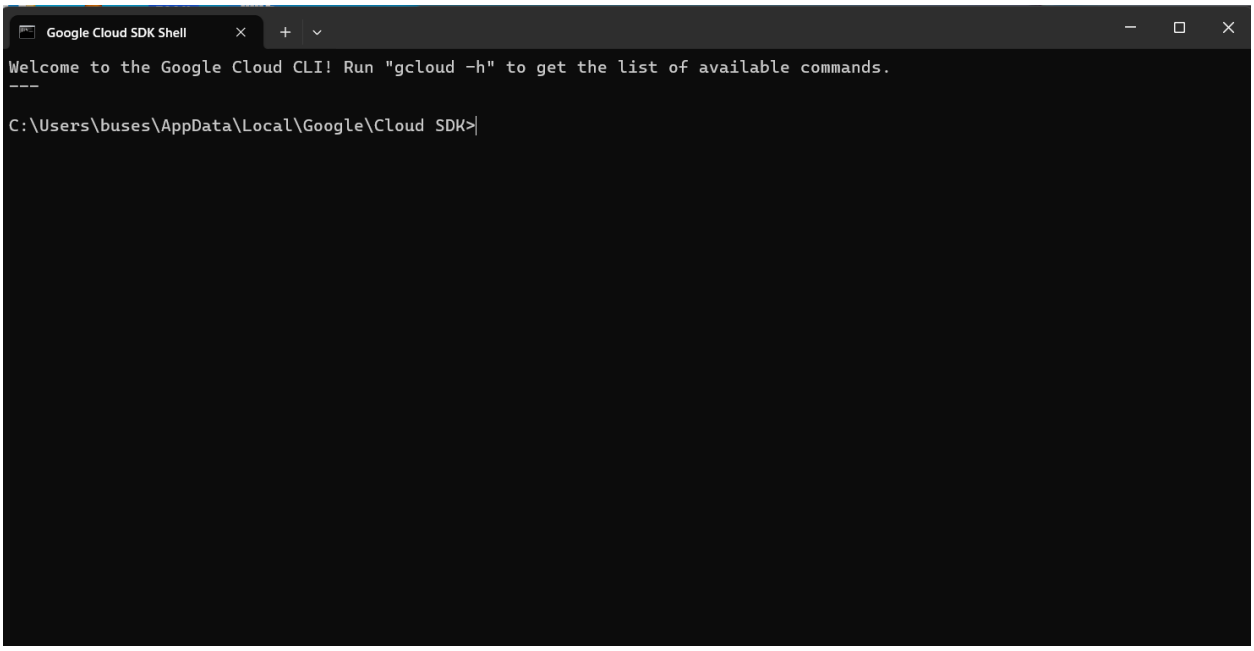


Figure 8 – Google Cloud SDK

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

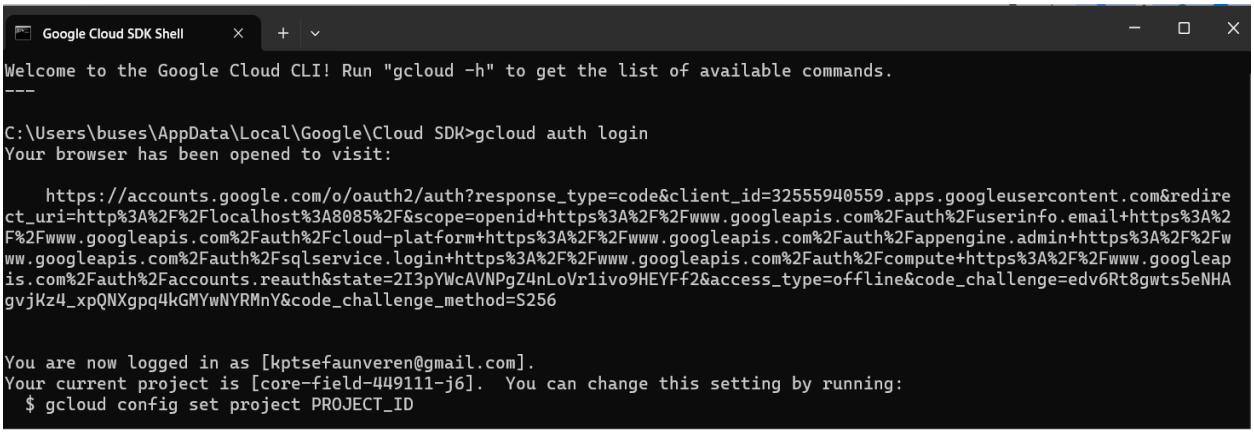


Figure 9 - Google Cloud SDK

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=eu
rope-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 quo
ta. 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 'IN_USE_ADDRESSES' quota. Requested 3.0, available 1.0. Your resource request exceeds your available quo
ta. See https://cloud.google.com/compute/resource-usage. Use https://cloud.google.com/docs/quotas/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 m
aster 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.

