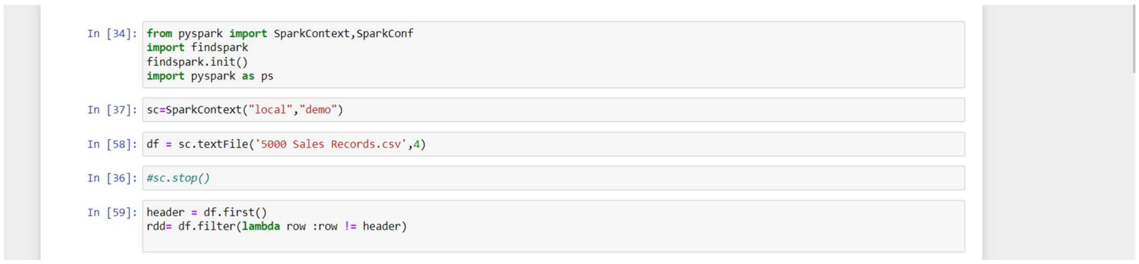
PySpark Project:

Sales dataset.

Requirements



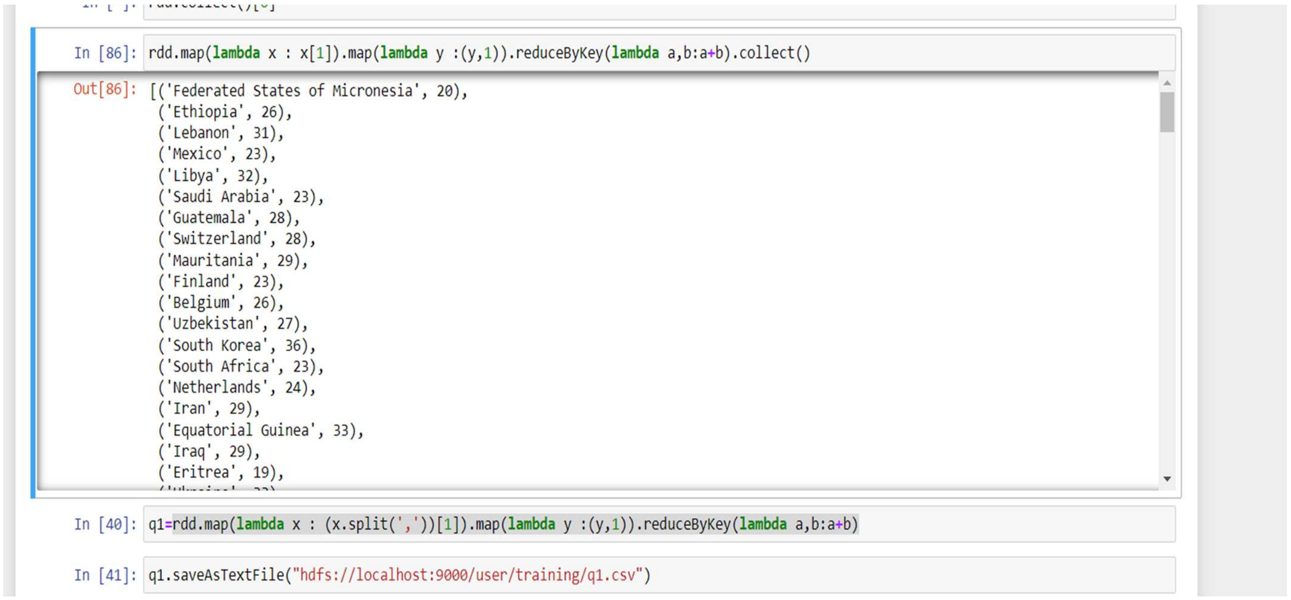
```
In [34]: from pyspark import SparkContext,SparkConf
         import findspark
         findspark.init()
         import pyspark as ps

In [37]: sc=SparkContext("local","demo")

In [58]: df = sc.textFile('5000 Sales Records.csv',4)

In [36]: #sc.stop()

In [59]: header = df.first()
         rdd= df.filter(lambda row :row != header)
```

rdd=rdd.map(lambda x:x.split(","))

rdd.cache()

