Lab 5 - Assignment

MM16B023[1]

[a]*Indian Institute of Technology Madras*

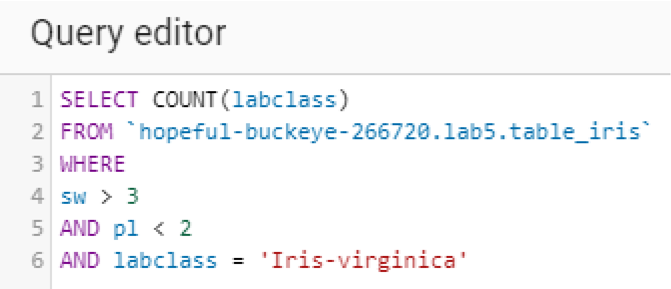
Keyword: BigQuery, StandardScaler, RandomForest, XGBoost

Abstract: This paper presents the solutions to third assignment of the Big Data Laboratory course (CS4830) at *IIT Madras.* All the notations used are as according with the textbook Mining of massive data sets by Anand Rajaraman

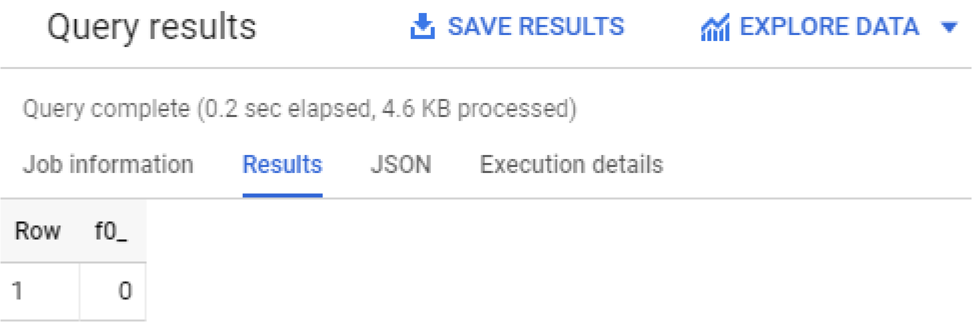
Problem 1

Count using BigQuery the number of Iris Virginica flowers which have sepal width greater than 3 cm and petal length smaller than 2 cm

Solution:



```
SELECT COUNT(labclass)
FROM `hopeful-buckeye-266720.lab5.table_iris`
WHERE
sw > 3
AND pl < 2
AND labclass = 'Iris-virginica'
```



Query complete (0.2 sec elapsed, 4.6 KB processed)

Row	f0_
1	0

⟹ There are no flowers with sepal width greater than 3 cm and petal length smaller than 2 cm

Problem 2

Train a classification model on the dataset and report the accuracy for different preprocessing techniques and models. Provide the details of data exploration and feature engineering steps

Solution:

```
### Written by H.Vishal MM16B023 at 1:43 PM on 29th Feb, 2020 ###

from __future__ import print_function
from pyspark.context import SparkContext
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression
from pyspark.sql.session import SparkSession
from pyspark.ml import Pipeline
from pyspark.ml.feature import PCA
from pyspark.ml.linalg import Vectors
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.util import MLUtils
import numpy as np
from pyspark.ml.feature import StandardScaler
import pyspark.sql.functions as f
import pyspark.sql.types
from pyspark.sql import Row
from pyspark.sql.types import DoubleType
from pyspark.ml import Pipeline
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.feature import StringIndexer, VectorIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.feature import IndexToString, StringIndexer, VectorIndexer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.feature import PCA

sc = SparkContext()
spark = SparkSession(sc)

## Reading Dataframe

spark_df = spark.read.format("bigquery").option("table", "lab5.table_iris").load().toDF("sl", "sw", "pl
clean_data = spark_df.withColumn("label", spark_df["labclass"])

cols = spark_df.drop('labclass').columns

assembler = VectorAssembler(inputCols=cols, outputCol = 'features')
labelIndexer = StringIndexer(inputCol="labclass", outputCol="indexedLabel").fit(spark_df)

## Standardize the columns

scaler = StandardScaler(inputCol="features", outputCol="scaledFeatures", withStd=False, withMean=True)

## Principal component analysis

pca = PCA(k=3, inputCol='scaledFeatures', outputCol='pcaFeature')

(trainingData, testData) = spark_df.randomSplit([0.8, 0.2])

## Training a RandomForest model

rf = RandomForestClassifier(labelCol="indexedLabel", featuresCol="pcaFeature", numTrees=10)

## Retrieve orginal labels from indexed labels

labelConverter = IndexToString(inputCol="prediction", outputCol="predictedLabel",
                               labels=labelIndexer.labels)

## Modying indexers and forest in a Pipeline
```

