Introduction to Big Data Graded Assignment

Name - Avijeet Palit
Roll - 21f1005675
Date - 22/11/2024

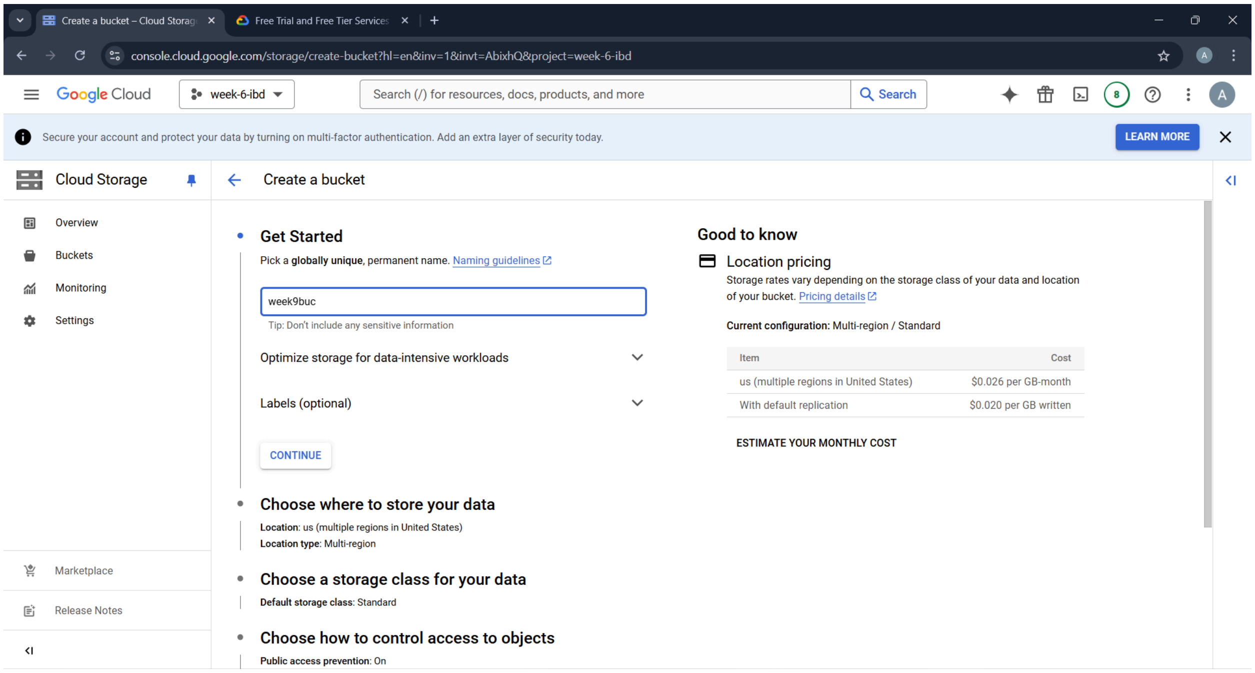
Objective

Convert the Spark MLib code from the Databricks decision trees example to use the CrossValidator autotuner. Analyze and report the best-performing model parameters.

Detailed Steps

Step 1: Create a GCS Bucket

- Navigate to the Cloud Storage Section:
- Log in to your Google Cloud Platform (GCP) Console.
- Click on Cloud Storage from the navigation menu.
- Create a New Bucket:


Create a bucket – Cloud Storag
Free Trial and Free Tier Services
console.cloud.google.com/storage/create-bucket?hl=en&inv=1&invt=AbixhQ&project=week-6-ibd
Google Cloud
week-6-ibd
Search (/) for resources, docs, products, and more
Search
8
A
LEARN MORE
Cloud Storage
Overview
Buckets
Monitoring
Settings
Marketplace
Release Notes

Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.