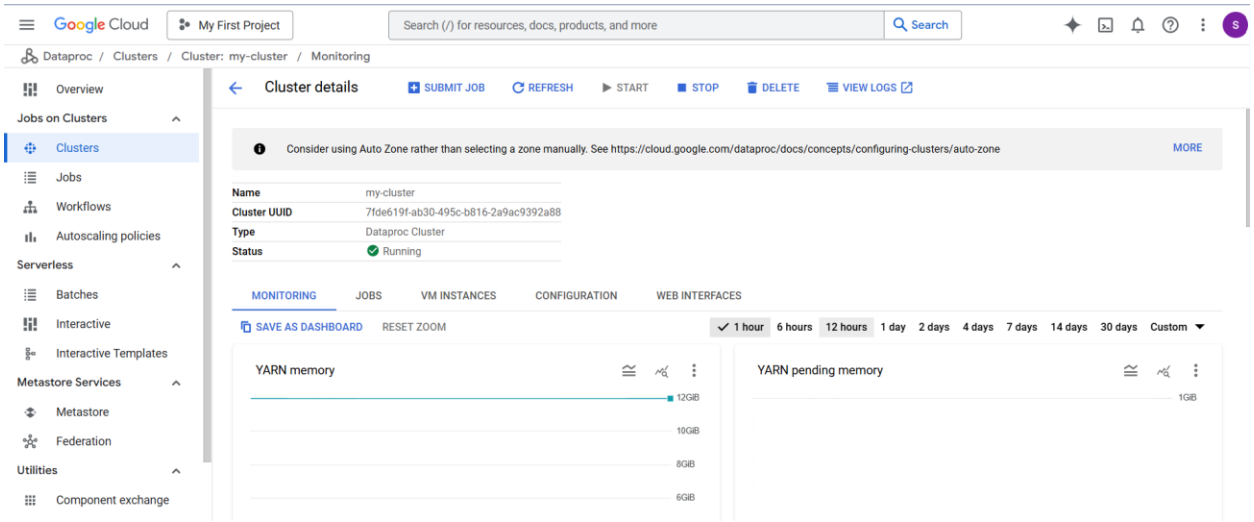


Figure 11 – My Cluster

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

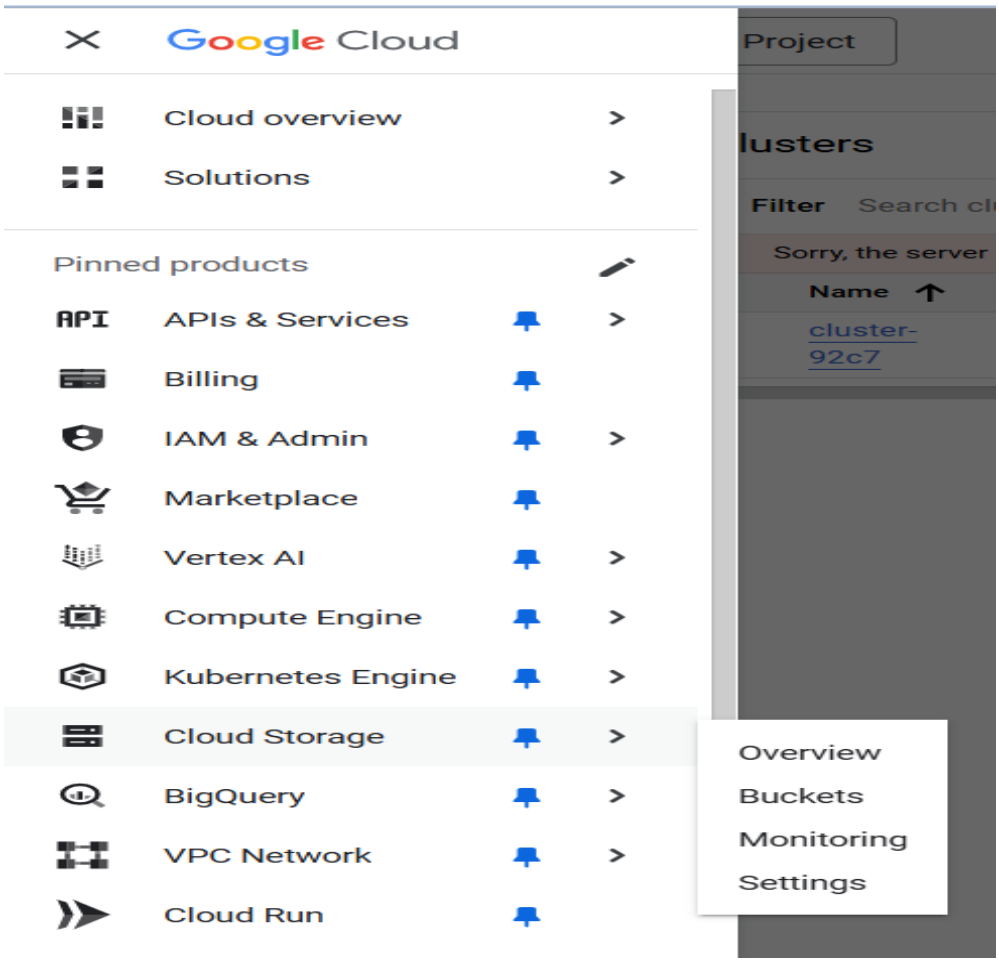


Figure 12 – Create Buckets

