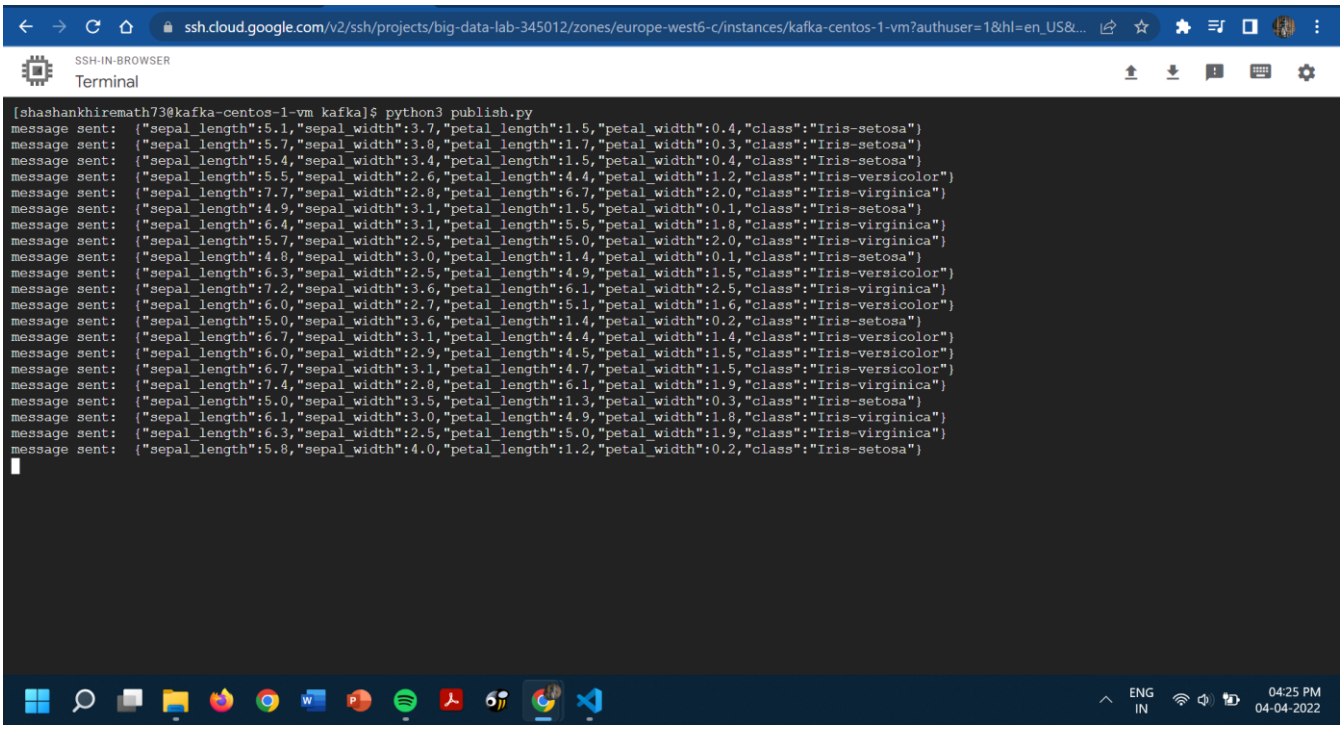
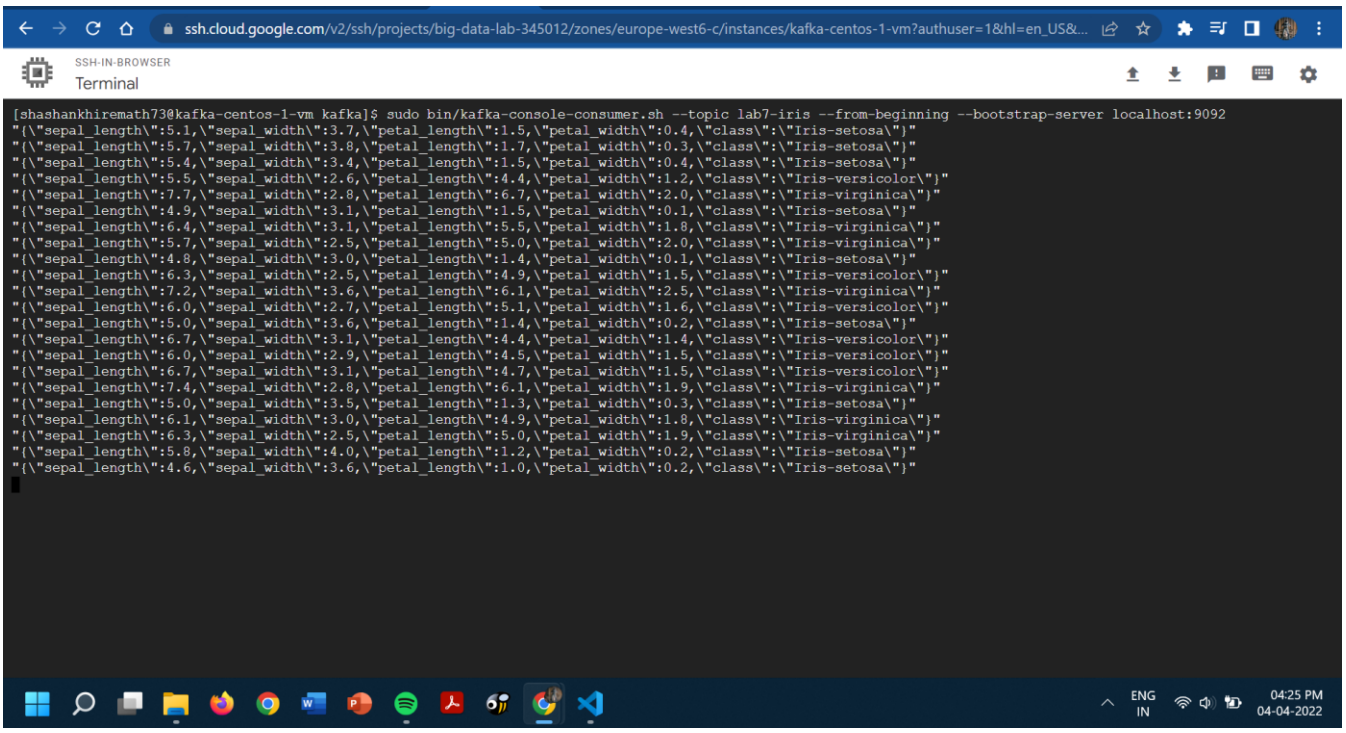
CS4830 Big Data Lab Assignment 5

