Big Data Programming

## CSEE5590/490

**Module 2 Lab 2**

**Report**

Submitted By:

Sneha Mishra:

Class ID: 11

Aditya Soman

Class ID: 19

Team: 12  
Professor: Yugyung Lee

Name: Sneha Mishra  
Class ID: 11  
Email: [smccr@mail.umkc.edu](mailto:smccr@mail.umkc.edu)  
[MyGitHub](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/wiki/Module-2:-Lab-%231)

Technical Partner:  
Name: Aditya Soman  
Class ID: 19  
Email: [aditya.soman@mail.umkc.edu](mailto:aditya.soman@mail.umkc.edu)  
[GitHub](https://github.com/adityasomann/Big-Data-Programming)

YouTube Link explaining the Lab work can be found [here](https://www.youtube.com/watch?v=FodefHGkafs) .  
The report for the Lab work is [here](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/wiki/-----------) .  
The code for this lab work can be found [here](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/tree/master/Lab/Mod2-Lab%231/Source)

**Objective**

Understanding Hadoop MapReduce Algorithm and Spark Data Frames better and perform complex intuitive queries on the created database.

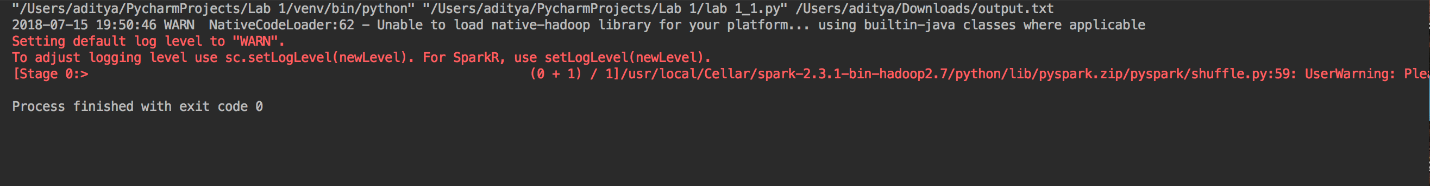
**Features**

1. Run the MapReduce job on Apache Spark.
2. Create a report including your algorithm and result screenshots to find Facebook common friends.
3. Create a Spark DataFrame using one of datasets.
4. Perform 10 intuitive queries in Database.
5. Perform any 5 queries in Spark RDD’s and Spark Data Frames.
6. Compare the results.

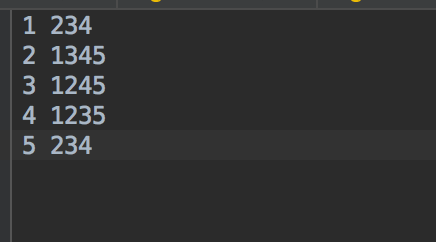
**Steps:**

**Part 1: Hadoop MapReduce Algorithm Use Case**

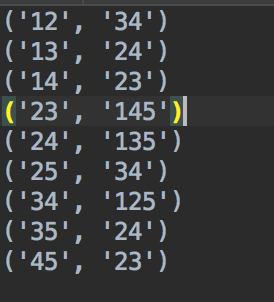
**Execution:**



**Input:**



**Output:**



**Part 2: Spark Data Frames Use Case**

**Another dataset needs to be created to perform some operations such as Join, Union etc on the given Dataset. Created dataset can be found**[**here**](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/blob/master/Lab/Mod2-Lab%231/Source/cric.csv)