Create a bucket

Get Started

Pick a **globally unique**, permanent name. Naming guidelines

week9buc

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

Good to know

Location pricing

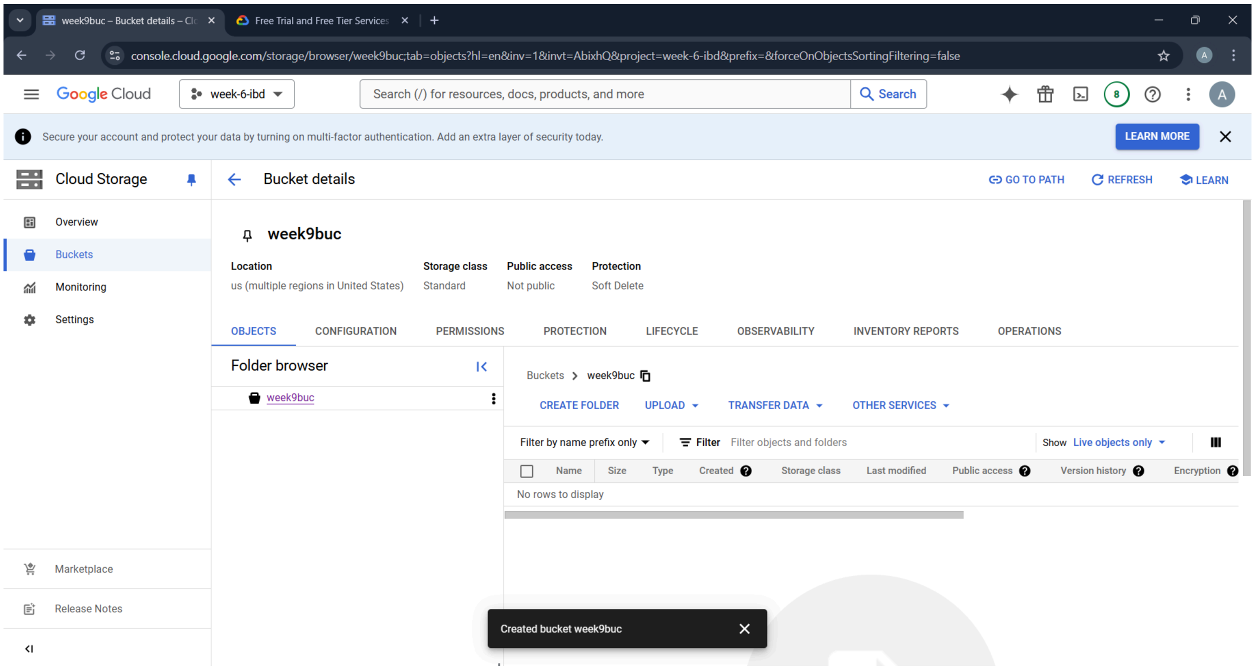
Storage rates vary depending on the storage class of your data and location of your bucket. Pricing details

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

- Provide a name for the bucket (e.g., "week9buc").
- Set Location and Permissions:
- Select the desired bucket location (e.g., Regional/Multiregional).



console.cloud.google.com/storage/browser/week9buc;tab=objects?hl=en&inv=1&invt=AbixhQ&project=week-6-ibd&prefix=&forceOnObjectsSortingFiltering=false

Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.

week9buc

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	Soft Delete

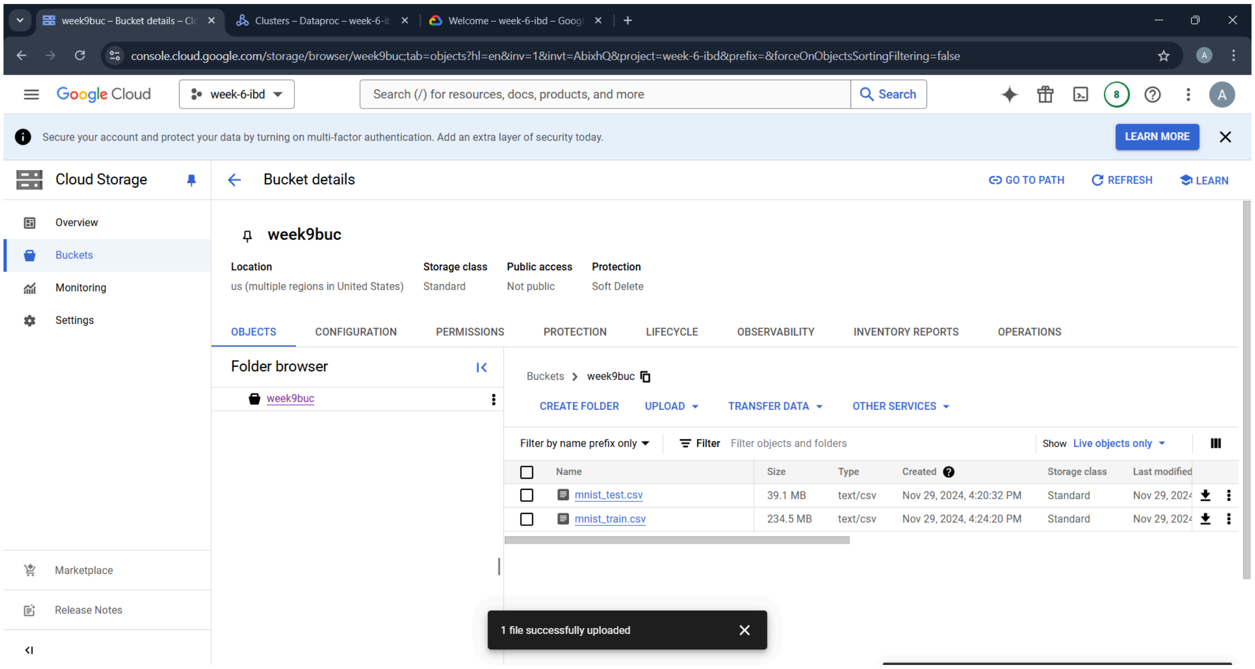
□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption

No rows to display

Created bucket week9buc

Step 2: Generate or Use MNIST Data

- Obtain MNIST Dataset:
- Either generate the MNIST Train and Test datasets or download them from a reliable source.



console.cloud.google.com/storage/browser/week9buc;tab=objects?hl=en&inv=1&invt=AbixhQ&project=week-6-ibd&prefix=&forceOnObjectsSortingFiltering=false