Shashank H S BE18B006

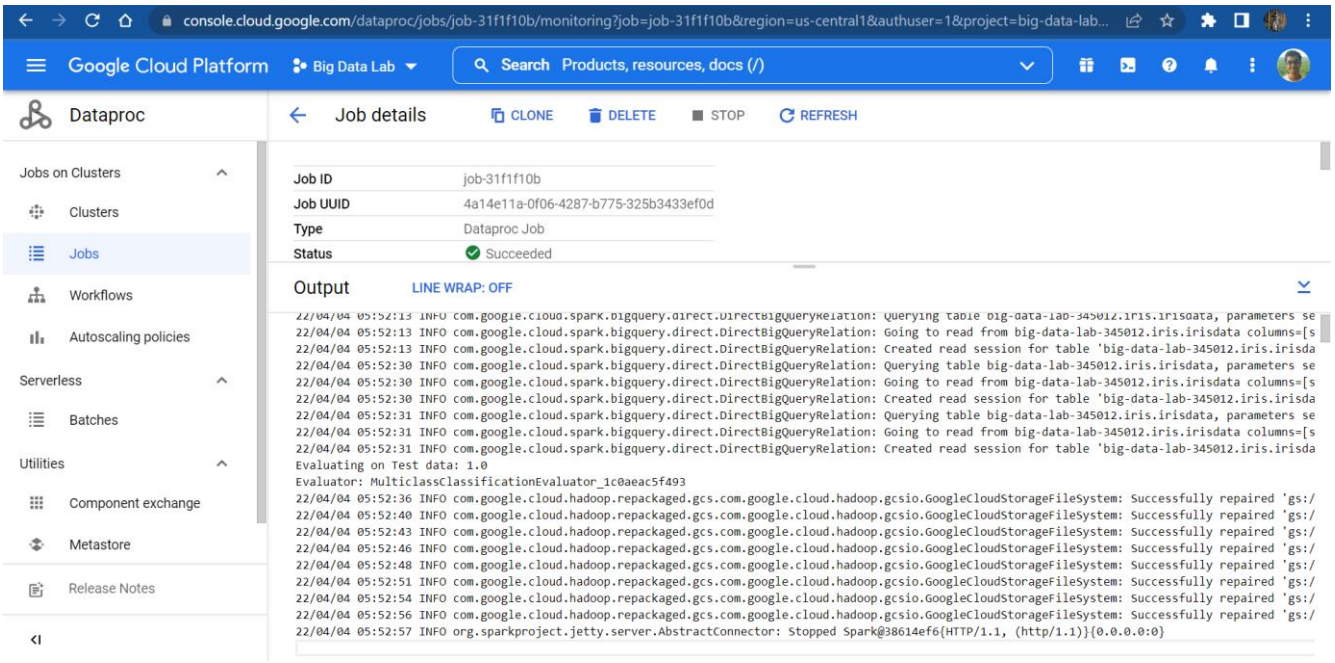
1. The screenshots show that publish.py is working as required.





