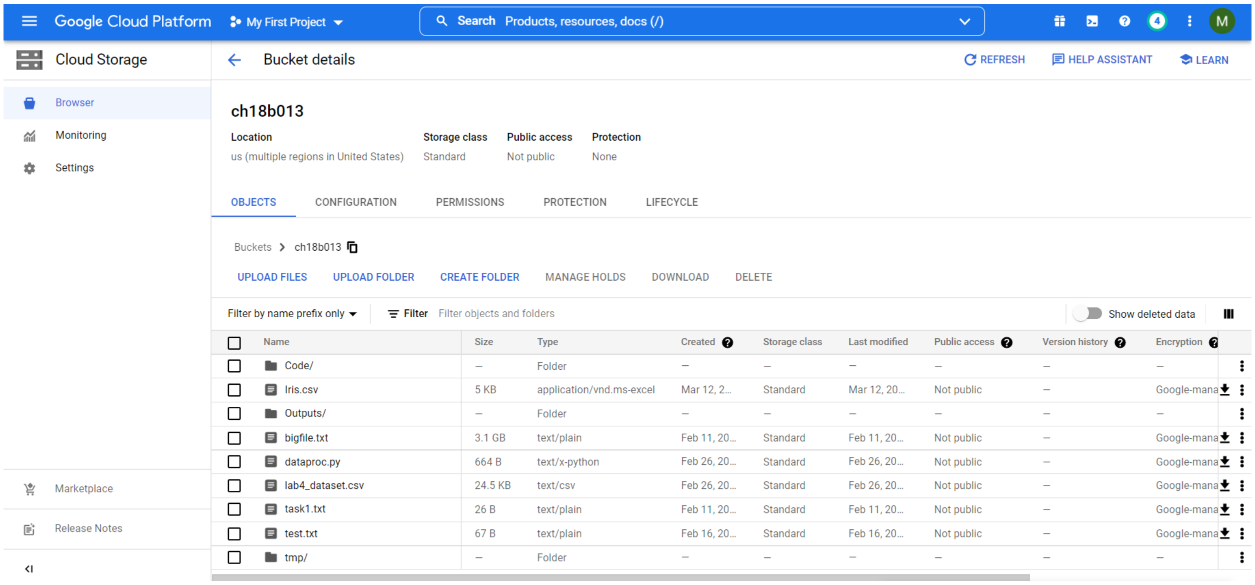
CS4830 - BIG DATA LABORATORY
LAB - 5
VISHAL RISHI MK - CH18B013

1. Download the dataset and upload it into your bucket.

The screenshots after uploading the iris dataset in the bucket and as a BigQuery table are shown below.



ch18b013

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption
□	Code/	–	Folder	–	–	–	–	–	–
□	Iris.csv	5 KB	application/vnd.ms-excel	Mar 12, 2...	Standard	Mar 12, 20...	Not public	–	Google-mana
□	Outputs/	–	Folder	–	–	–	–	–	–
□	bigfile.txt	3.1 GB	text/plain	Feb 11, 20...	Standard	Feb 11, 20...	Not public	–	Google-mana
□	dataproc.py	664 B	text/x-python	Feb 26, 20...	Standard	Feb 26, 20...	Not public	–	Google-mana
□	lab4_dataset.csv	24.5 KB	text/csv	Feb 26, 20...	Standard	Feb 26, 20...	Not public	–	Google-mana
□	task1.txt	26 B	text/plain	Feb 11, 20...	Standard	Feb 11, 20...	Not public	–	Google-mana
□	test.txt	67 B	text/plain	Feb 16, 20...	Standard	Feb 16, 20...	Not public	–	Google-mana
□	tmp/	–	Folder	–	–	–	–	–	–