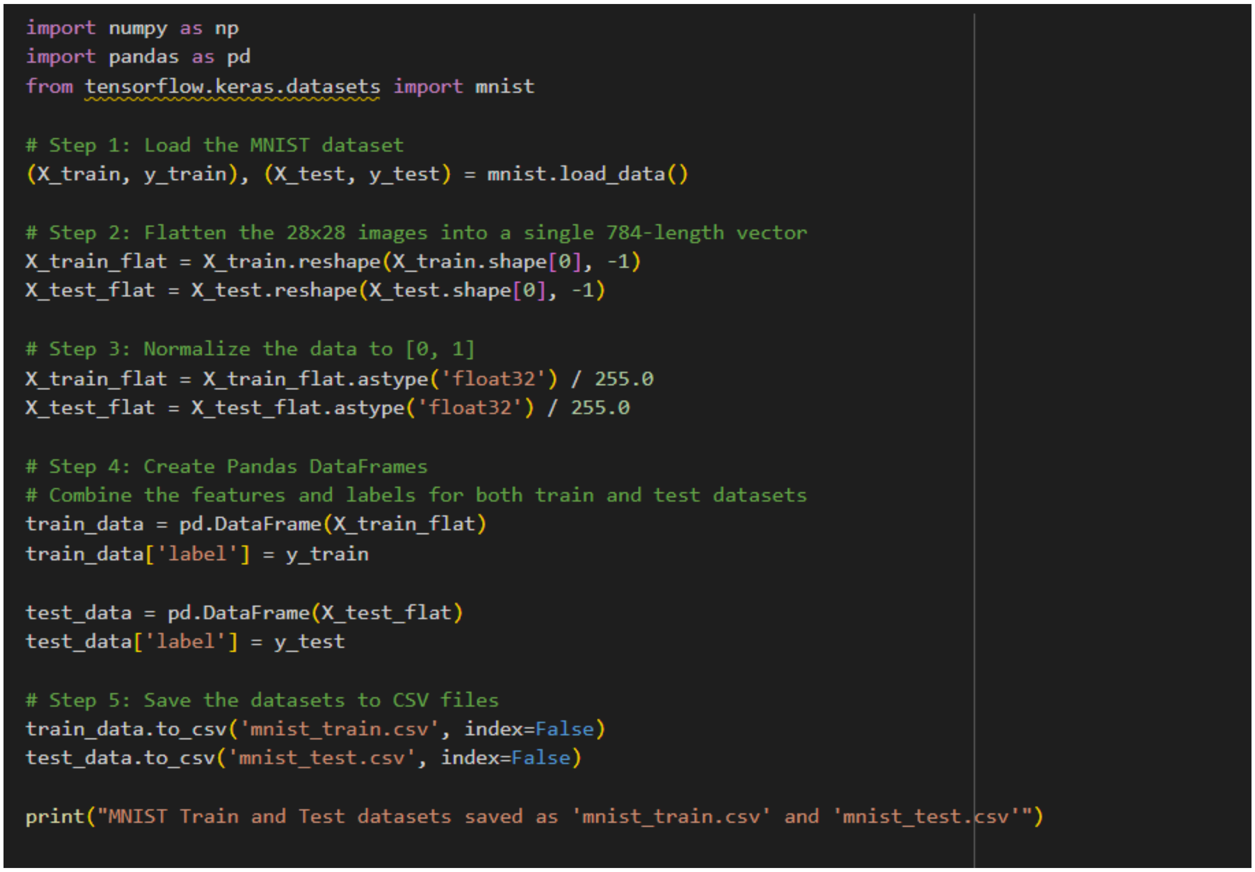
Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.

week9buc

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	Soft Delete

□	Name	Size	Type	Created	Storage class	Last modified
□	mnist_test.csv	39.1 MB	text/csv	Nov 29, 2024, 4:20:32 PM	Standard	Nov 29, 2024
□	mnist_train.csv	234.5 MB	text/csv	Nov 29, 2024, 4:24:20 PM	Standard	Nov 29, 2024