**Part 2.a) Create a Spark DataFrame:**

**Query: cricket\_df = spark.read.option("header", "true").csv("/Users/snehamishra/Downloads/cric.csv")**

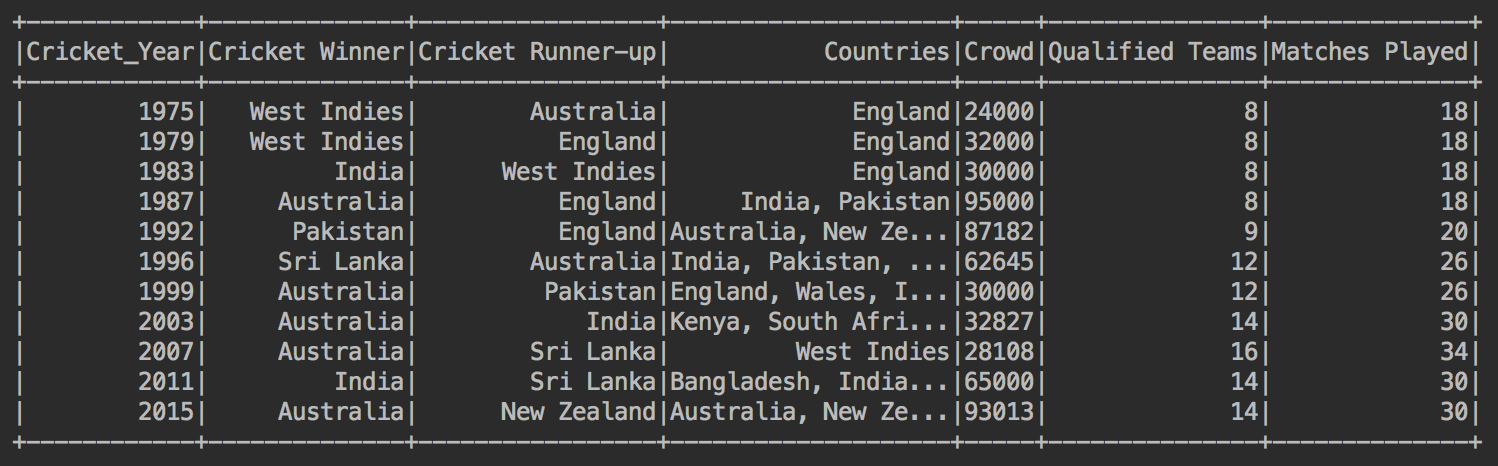
**The Cricket DataFrame can be found**[**here**](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/blob/master/Lab/Mod2-Lab%231/Source/cric.csv)

**The football DataFrame can be found**[**here**](https://github.com/SnehaMishra28/BigData_Programming_Summer2018/blob/master/Lab/Mod2-Lab%231/Source/WorldCups.csv)

**Football DataFrame:**



**Cricket DataFrame:**



**Part 2.b) Perform 10 intuitive queries:**

**Query 1: Total count of football World Cups**

**Query**

count = football\_df.count()

**Output -**

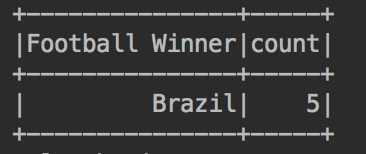
https://github.com/SnehaMishra28/BigData_Programming_Summer2018/raw/master/Lab/Mod2-Lab%231/Documents/2_2.png

**Query 2: Maximum number of world cup wins by any team**

**Query**

football\_df.groupBy(['Winner']).count().orderBy("count", ascending=False).show(1)

**Output -**



**Query 3: Most un-lucky team(country) in the history of World Cups (Runner up)**

**Query**

football\_df.groupBy(['Football Runners-Up']).count().orderBy("count", ascending=False).show()

**Output -**

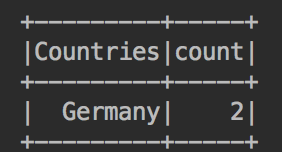


**Query 4: Most number of times a country has hosted**

**Query**

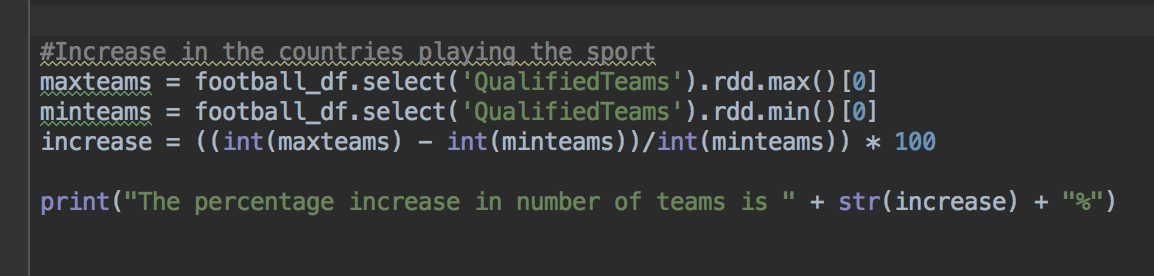
football\_df.groupBy(['Countries']).count().orderBy("count", ascending=False).show(1)