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.

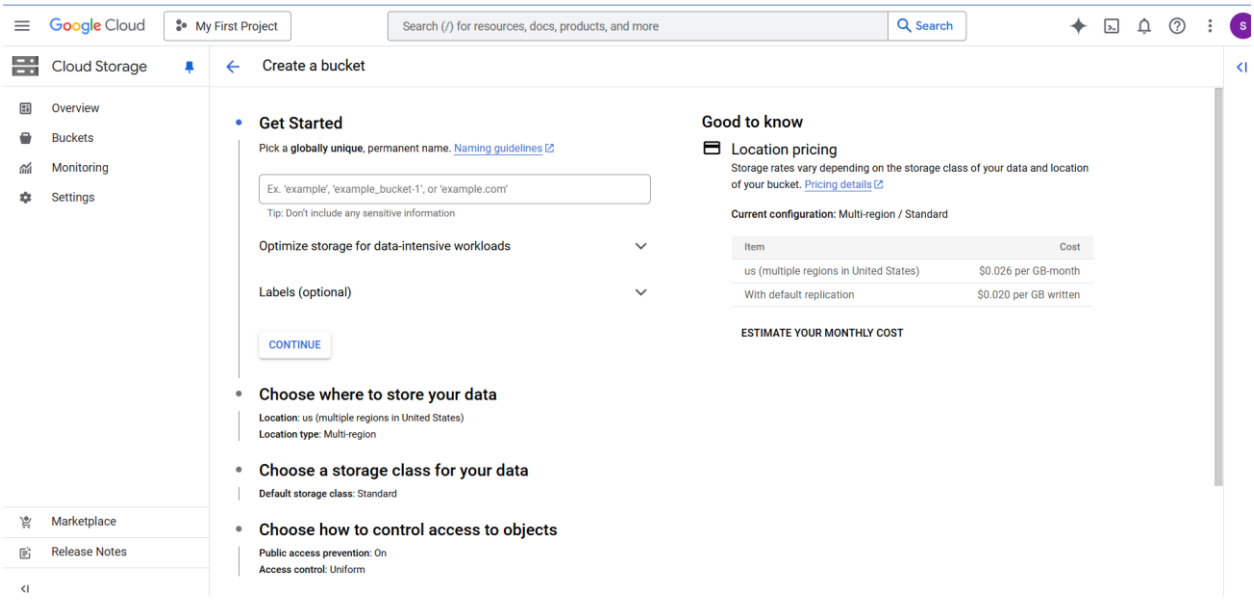


Figure 13 – Creating Bucket

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.