1 file successfully uploaded

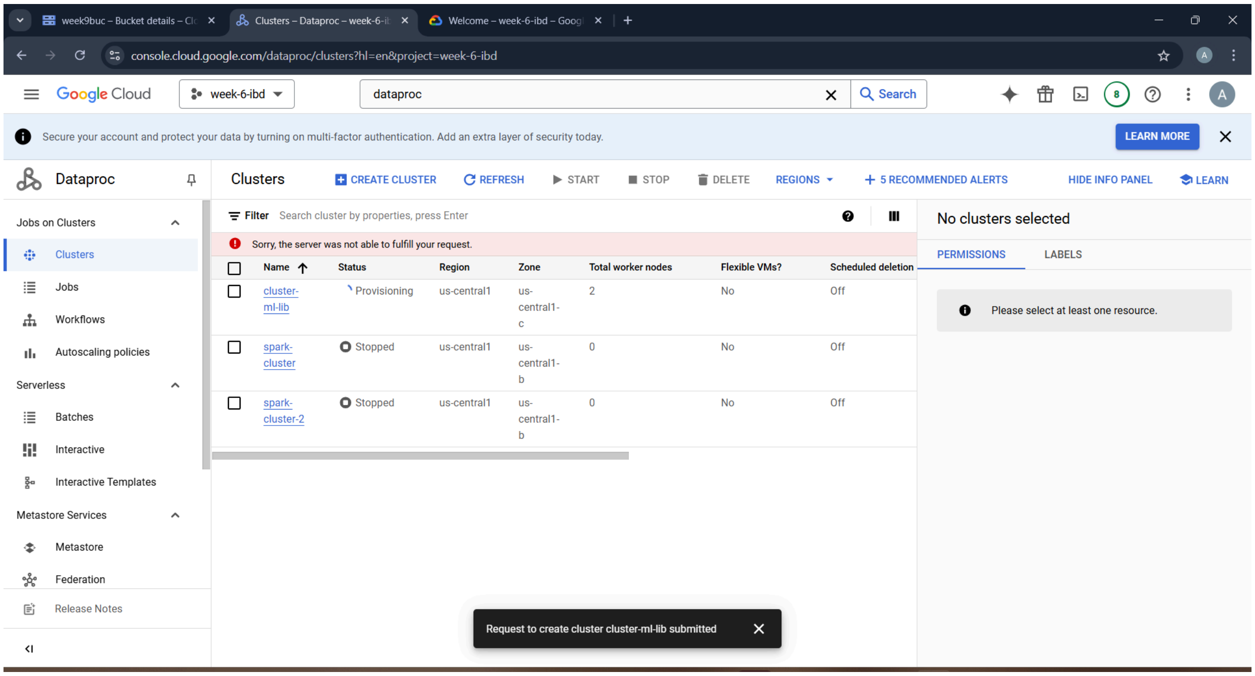


```
import numpy as np
import pandas as pd
from tensorflow.keras.datasets import mnist

# Step 1: Load the MNIST dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()

# Step 2: Flatten the 28x28 images into a single 784-length vector
X_train_flat = X_train.reshape(X_train.shape[0], -1)
X_test_flat = X_test.reshape(X_test.shape[0], -1)

# Step 3: Normalize the data to [0, 1]
X_train_flat = X_train_flat.astype('float32') / 255.0
X_test_flat = X_test_flat.astype('float32') / 255.0

# Step 4: Create Pandas DataFrames
# Combine the features and labels for both train and test datasets
train_data = pd.DataFrame(X_train_flat)
train_data['label'] = y_train

test_data = pd.DataFrame(X_test_flat)
test_data['label'] = y_test

# Step 5: Save the datasets to CSV files
train_data.to_csv('mnist_train.csv', index=False)
test_data.to_csv('mnist_test.csv', index=False)

print("MNIST Train and Test datasets saved as 'mnist_train.csv' and 'mnist_test.csv'")
```

- Upload Data to the GCS Bucket:

Step 3: Create a Dataproc Cluster

- Go to the Dataproc Section:
- In the GCP Console, navigate to the Dataproc page.
- Create a Cluster:
- Click on Create Cluster.
- Configure Cluster Settings:
    - Set Cluster Type to "Standard".
    - Choose appropriate machine types for the master and worker nodes based on your data size and computational needs.
    - Enable the Component Gateway for easier job monitoring and debugging.
    - Ensure Spark Compatibility



Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.

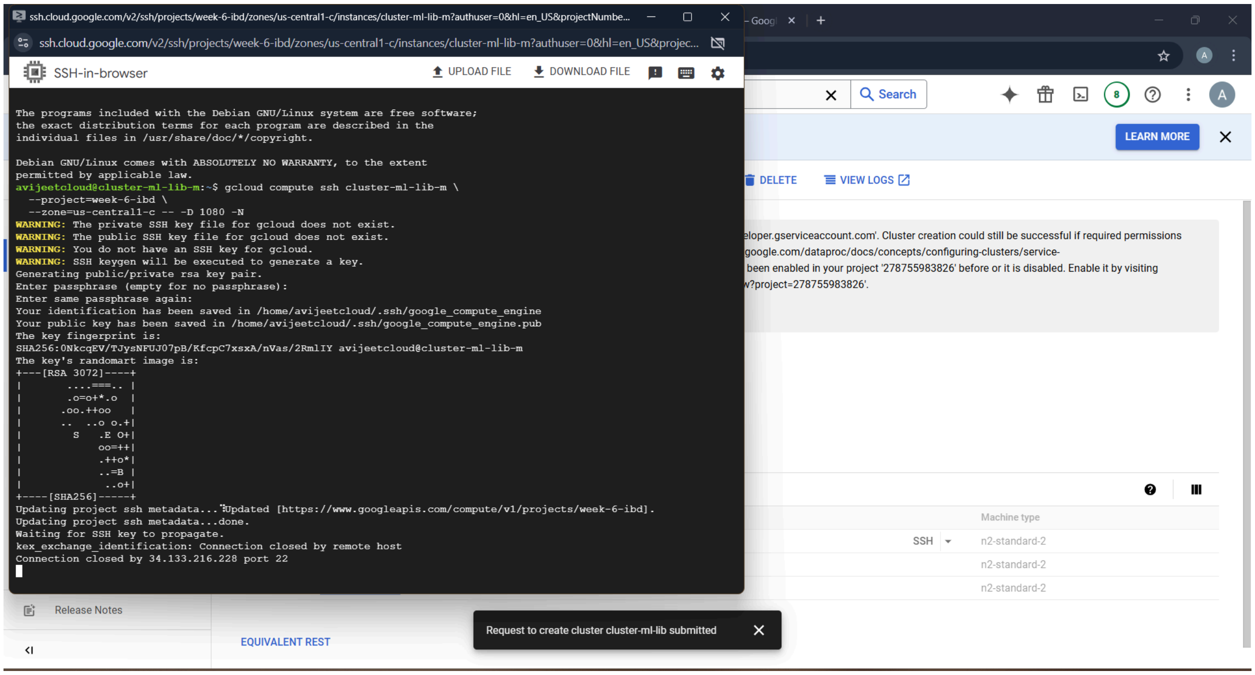
Sorry, the server was not able to fulfill your request.

