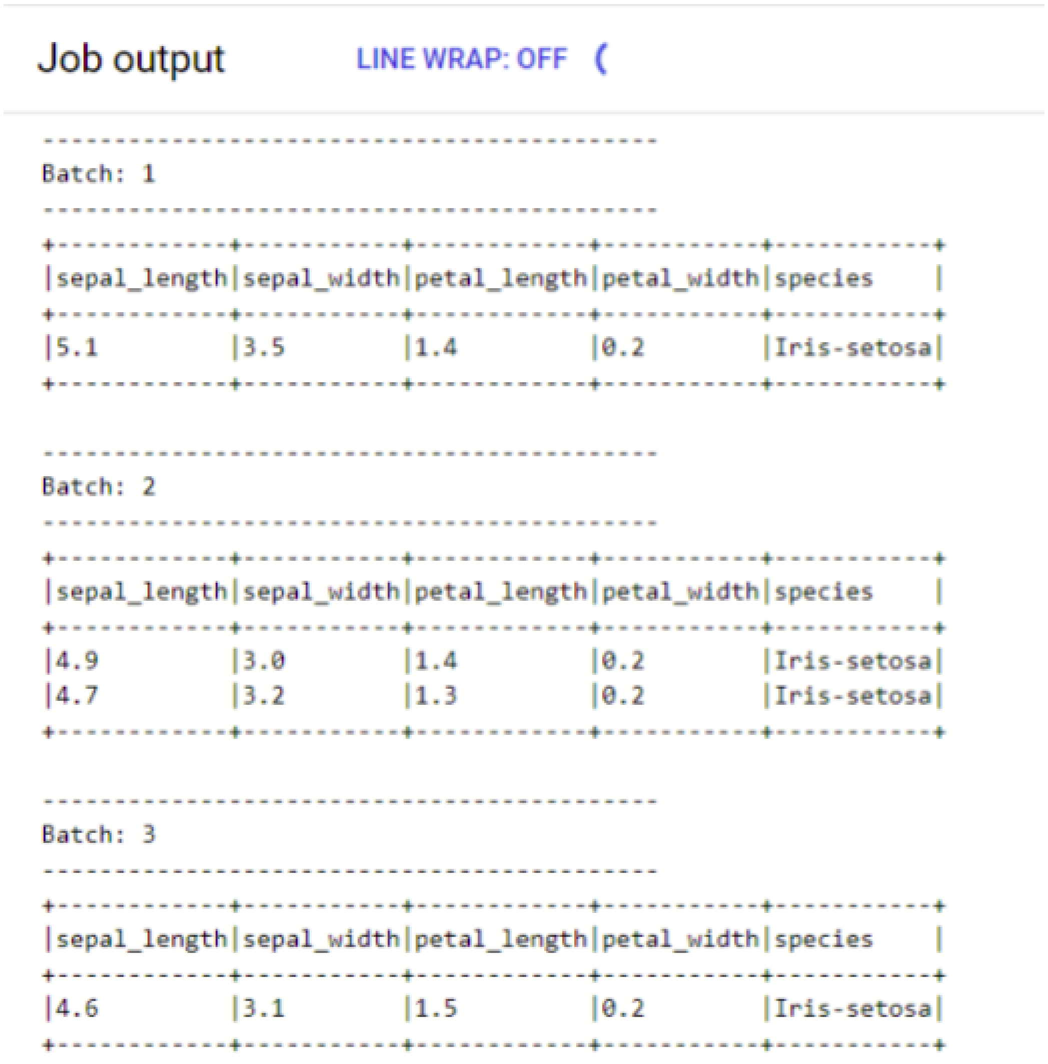
CS4830: Big Data Lab Assignment-7
Shania Mitra CH18B067

Q1. 1. Write a producer.py file that reads the iris.csv line by line and writes each row into a particular topic in Kafka.

Each iris.csv file is read as a dataframe by producer.py, which then converts it to JSON using the same format as iris.csv. Then it encodes each row before publishing it to the irispred topic. In subscriber.py, this is decoded and translated from json to dataframe using the same structure as in producer.py. Working producer and subscriber screenshot:



```
-------------------------------------------
Batch: 1
-------------------------------------------
+------------+-----------+------------+-----------+-----------+
|sepal_length|sepal_width|petal_length|petal_width|species    |
+------------+-----------+------------+-----------+-----------+
|5.1         |3.5        |1.4         |0.2        |Iris-setosa|
+------------+-----------+------------+-----------+-----------+

-------------------------------------------
Batch: 2
-------------------------------------------
+------------+-----------+------------+-----------+-----------+
|sepal_length|sepal_width|petal_length|petal_width|species    |
+------------+-----------+------------+-----------+-----------+
|4.9         |3.0        |1.4         |0.2        |Iris-setosa|
|4.7         |3.2        |1.3         |0.2        |Iris-setosa|
+------------+-----------+------------+-----------+-----------+

-------------------------------------------
Batch: 3
-------------------------------------------
+------------+-----------+------------+-----------+-----------+
|sepal_length|sepal_width|petal_length|petal_width|species    |
+------------+-----------+------------+-----------+-----------+
|4.6         |3.1        |1.5         |0.2        |Iris-setosa|
+------------+-----------+------------+-----------+-----------+
```

The row / batch of rows (depending on processing time) is printed on the console at each instant. Upon receiving rows from the subscriber, we go on to making real-time forecasts.