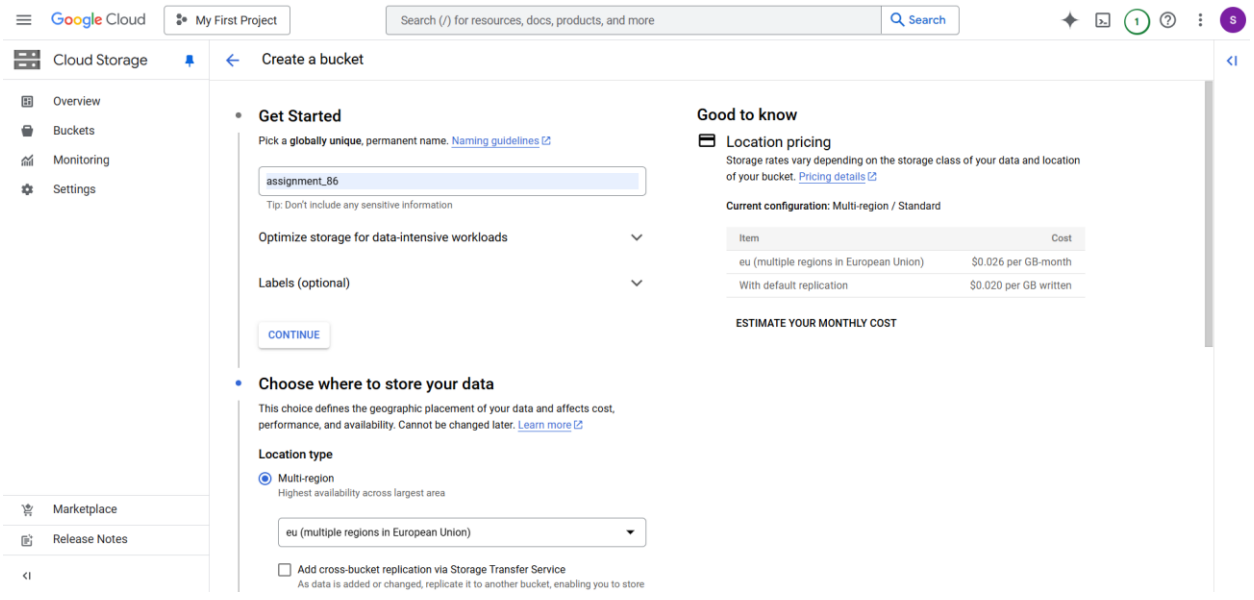


Figure 14 – Creating Bucket

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

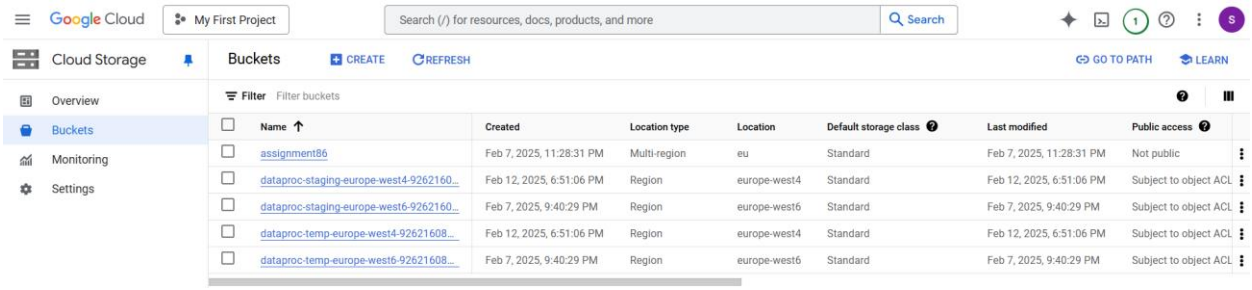


Figure 15 – Bucket List

Now we can upload our dataset in these buckets.

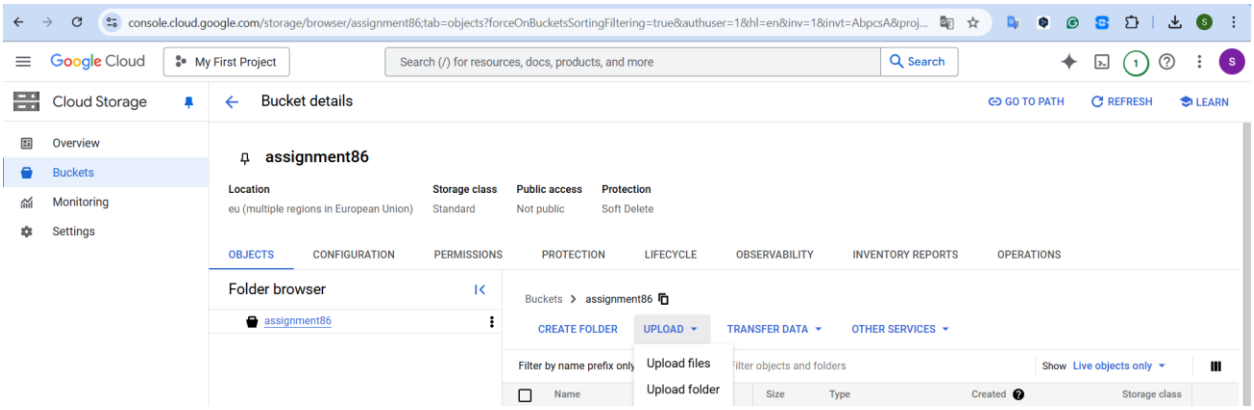


Figure 16 – Upload Files

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.

