

# HOME ASSIGNMENT 4

**Roll No:** 208W1A1299

**Name:** MOHAMMAD RIZWANULLAH

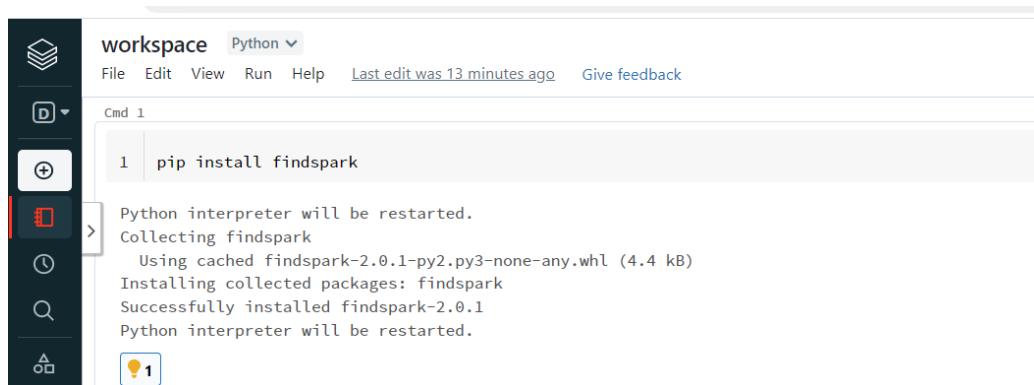
**Aim:** To do analysis using Apache spark in data bricks.

**Execution:**

- 1) Open databricks and create a new cluster with latest configuration version.
- 2) Now create a Notebook to work with spark.
- 3) Import csv file or any file for using it and do analysis using pyspark.
- 4) Now write code to import dataset and create some RDD to read input in spark.
- 5) Run the cell.

**Below are the screenshots of execution:**

Twitter followers count and protected data analysis.



The screenshot shows a Databricks workspace interface. The top bar indicates the language is Python. Below the menu bar, the command 'pip install findspark' is entered in a cell. The output shows the Python interpreter being restarted, the package 'findspark' being collected, and successfully installed. A small yellow lightbulb icon with the number 1 is visible at the bottom of the cell.

```
workspace Python
File Edit View Run Help Last edit was 13 minutes ago Give feedback

Cmd 1

1 pip install findspark

Python interpreter will be restarted.
Collecting findspark
Using cached findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
Python interpreter will be restarted.
```



The screenshot shows a Databricks workspace interface with a Python notebook. The code imports findspark and initializes it. It then reads a CSV file from the FileStore, filters for 'tags' and 'views', and processes the data. The output shows the Spark Jobs tab with the execution details.

```
workspace Python
File Edit View Run Help Last edit was 14 minutes ago Give feedback

Cmd 2

1 import findspark
2 findspark.init()
3 from pyspark.context import SparkContext
4 from pyspark.sql.session import SparkSession
5 from operator import add
6 import math
7 sc = SparkContext.getOrCreate()
8 spark = SparkSession(sc)
9 tags_views = spark.read.csv("dbfs:/FileStore/shared_uploads/ajaynagaraju32@outlook.com/USVideos-1.csv", inferSchema = True, header = True).dropna().select("tags", "views")
10 def tags_split(x):
11     tags=x["tags"].split(",")
12     result=[]
13     for every in tags:
14         if not x["views"] or every==None:
15             continue
16         result.append((every.strip(",").lower(),int(math.log(int(x["views"]))))))
17     return tuple(result)
18 rdd=tags_views.rdd.flatMap(tags_split).reduceByKey(add)
19 toptags=rdd.takeOrdered(10, key = lambda x: -x[1])
20 df=spark.createDataFrame(toptags)
21
22

▶ (3) Spark Jobs
▶ tags_views: pyspark.sql.dataframe.DataFrame = [tags: string, views: integer]
▶ dt: pyspark.sql.dataframe.DataFrame = [1: string, 2: long]
```

```
1 df.show()
```

▶ (3) Spark Jobs

```
+-----+-----+
|      _1|      _2|
+-----+-----+
|  funny|10202|
| comedy| 8911|
| [none]| 5513|
|  music| 3807|
| how to| 3783|
|   2017| 3750|
| makeup| 3635|
| trailer| 3578|
|   news| 3430|
|   humor| 3376|
+-----+-----+
```

```
1 import matplotlib.pyplot as plt
2
3 my_data = [10202, 8911, 3635, 3807, 3578, 3430, 3376]
4 my_labels = 'funny', 'comedy', 'makeup', 'music', 'trailer', 'news', 'humor'
5 plt.pie(my_data, labels=my_labels, autopct='%1.1f%%')
6 plt.title('My Tasks')
7 plt.axis('equal')
8 plt.show()
```