**Output -**



**Query 5: Percentage Increase in the countries playing the soccer**

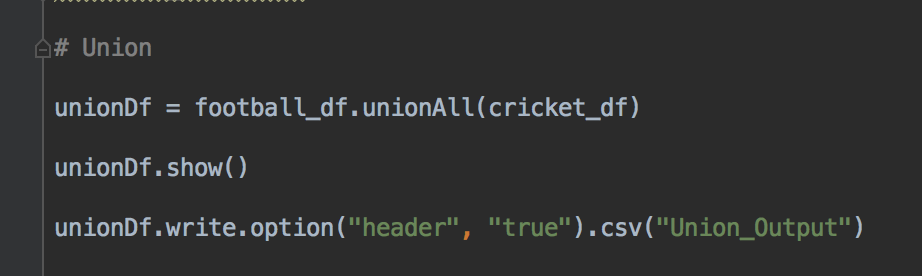
**Query -**



**Output -**

https://github.com/SnehaMishra28/BigData_Programming_Summer2018/raw/master/Lab/Mod2-Lab%231/Documents/2_3.png

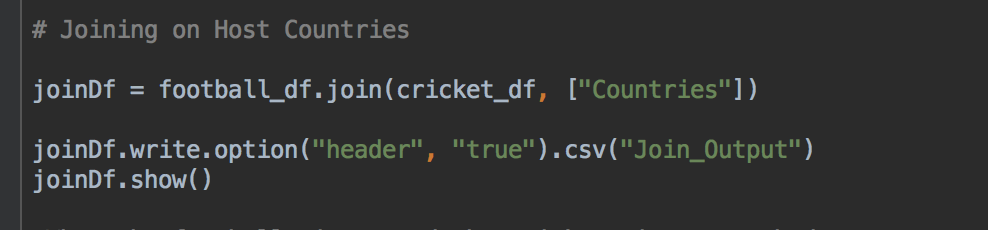
**Query 6: Union of Football Dataframe with Cricket Dataframe**

**Query -**

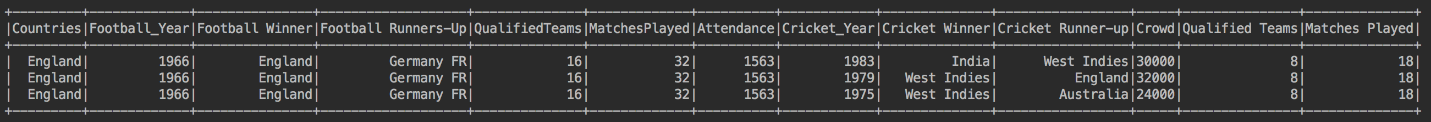
**Output –**



**Query 7: Joining of Football Dataframe with Cricket Dataframe on Host Countries**

**Query -**

**Output –**

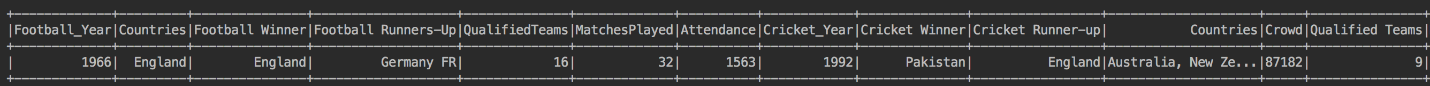


**Query 8: When the football winner and the cricket winner were same team**

**Query**

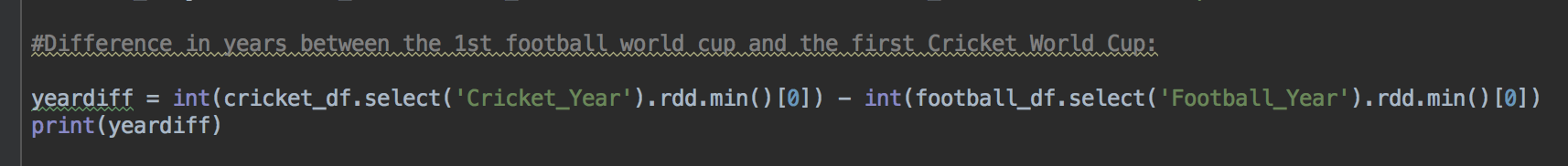
football\_df.join(cricket\_df, football\_df['Football Winner'] == cricket\_df['Cricket Runner-Up']).show(1)