Figure 1: Uploading the iris dataset in the bucket

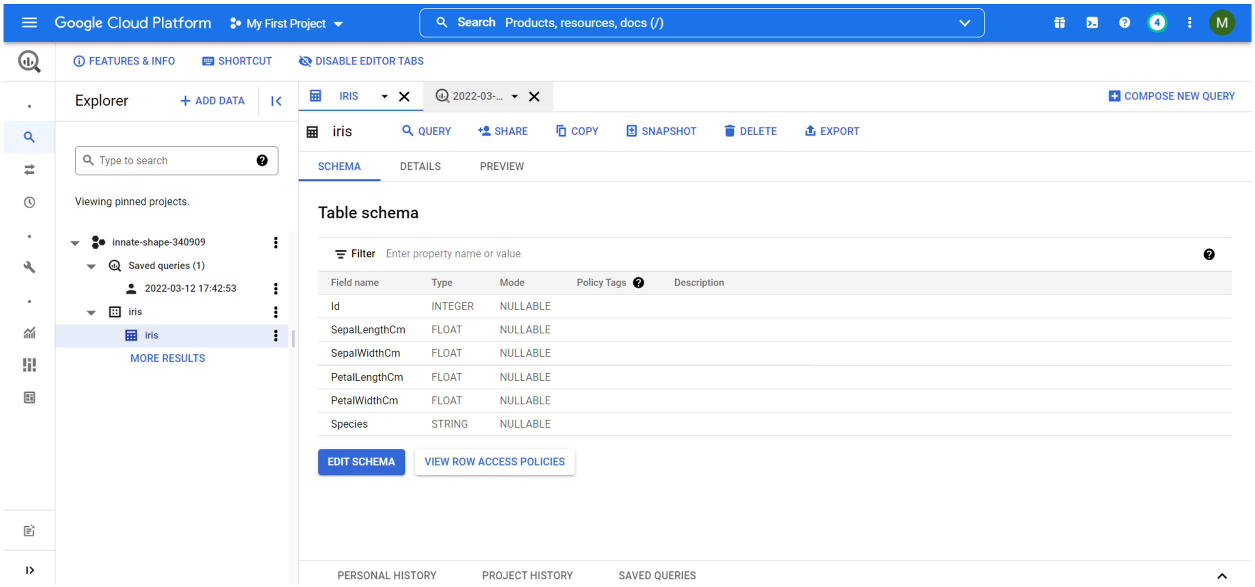
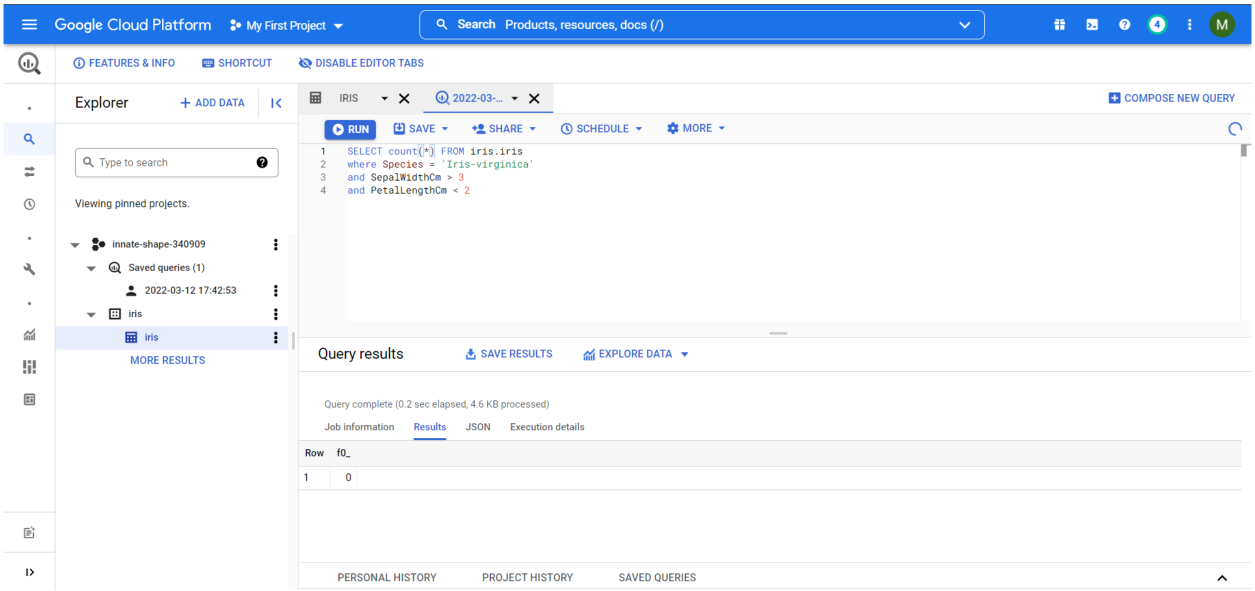