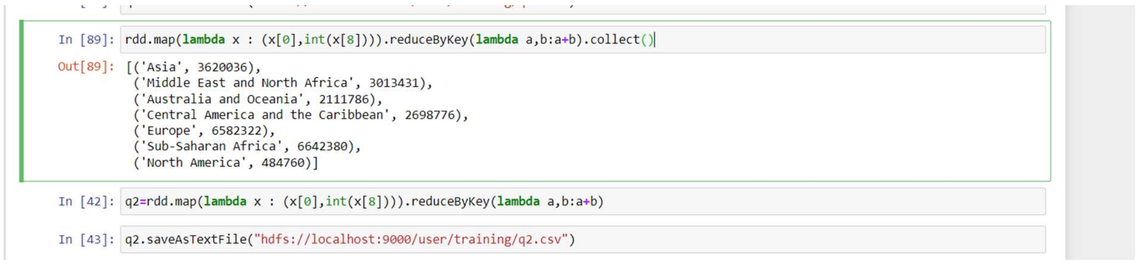
Problem Statements:

1. Display the number of items sold in countries present in the data(Using Hive)



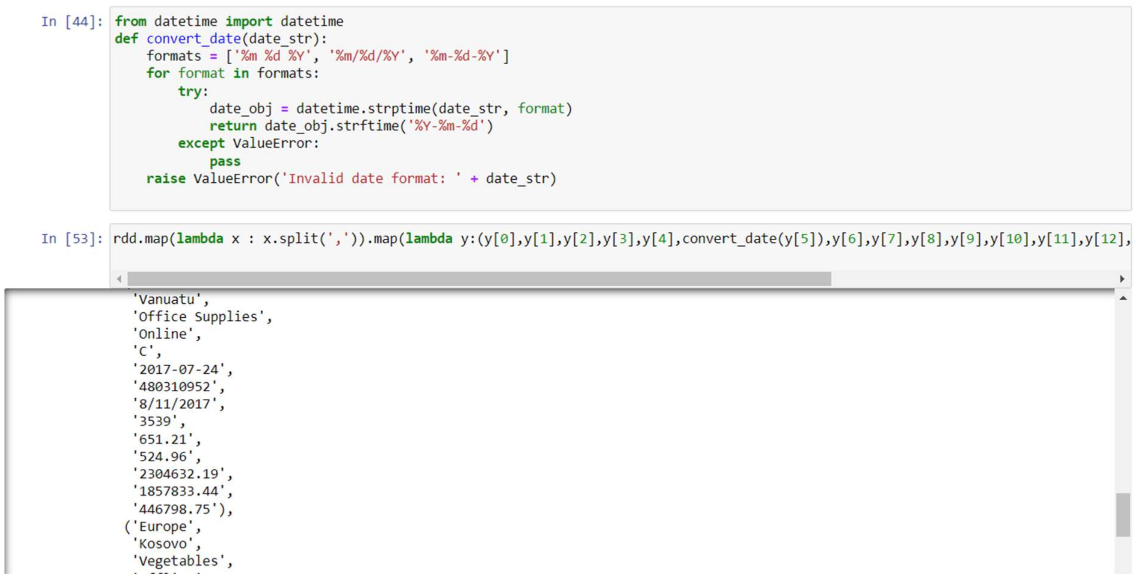
```
In [86]: rdd.map(lambda x : x[1]).map(lambda y :(y,1)).reduceByKey(lambda a,b:a+b).collect()

Out[86]: [('Federated States of Micronesia', 20),
          ('Ethiopia', 26),
          ('Lebanon', 31),
          ('Mexico', 23),
          ('Libya', 32),
          ('Saudi Arabia', 23),
          ('Guatemala', 28),
          ('Switzerland', 28),
          ('Mauritania', 29),
          ('Finland', 23),
          ('Belgium', 26),
          ('Uzbekistan', 27),
          ('South Korea', 36),
          ('South Africa', 23),
          ('Netherlands', 24),
          ('Iran', 29),
          ('Equatorial Guinea', 33),
          ('Iraq', 29),
          ('Eritrea', 19),

In [40]: q1=rdd.map(lambda x : (x.split(','))[1]).map(lambda y :(y,1)).reduceByKey(lambda a,b:a+b)

In [41]: q1.saveAsTextFile("hdfs://localhost:9000/user/training/q1.csv")
```

2. Display the number of units sold in each region.(Using Hive)



```
In [89]: rdd.map(lambda x : (x[0],int(x[8]))).reduceByKey(lambda a,b:a+b).collect()

Out[89]: [('Asia', 3620036),
          ('Middle East and North Africa', 3013431),
          ('Australia and Oceania', 2111786),
          ('Central America and the Caribbean', 2698776),
          ('Europe', 6582322),
          ('Sub-Saharan Africa', 6642380),
          ('North America', 484760)]

In [42]: q2=rdd.map(lambda x : (x[0],int(x[8]))).reduceByKey(lambda a,b:a+b)

In [43]: q2.saveAsTextFile("hdfs://localhost:9000/user/training/q2.csv")
```