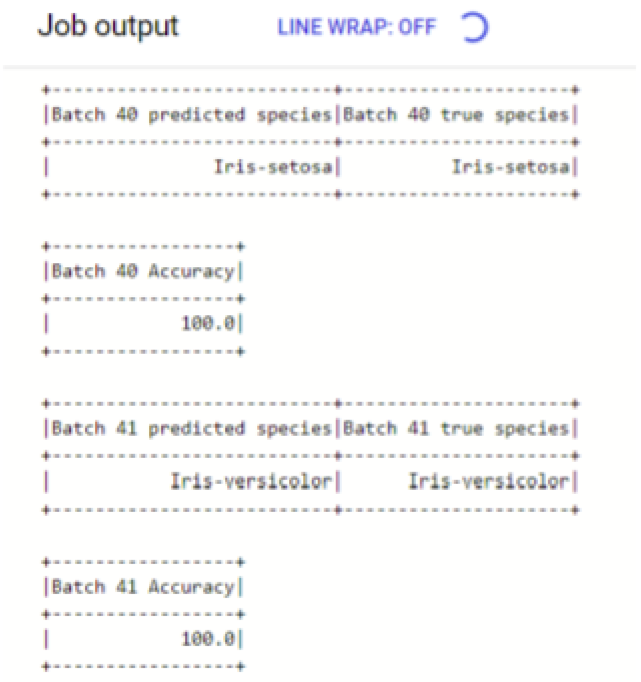
Q2. Write a subscriber.py file that uses spark streaming (can be receiver-based, dstream or structured) for producing real-time predictions on these rows by utilising

the model trained in lab5 and calculates the accuracy (the real-time predictions, true labels and accuracy all should get printed on console).

We must save the pipeline in order to create real-time forecasts. The pipeline is saved with the.save() function and loaded with the.load() method for real-time predictions. The following pipeline was used:

string to index → vector assembler → minmax scaling → random forest → index to string

This pipeline was developed on scrambled iris data with a training set of 80% and a test set of 20%. Before passing the dataframe to the pipeline, we additionally delete the sepal width feature.
Pyspark mllib's MulticlassClassificationEvaluator is utilised to find accuracy, and the foreachBatch method is used in writeStream to apply it to each batch.
Real-time prediction with true label and accuracies (screenshot):
es:



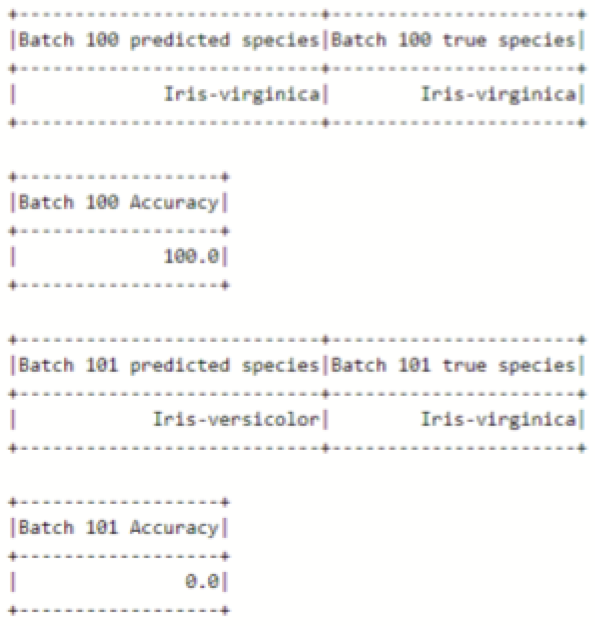
```
+--------------------------+--------------------+
|Batch 40 predicted species|Batch 40 true species|
+--------------------------+--------------------+
|               Iris-setosa|         Iris-setosa|
+--------------------------+--------------------+

+-----------------+
|Batch 40 Accuracy|
+-----------------+
|            100.0|
+-----------------+

+--------------------------+--------------------+
|Batch 41 predicted species|Batch 41 true species|
+--------------------------+--------------------+
|           Iris-versicolor|     Iris-versicolor|
+--------------------------+--------------------+

+-----------------+
|Batch 41 Accuracy|
+-----------------+
|            100.0|
+-----------------+
```

Example of misclassification:



```
+---------------------------+----------------------+
|Batch 100 predicted species|Batch 100 true species|
+---------------------------+----------------------+
|             Iris-virginica|        Iris-virginica|
+---------------------------+----------------------+

+------------------+
|Batch 100 Accuracy|
+------------------+
|             100.0|
+------------------+

+---------------------------+----------------------+
|Batch 101 predicted species|Batch 101 true species|
+---------------------------+----------------------+
|            Iris-versicolor|        Iris-virginica|
+---------------------------+----------------------+

+------------------+
|Batch 101 Accuracy|
+------------------+
|               0.0|
+------------------+
```