Table schema

Field name	Type	Mode	Policy Tags	Description
Id	INTEGER	NULLABLE		
SepalLengthCm	FLOAT	NULLABLE		
SepalWidthCm	FLOAT	NULLABLE		
PetalLengthCm	FLOAT	NULLABLE		
PetalWidthCm	FLOAT	NULLABLE		
Species	STRING	NULLABLE		

Figure 2: Uploading the iris dataset as a BigQuery table

2. Count the number of Iris Virginica flowers that have sepal width greater than 3 cm and petal length smaller than 2 cm. (use BigQuery)
The screenshot below shows the SQL query and the query result.



```
SELECT count(*) FROM iris.iris
where Species = 'Iris-virginica'
and SepalWidthCm > 3
and PetalLengthCm < 2
```

Query complete (0.2 sec elapsed, 4.6 KB processed)

Row	f0_
1	0

Figure 3: SQL query and the result (count: 0)

Hence, using BigQuery on the Iris dataset, we get that there are no Iris Virginica flowers that have sepal width greater than 3 cm and petal length smaller than 2 cm. **The answer is zero.**

3. Train an ML model on the dataset to predict the class of the iris plant and report the accuracy for different preprocessing techniques and models. Provide the details of data exploration and feature engineering steps.
Here, we train the dataset to find the species of the flowers using the following steps.

Step 1: We load the data from the BigQuery Table.
Step 2: We separate the data into features (all columns except 'Species') and labels (the column 'Species').
Step 3: We create different pipelines for feature engineering and learning. The details are given below.
Step 4: We split the data into train and test sets.
Step 5: We train different pipelines on the training dataset.
Step 6: We evaluate the pipelines on the test set.

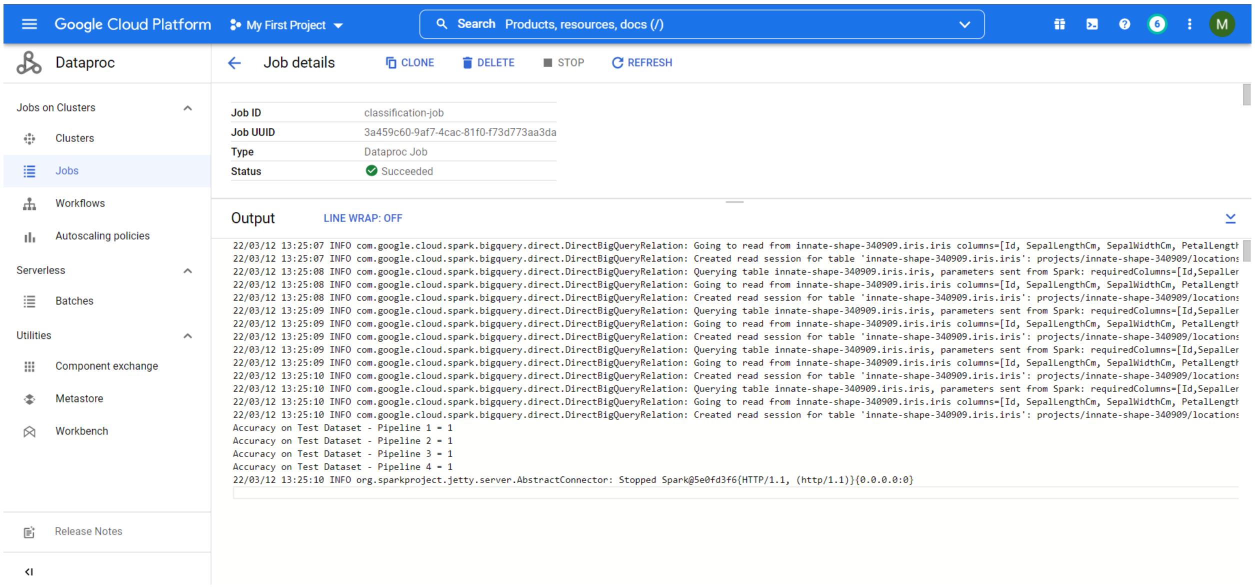
Pipeline 1:
We normalize the features using Standard Scaler. The normalized features are transformed into three-dimensional vectors using PCA. The transformed features are then fed into a random forest classifier to learn the patterns in the data.