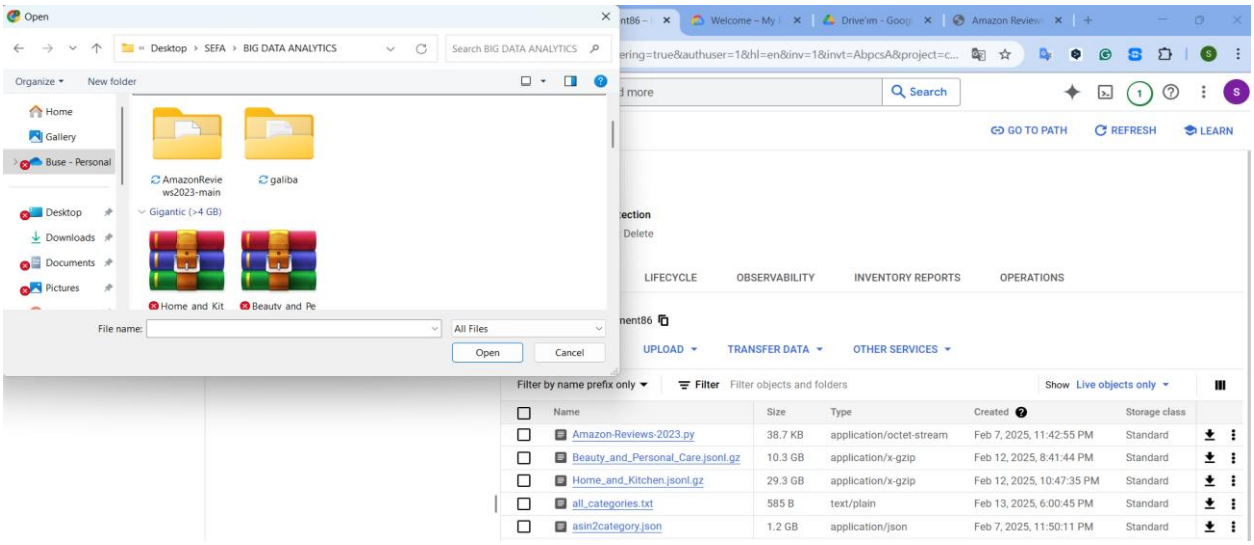


Figure 17 – Upload Files

# 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.

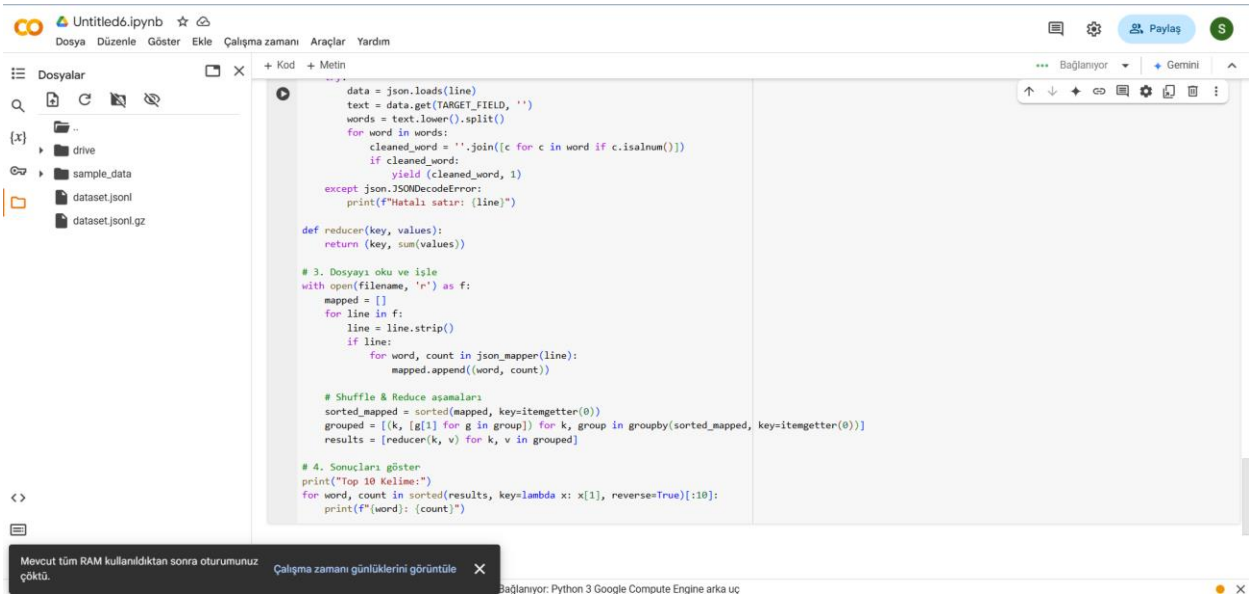


Figure 18 – MapReduce Error

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 succeded to analyze.


Figure 19 – Python Codes for MapReduce

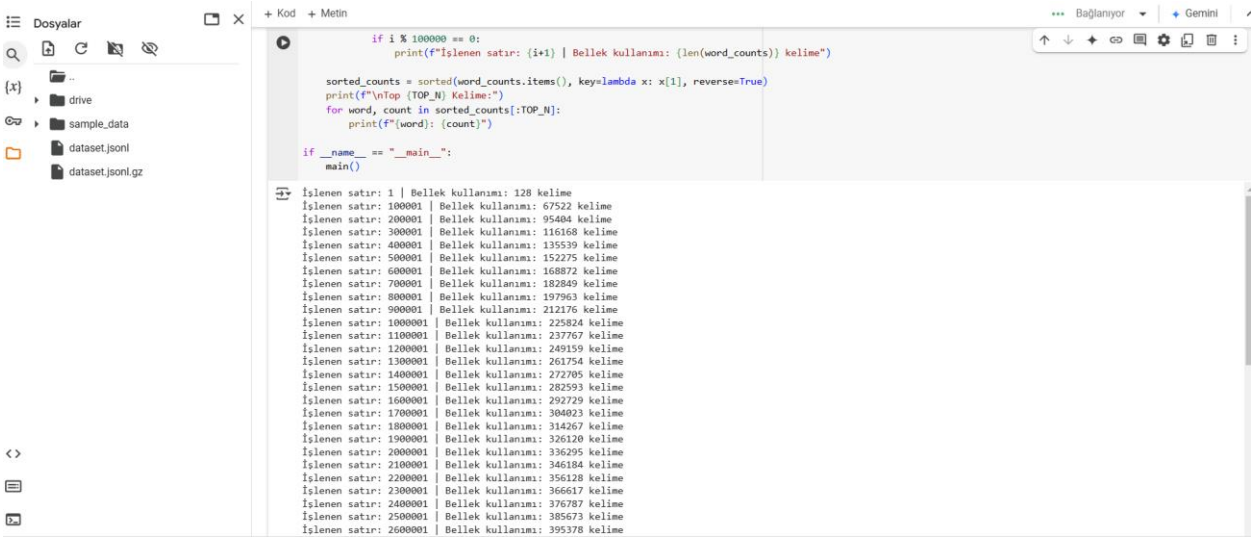


Figure 20 - Python Codes for MapReduce


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.