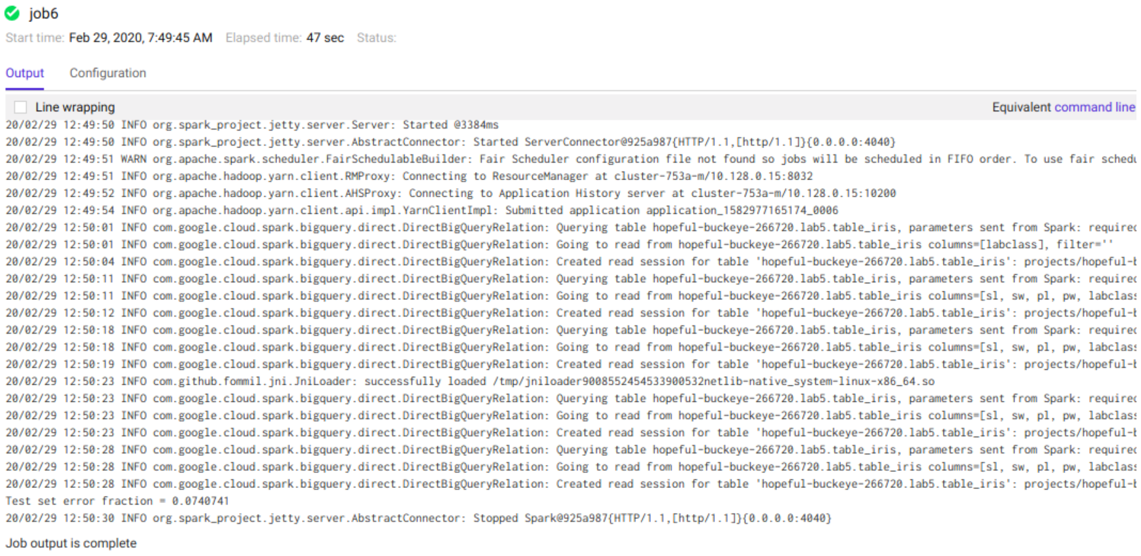
```
pipeline = Pipeline(stages=[labelIndexer, assembler,scaler,pca, rf, labelConverter])

## Train the ML model

model = pipeline.fit(trainingData)

## Predictions

predictions = model.transform(testData)
evaluator = MulticlassClassificationEvaluator(
    labelCol="indexedLabel", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set error fraction = %g" % (1.0 - accuracy))
```

Output

Model	No. of comp	Test set error
Standardization & RandomForest	2	16.129 %
	3	7.407 %
Normalization & xgboost	2	12.424 %
	3	4.975%

Discussion

- As the number of principal components included increases, better the fit is which indeed results in better performance on the given test data set.
- Ideally, one would keep increasing the number of principal components until the error converges. This would give the optimal number of the components to be included.
- Comparison is made for the above 2 different techniques preprocessing techniques and models. In this case, the 2^{nd} choice seems to perform better. In general, one needs try out at least a few different combinations as there in no set rule as to which is going to perform better for a particular problem.



job6

Start time: Feb 29, 2020, 7:49:45 AM Elapsed time: 47 sec Status:

```
20/02/29 12:49:50 INFO org.spark_project.jetty.server.Server: Started @3384ms
20/02/29 12:49:50 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@925a987{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
20/02/29 12:49:51 WARN org.apache.spark.scheduler.FairSchedulableBuilder: Fair Scheduler configuration file not found so jobs will be scheduled in FIFO order. To use fair schedu
20/02/29 12:49:51 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-753a-m/10.128.0.15:8032
20/02/29 12:49:52 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at cluster-753a-m/10.128.0.15:10200
20/02/29 12:49:54 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1582977165174_0006
20/02/29 12:50:01 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table hopeful-buckeye-266720.lab5.table_iris, parameters sent from Spark: required
20/02/29 12:50:01 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from hopeful-buckeye-266720.lab5.table_iris columns=[labclass], filter=''
20/02/29 12:50:04 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'hopeful-buckeye-266720.lab5.table_iris': projects/hopeful-b
20/02/29 12:50:11 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table hopeful-buckeye-266720.lab5.table_iris, parameters sent from Spark: required
20/02/29 12:50:11 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from hopeful-buckeye-266720.lab5.table_iris columns=[sl, sw, pl, pw, labclass
20/02/29 12:50:12 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'hopeful-buckeye-266720.lab5.table_iris': projects/hopeful-b
20/02/29 12:50:18 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table hopeful-buckeye-266720.lab5.table_iris, parameters sent from Spark: required
20/02/29 12:50:18 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from hopeful-buckeye-266720.lab5.table_iris columns=[sl, sw, pl, pw, labclass
20/02/29 12:50:19 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'hopeful-buckeye-266720.lab5.table_iris': projects/hopeful-b
20/02/29 12:50:23 INFO com.github.fommil.jni.JniLoader: successfully loaded /tmp/jniloader9008552454533900532netlib-native_system-linux-x86_64.so
20/02/29 12:50:23 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table hopeful-buckeye-266720.lab5.table_iris, parameters sent from Spark: required
20/02/29 12:50:23 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from hopeful-buckeye-266720.lab5.table_iris columns=[sl, sw, pl, pw, labclass
20/02/29 12:50:23 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'hopeful-buckeye-266720.lab5.table_iris': projects/hopeful-b
20/02/29 12:50:28 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Querying table hopeful-buckeye-266720.lab5.table_iris, parameters sent from Spark: required
20/02/29 12:50:28 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Going to read from hopeful-buckeye-266720.lab5.table_iris columns=[sl, sw, pl, pw, labclass
20/02/29 12:50:28 INFO com.google.cloud.spark.bigquery.direct.DirectBigQueryRelation: Created read session for table 'hopeful-buckeye-266720.lab5.table_iris': projects/hopeful-b
Test set error fraction = 0.0740741
20/02/29 12:50:30 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@925a987{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
```

Job output is complete

Fig: PCA ($k = 3$) with StandardScaler using RandomForest