□	Name ↑	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion
□	cluster-ml-lib	Provisioning	us-central1	us-central1-c	2	No	Off
□	spark-cluster	Stopped	us-central1	us-central1-b	0	No	Off
□	spark-cluster-2	Stopped	us-central1	us-central1-b	0	No	Off

No clusters selected

Please select at least one resource.

Request to create cluster cluster-ml-lib submitted


ssh.cloud.google.com/v2/ssh/projects/week-6-ibd/zones/us-central1-c/instances/cluster-ml-lib-m?authuser=0&hl=en_US&projectNumbe...
ssh.cloud.google.com/v2/ssh/projects/week-6-ibd/zones/us-central1-c/instances/cluster-ml-lib-m?authuser=0&hl=en_US&projec...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Search
LEARN MORE
DELETE
VIEW LOGS
Machine type
SSH
Release Notes
EQUIVALENT REST

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
avijeetcloud@cluster-ml-lib-m:~$ gcloud compute ssh cluster-ml-lib-m \
  --project=week-6-ibd \
  --zone=us-central1-c -- -D 1080 -N
WARNING: The private SSH key file for gcloud does not exist.
WARNING: The public SSH key file for gcloud does not exist.
WARNING: You do not have an SSH key for gcloud.
WARNING: SSH keygen will be executed to generate a key.
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/avijeetcloud/.ssh/google_compute_engine
Your public key has been saved in /home/avijeetcloud/.ssh/google_compute_engine.pub
The key fingerprint is:
SHA256:0NkcqEV/TJysNFUJ07pB/KfcpC7xsxA/nVas/2RmlIY avijeetcloud@cluster-ml-lib-m
The key's randomart image is:
+---[RSA 3072]----+
|       ....===.. |
|       .o=o+*.o  |
|      .oo.++oo   |
|      ..  ..o o.+|
|        S   .E O+|
|            oo=++|
|            .++o*|
|            ..=B |
|              ..o+|