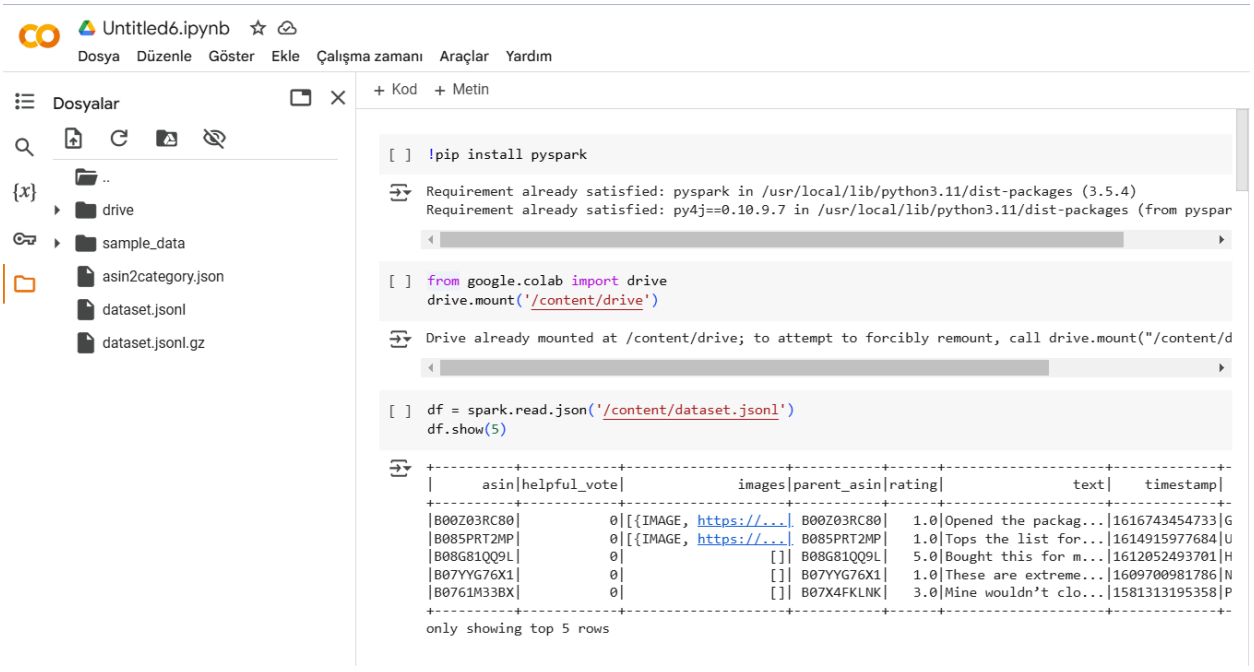


Figure 22 – Python codes for Spark

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.


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 ıt’s means JSONL files extracted successfully and saved.