publish.py reads each row in iris.csv as a dataframe and dumps it using JSON. Then it encodes each row and publishes it to the topic irispred. This is decoded back in subscribe.py and converted from json to dataframe following the appropriate schema.

2. The random forest classifier trained on the iris dataset with 100 trees was saved in the bucket. The iris2.py file is in the zipped folder.



The topic was created on the kafka server and tested. Next, the subscribe.py file was created which gives real-time predictions for each row in the iris.csv file.

The pictures are attached below.

```
|      class|indexedLabel|prediction|predictedLabel|
+-----------+------------+----------+--------------+
|Iris-setosa|         0.0|       0.0|   Iris-setosa|
|Iris-setosa|         0.0|       0.0|   Iris-setosa|
+-----------+------------+----------+--------------+

Accuracy for batch 3:  100.0
-----------------------
+---------------+------------+----------+---------------+
|          class|indexedLabel|prediction| predictedLabel|
+---------------+------------+----------+---------------+
|    Iris-setosa|         0.0|       0.0|    Iris-setosa|
|    Iris-setosa|         0.0|       0.0|    Iris-setosa|
|Iris-versicolor|         1.0|       1.0|Iris-versicolor|
+---------------+------------+----------+---------------+

Accuracy for batch 4:  100.0
------------------------
+---------------+------------+----------+---------------+
|          class|indexedLabel|prediction| predictedLabel|
+---------------+------------+----------+---------------+
|    Iris-setosa|         0.0|       0.0|    Iris-setosa|
|    Iris-setosa|         0.0|       0.0|    Iris-setosa|
|Iris-versicolor|         1.0|       1.0|Iris-versicolor|
|    Iris-setosa|         0.0|       0.0|    Iris-setosa|
+---------------+------------+----------+---------------+

Accuracy for batch 5:  100.0
------------------------
```

```
[shashankhiremath73@kafka-centos-1-vm kafka]$ python3 sub.py
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/04 19:21:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/04/04 19:21:54 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
/usr/local/lib/python3.6/site-packages/pyspark/context.py:238: FutureWarning: Python 3.6 support is deprecated in Spark 3.2.
  FutureWarning
Done loading the model.
+-----------+------------+----------+--------------+
|      class|indexedLabel|prediction|predictedLabel|
+-----------+------------+----------+--------------+
|Iris-setosa|         0.0|       0.0|   Iris-setosa|
+-----------+------------+----------+--------------+

Accuracy for batch 2:  100.0
------------------------
+-----------+------------+----------+--------------+
|      class|indexedLabel|prediction|predictedLabel|
+-----------+------------+----------+--------------+
|Iris-setosa|         0.0|       0.0|   Iris-setosa|
|Iris-setosa|         0.0|       0.0|   Iris-setosa|
+-----------+------------+----------+--------------+

Accuracy for batch 3:  100.0
------------------------
```

The RF classifier with 100 trees has not misclassified any data points (after several batches, picture not shown).