+----[SHA256]-----+
Updating project ssh metadata...Updated [https://www.googleapis.com/compute/v1/projects/week-6-ibd].
Updating project ssh metadata...done.
Waiting for SSH key to propagate.
kex_exchange_identification: Connection closed by remote host
Connection closed by 34.133.216.228 port 22
```

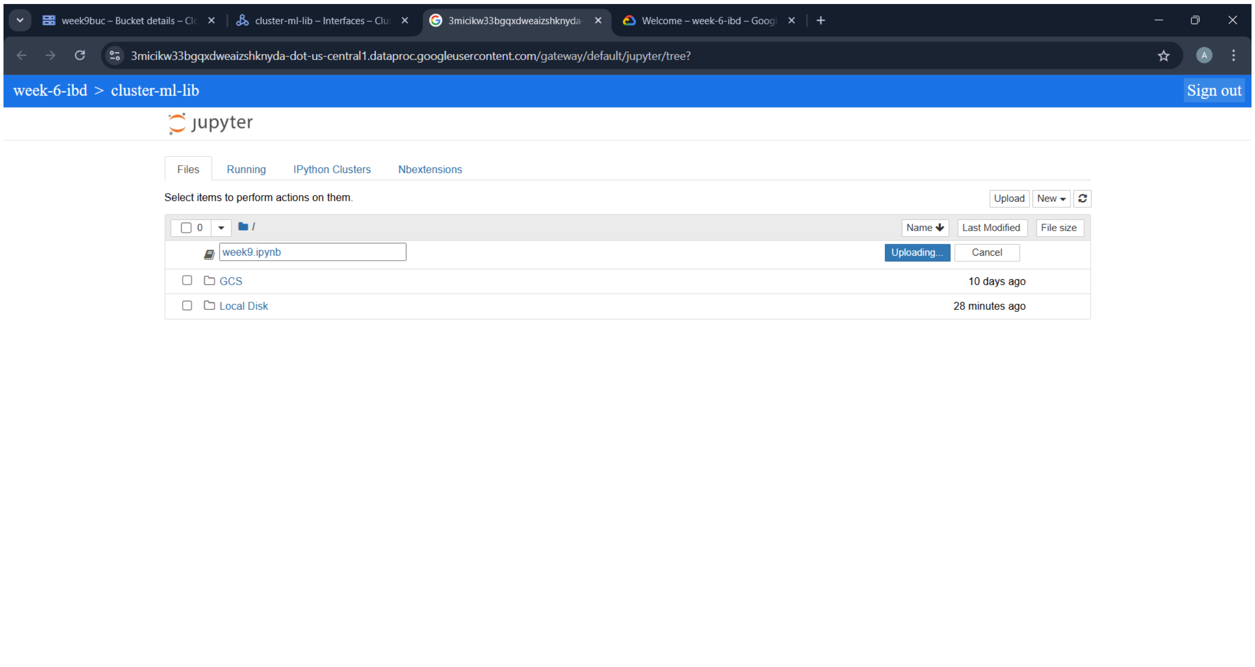
eloper.gserviceaccount.com'. Cluster creation could still be successful if required permissions
google.com/dataproc/docs/concepts/configuring-clusters/service-
been enabled in your project '278755983826' before or it is disabled. Enable it by visiting
v?project=278755983826'.

	Machine type
SSH	n2-standard-2
	n2-standard-2
	n2-standard-2

Request to create cluster cluster-ml-lib submitted

Step 4: Write and Modify the Spark Code

- Fetch Base Code:
  - Download the sample decision trees code from the Databricks example provided in the assignment.

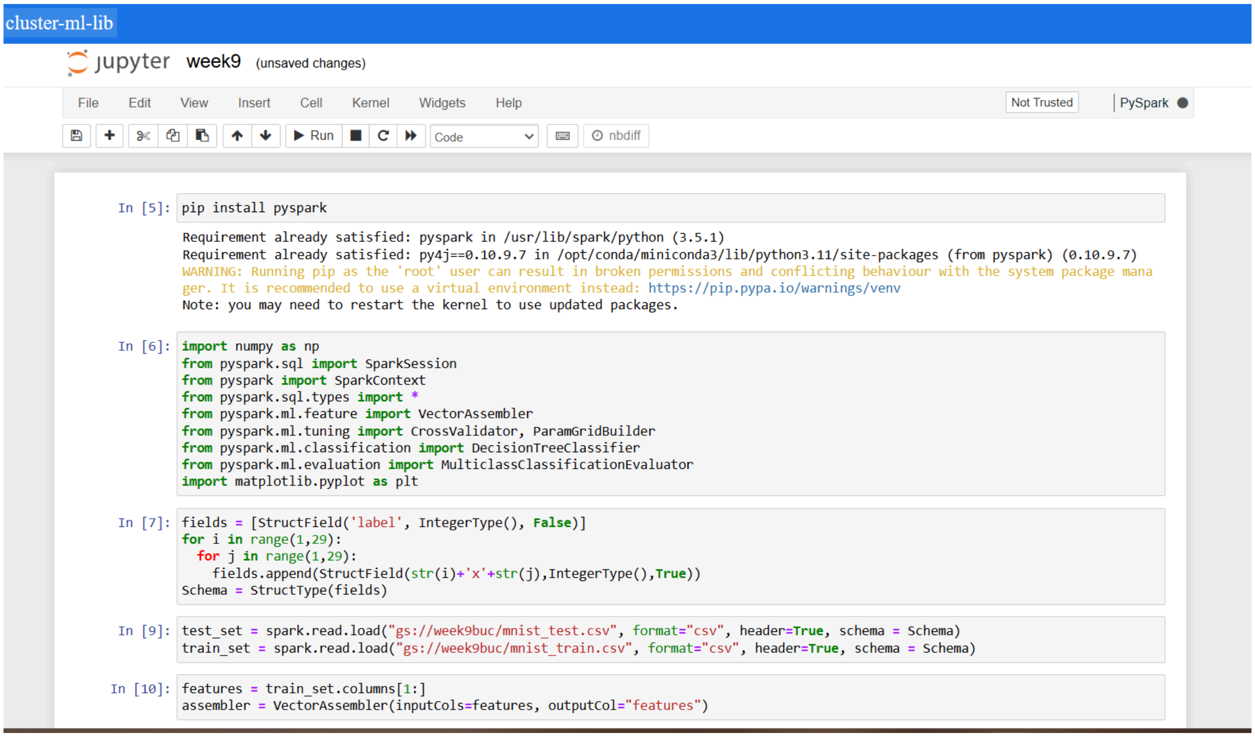

week9buc – Bucket details – Clo
cluster-ml-lib – Interfaces – Clu
3micikw33bgqxdweaizshknyda
Welcome – week-6-ibd – Googl
week-6-ibd > cluster-ml-lib
Sign out
jupyter
Files
Running
IPython Clusters
Nbextensions
Upload
New
Name
Last Modified
File size
Uploading...
Cancel

3micikw33bgqxdweaizshknyda-dot-us-central1.dataproc.googleusercontent.com/gateway/default/jupyter/tree?

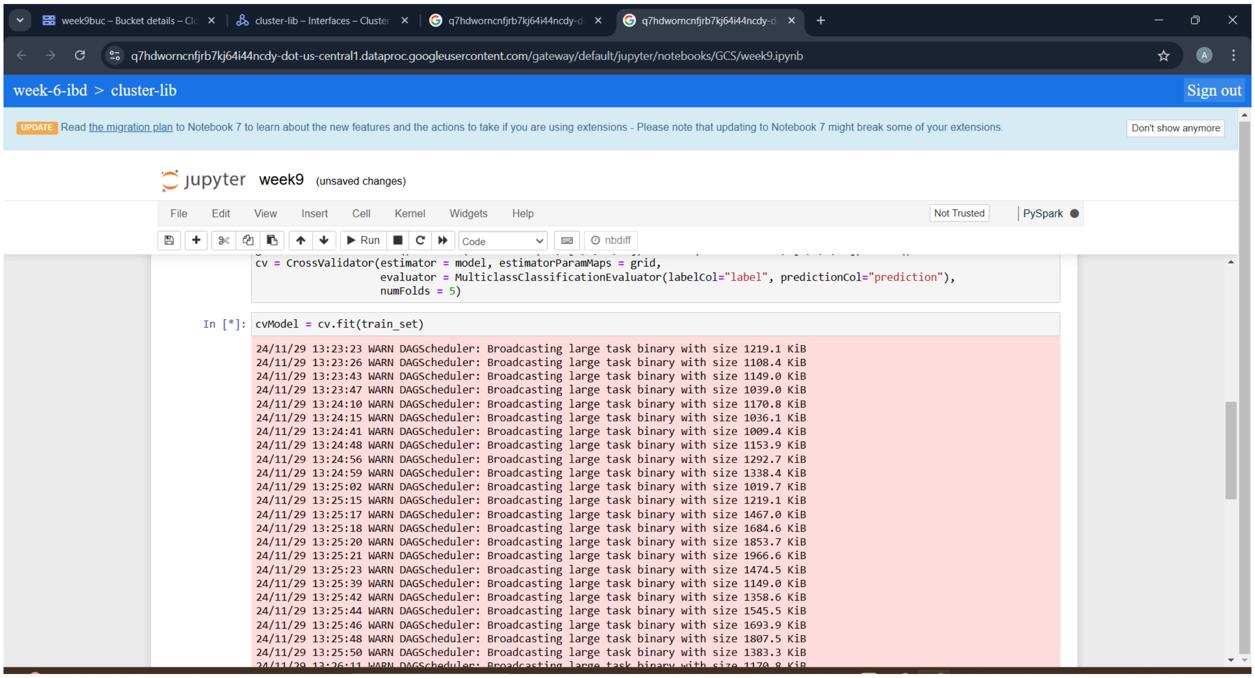
Select items to perform actions on them.

□ 0	/	
	week9.ipynb	
□	GCS	10 days ago
□	Local Disk	28 minutes ago

- Modify the Code to Use CrossValidator:
  - Use CrossValidator to tune the model parameters (e.g., maxDepth, maxBins).
  - Update Model Evaluation Metrics:
  - Adjust the code to output performance metrics for each model.