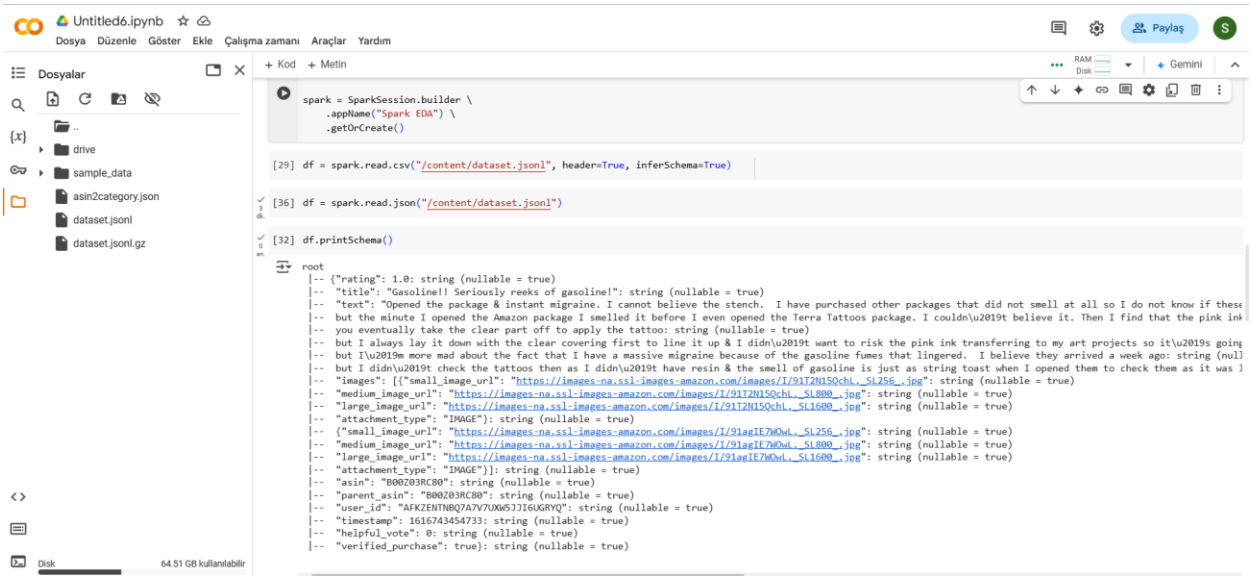


Figure 24 - Python codes for Spark

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

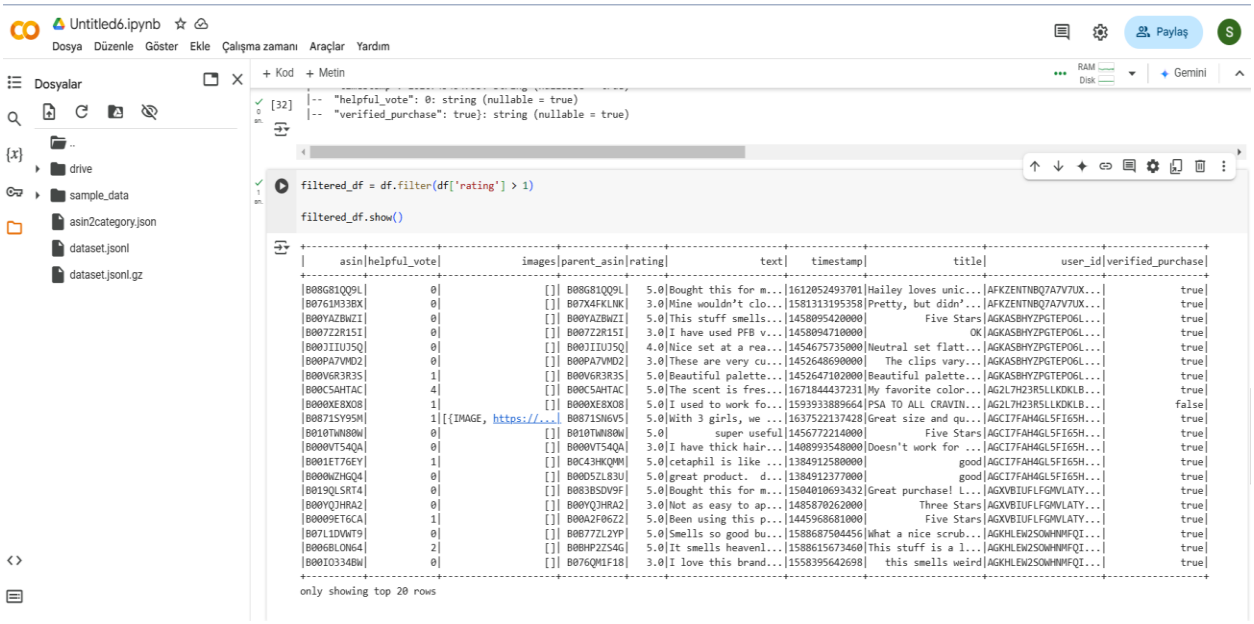


Figure 25 - Python codes for Spark

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.