Pipeline 2:
We normalize the features using Standard Scaler. The features are then fed into a random forest classifier to learn the patterns in the data.

Pipeline 3:
We normalize the features using Standard Scaler. The normalized features are transformed into two-dimensional vectors using PCA. The transformed features are then fed into a random forest classifier to learn the patterns in the data.

Pipeline 4:
We normalize the features using Standard Scaler. The normalized features are transformed into one-dimensional vectors using PCA. The transformed feature is then fed into a random forest classifier to learn the patterns in the data.

Given below is the output of the job in data proc:



Job ID classification-job
Job UUID 3a459c60-9af7-4cac-81f0-f73d773aa3da
Type Dataproc Job
Status ✓ Succeeded

Output

```
22/03/12 13:25:07 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from innate-shape-340909.iris.iris columns=[Id, SepalLengthCm, SepalWidthCm, PetalLength
22/03/12 13:25:07 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'innate-shape-340909.iris.iris': projects/innate-shape-340909/locations
22/03/12 13:25:08 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table innate-shape-340909.iris.iris, parameters sent from Spark: requiredColumns=[Id,SepalLen
22/03/12 13:25:08 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from innate-shape-340909.iris.iris columns=[Id, SepalLengthCm, SepalWidthCm, PetalLength
22/03/12 13:25:08 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'innate-shape-340909.iris.iris': projects/innate-shape-340909/locations
22/03/12 13:25:09 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table innate-shape-340909.iris.iris, parameters sent from Spark: requiredColumns=[Id,SepalLen
22/03/12 13:25:09 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from innate-shape-340909.iris.iris columns=[Id, SepalLengthCm, SepalWidthCm, PetalLength
22/03/12 13:25:09 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'innate-shape-340909.iris.iris': projects/innate-shape-340909/locations
22/03/12 13:25:09 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table innate-shape-340909.iris.iris, parameters sent from Spark: requiredColumns=[Id,SepalLen
22/03/12 13:25:09 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from innate-shape-340909.iris.iris columns=[Id, SepalLengthCm, SepalWidthCm, PetalLength
22/03/12 13:25:10 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'innate-shape-340909.iris.iris': projects/innate-shape-340909/locations
22/03/12 13:25:10 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table innate-shape-340909.iris.iris, parameters sent from Spark: requiredColumns=[Id,SepalLen
22/03/12 13:25:10 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from innate-shape-340909.iris.iris columns=[Id, SepalLengthCm, SepalWidthCm, PetalLength
22/03/12 13:25:10 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'innate-shape-340909.iris.iris': projects/innate-shape-340909/locations
Accuracy on Test Dataset - Pipeline 1 = 1
Accuracy on Test Dataset - Pipeline 2 = 1
Accuracy on Test Dataset - Pipeline 3 = 1
Accuracy on Test Dataset - Pipeline 4 = 1
22/03/12 13:25:10 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@5e0fd3f6{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

Figure 4: Output of the job in data proc

Accuracy on the test dataset - Pipeline 1: 1.0
Accuracy on the test dataset - Pipeline 2: 1.0
Accuracy on the test dataset - Pipeline 3: 1.0
Accuracy on the test dataset - Pipeline 4: 1.0