```
In [5]: pip install pyspark

        Requirement already satisfied: pyspark in /usr/lib/spark/python (3.5.1)
        Requirement already satisfied: py4j==0.10.9.7 in /opt/conda/miniconda3/lib/python3.11/site-packages (from pyspark) (0.10.9.7)
        WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package mana
        ger. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
        Note: you may need to restart the kernel to use updated packages.

In [6]: import numpy as np
        from pyspark.sql import SparkSession
        from pyspark import SparkContext
        from pyspark.sql.types import *
        from pyspark.ml.feature import VectorAssembler
        from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
        from pyspark.ml.classification import DecisionTreeClassifier
        from pyspark.ml.evaluation import MulticlassClassificationEvaluator
        import matplotlib.pyplot as plt

In [7]: fields = [StructField('label', IntegerType(), False)]
        for i in range(1,29):
          for j in range(1,29):
            fields.append(StructField(str(i)+'x'+str(j),IntegerType(),True))
        Schema = StructType(fields)

In [9]: test_set = spark.read.load("gs://week9buc/mnist_test.csv", format="csv", header=True, schema = Schema)
        train_set = spark.read.load("gs://week9buc/mnist_train.csv", format="csv", header=True, schema = Schema)

In [10]: features = train_set.columns[1:]
         assembler = VectorAssembler(inputCols=features, outputCol="features")
```


week9buc – Bucket details – Clo
cluster-lib – Interfaces – Cluster
q7hdworncnfjrb7kj64i44ncdy-d
q7hdworncnfjrb7kj64i44ncdy-d
q7hdworncnfjrb7kj64i44ncdy-dot-us-central1.dataproc.googleusercontent.com/gateway/default/jupyter/notebooks/GCS/week9.ipynb
week-6-ibd > cluster-lib
Sign out
jupyter week9 (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
PySpark
Run
Code
nbdiff
Don't show anymore

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

```
cv = CrossValidator(estimator = model, estimatorParamMaps = grid,
                    evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction"),
                    numFolds = 5)
```