**Output -**

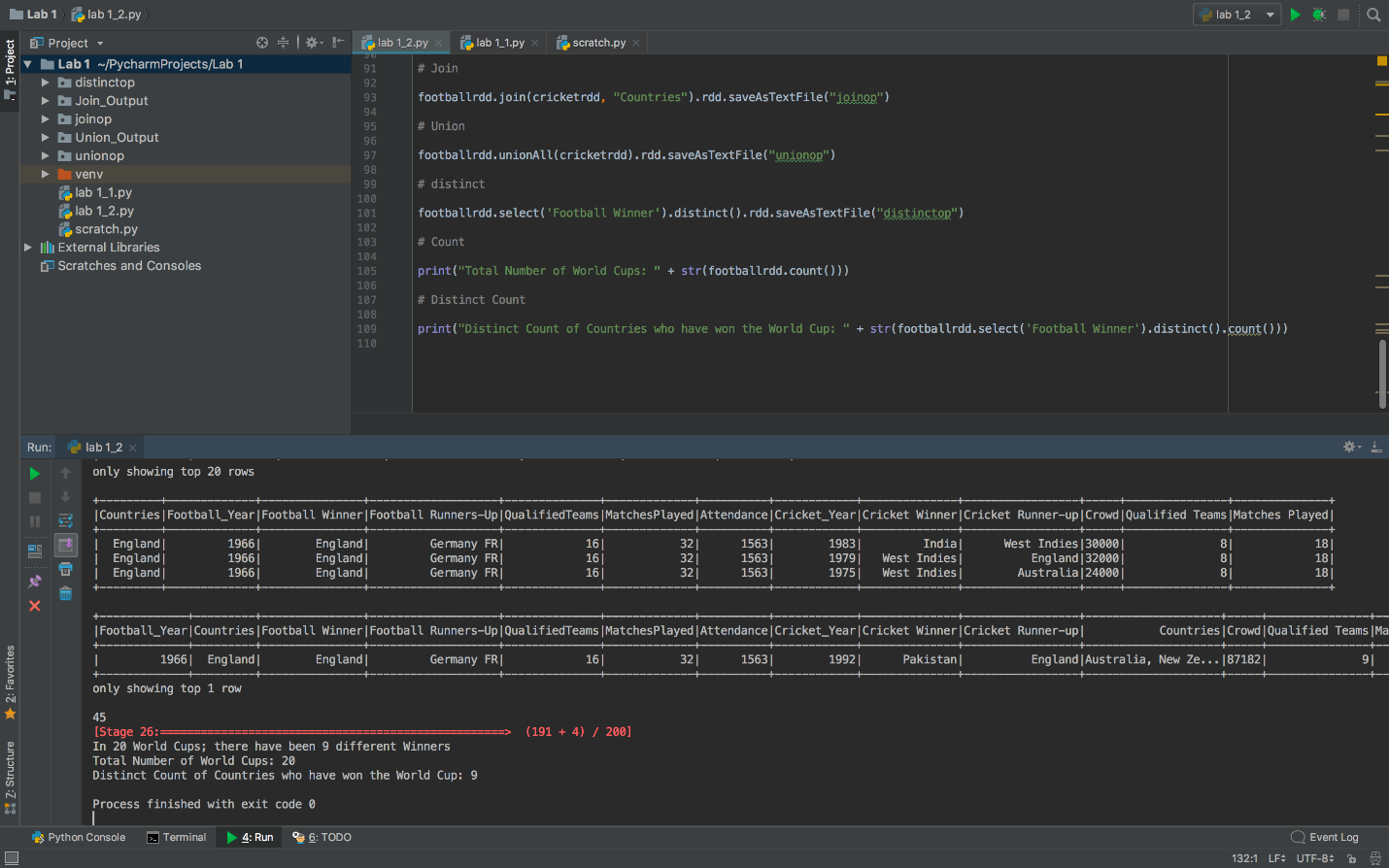


**Query 9: Difference in years between the 1st football world cup and the first Cricket World Cup**

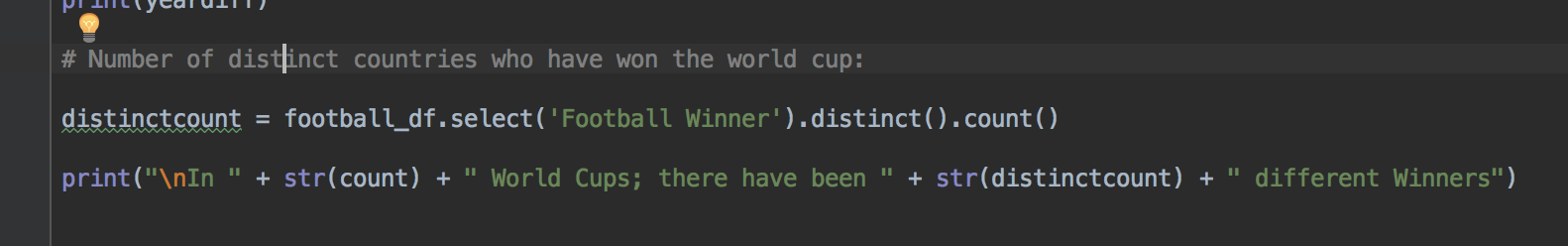
**Query –**



**Output –**



**Query 10: Number of distinct countries who have won the world cup**

**Query -**

**Output -**

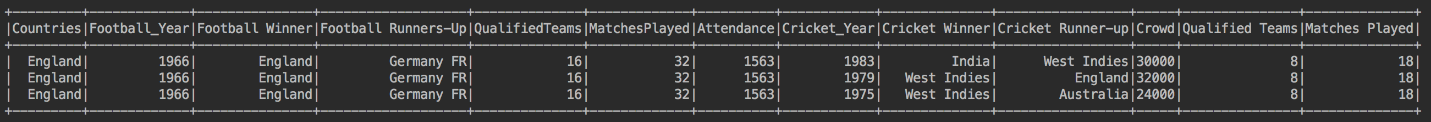
https://github.com/SnehaMishra28/BigData_Programming_Summer2018/raw/master/Lab/Mod2-Lab%231/Documents/2_12.png

**Part 2.c) Perform any 5 queries in Spark RDD’s and Spark Data Frames:**

**Query 1: Join of Football Dataframe with Cricket Dataframe on Host Countries**

**Query - footballrdd.join(cricketrdd, "Countries").rdd.saveAsTextFile("joinop")**

**Output -**



**Query 2: Union Football Dataframe with Cricket Dataframe**

**Query - footballrdd.unionAll(cricketrdd).rdd.saveAsTextFile("unionop")**

**Output -**



**Query 3: distinct countries that have won the Football World Cup**

**Query - footballrdd.unionAll(cricketrdd).rdd.saveAsTextFile("unionop")**

**Output -**

https://github.com/SnehaMishra28/BigData_Programming_Summer2018/raw/master/Lab/Mod2-Lab%231/Documents/2_12.png