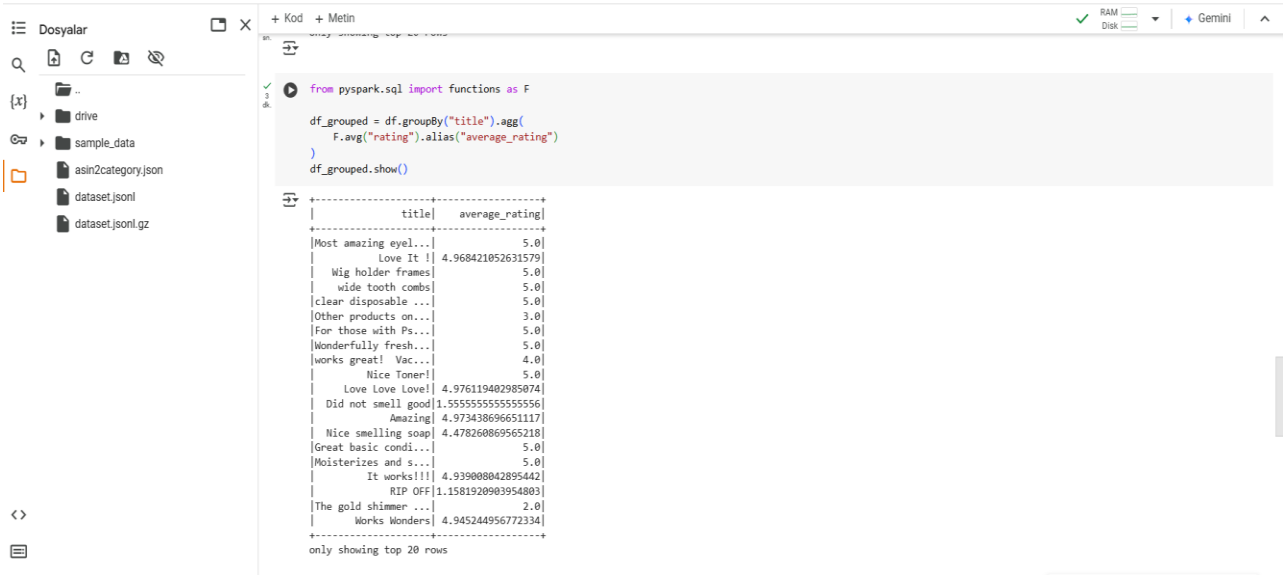


Figure 26 - Python codes for Spark

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


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/1VCacooTP8bwel5I77qEylWdm8MQP6TPh#scrollTo=ElarG1PMSgOH