```
In [*]: cvModel = cv.fit(train_set)
```

```
24/11/29 13:23:23 WARN DAGScheduler: Broadcasting large task binary with size 1219.1 KiB
24/11/29 13:23:26 WARN DAGScheduler: Broadcasting large task binary with size 1108.4 KiB
24/11/29 13:23:43 WARN DAGScheduler: Broadcasting large task binary with size 1149.0 KiB
24/11/29 13:23:47 WARN DAGScheduler: Broadcasting large task binary with size 1039.0 KiB
24/11/29 13:24:10 WARN DAGScheduler: Broadcasting large task binary with size 1170.8 KiB
24/11/29 13:24:15 WARN DAGScheduler: Broadcasting large task binary with size 1036.1 KiB
24/11/29 13:24:41 WARN DAGScheduler: Broadcasting large task binary with size 1009.4 KiB
24/11/29 13:24:48 WARN DAGScheduler: Broadcasting large task binary with size 1153.9 KiB
24/11/29 13:24:56 WARN DAGScheduler: Broadcasting large task binary with size 1292.7 KiB
24/11/29 13:24:59 WARN DAGScheduler: Broadcasting large task binary with size 1338.4 KiB
24/11/29 13:25:02 WARN DAGScheduler: Broadcasting large task binary with size 1019.7 KiB
24/11/29 13:25:15 WARN DAGScheduler: Broadcasting large task binary with size 1219.1 KiB
24/11/29 13:25:17 WARN DAGScheduler: Broadcasting large task binary with size 1467.0 KiB
24/11/29 13:25:18 WARN DAGScheduler: Broadcasting large task binary with size 1684.6 KiB
24/11/29 13:25:20 WARN DAGScheduler: Broadcasting large task binary with size 1853.7 KiB
24/11/29 13:25:21 WARN DAGScheduler: Broadcasting large task binary with size 1966.6 KiB
24/11/29 13:25:23 WARN DAGScheduler: Broadcasting large task binary with size 1474.5 KiB
24/11/29 13:25:39 WARN DAGScheduler: Broadcasting large task binary with size 1149.0 KiB
24/11/29 13:25:42 WARN DAGScheduler: Broadcasting large task binary with size 1358.6 KiB
24/11/29 13:25:44 WARN DAGScheduler: Broadcasting large task binary with size 1545.5 KiB
24/11/29 13:25:46 WARN DAGScheduler: Broadcasting large task binary with size 1693.9 KiB
24/11/29 13:25:48 WARN DAGScheduler: Broadcasting large task binary with size 1807.5 KiB
24/11/29 13:25:50 WARN DAGScheduler: Broadcasting large task binary with size 1383.3 KiB
24/11/29 13:26:11 WARN DAGScheduler: Broadcasting large task binary with size 1170.8 KiB
```

Results

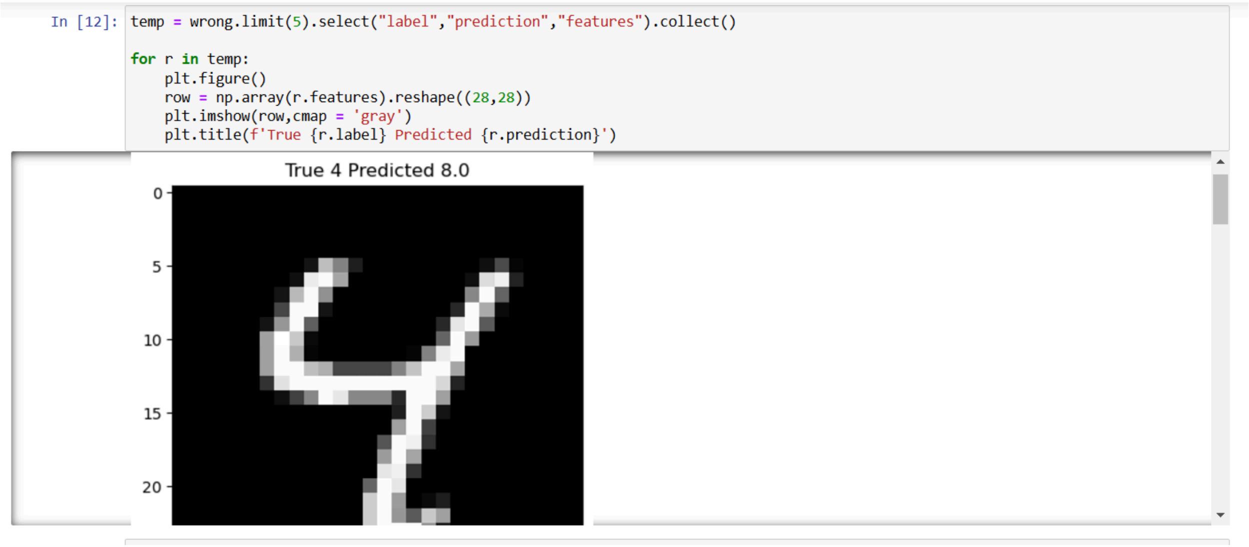
- Best Parameters after CrossValidator Autotuning:
- List the parameters that yielded the best performance (e.g., maxDepth = 10, maxBins = 32).

```
In [9]: est = cvModel.bestModel
        print(f'Best tree depth {est.getMaxDepth()}, Best bin size {est.getMaxBins()}')

        Best tree depth 16, Best bin size 2
```

```
In [10]: testresult = est.transform(test_set)
```

```
In [11]: print(f'Total test points {testresult.count()}')
         wrong = testresult.where(testresult.label!=testresult.prediction)
         print(f'Wrongly classified count {wrong.count()}')

         Total test points 9999

         24/11/29 14:15:48 WARN DAGScheduler: Broadcasting large task binary with size 1606.9 KiB
         [Stage 544:============================>                                    (1 + 1) / 2]

         Wrongly classified count 1124
```

- Misclassified Data Points:
- Analyze the misclassified data points to identify potential patterns or weaknesses in the model.



In [12]:

```
temp = wrong.limit(5).select("label","prediction","features").collect()

for r in temp:
    plt.figure()
    row = np.array(r.features).reshape((28,28))
    plt.imshow(row,cmap = 'gray')
    plt.title(f'True {r.label} Predicted {r.prediction}')
```