**Query 4: Count of Number of World Cups**

**Query - print(footballrdd.count())**

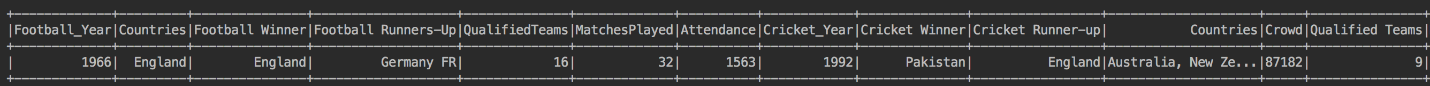
**Output -**

https://github.com/SnehaMishra28/BigData_Programming_Summer2018/raw/master/Lab/Mod2-Lab%231/Documents/2_2.png

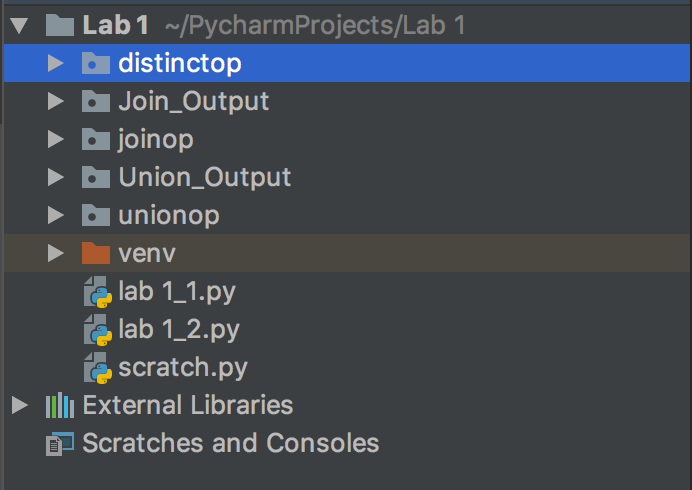
**Query 5: When the football winner and the cricket winner were same team**

**Query - print(footballrdd.select('Football Winner').distinct().count())**

**Output -**



**Output Folder:**



**Datasets:**

**FIFA World Cup Dataset:**

<https://www.kaggle.com/abecklas/fifa-world-cup#WorldCupMatches.csv>

**Kickstarter Projects Dataset:**

<https://www.kaggle.com/kemical/kickstarter-projects>

**Google-Landmarks Dataset:**

https://www.kaggle.com/google/google-landmarks-dataset

**References:**

1. <https://snap.stanford.edu/data/egonets-Facebook.html>
2. https://umkc.app.box.com/s/y6juor0fwe96f6louboy3mvbfpli6pgt