3.Display the 10 most recent sales.(Using Hive)



```
In [44]: from datetime import datetime
         def convert_date(date_str):
             formats = ['%m %d %Y', '%m/%d/%Y', '%m-%d-%Y']
             for format in formats:
                 try:
                     date_obj = datetime.strptime(date_str, format)
                     return date_obj.strftime('%Y-%m-%d')
                 except ValueError:
                     pass
             raise ValueError('Invalid date format: ' + date_str)
```

```
In [53]: rdd.map(lambda x : x.split(',')).map(lambda y:(y[0],y[1],y[2],y[3],y[4],convert_date(y[5]),y[6],y[7],y[8],y[9],y[10],y[11],y[12],
```

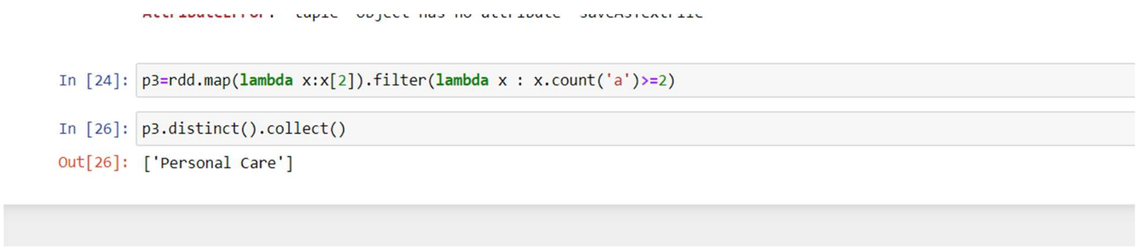
```
  'Vanuatu',
  'Office Supplies',
  'Online',
  'C',
  '2017-07-24',
  '480310952',
  '8/11/2017',
  '3539',
  '651.21',
  '524.96',
  '2304632.19',
  '1857833.44',
  '446798.75'),
 ('Europe',
  'Kosovo',
  'Vegetables',
```

rdd.map(lambda x : x.split(',')).map(lambda y:(y[0],y[1],y[2],y[3],y[4],convert_date(y[5]),y[6],y[7],y[8],y[9],y[10],y[11],y[12],y[13])).sortBy(lambda cols: cols[5],False).take(10)

q3=rdd.map(lambda x : x.split(',')).map(lambda y:(y[0],y[1],y[2],y[3],y[4],convert_date(y[5]),y[6],y[7],y[8],y[9],y[10],y[11],y[12],y[13])).sortBy(lambda cols: cols[5],False)

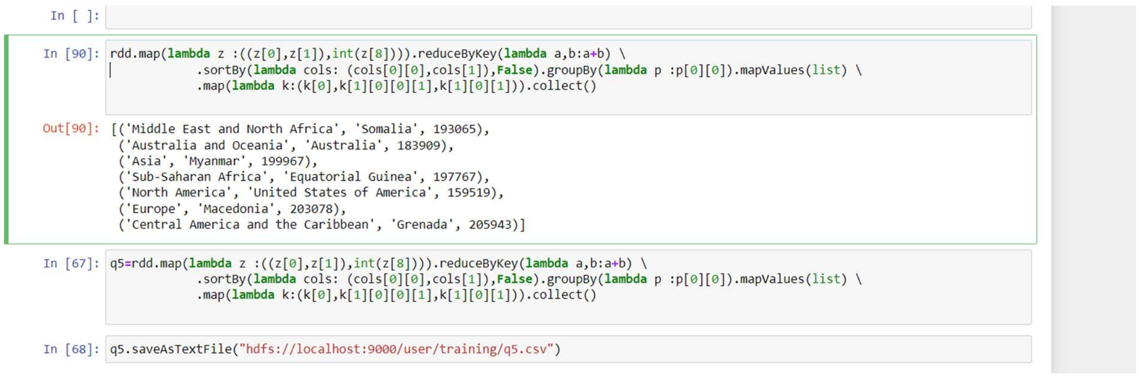
q3.saveAsTextFile("hdfs://localhost:9000/user/training/q3.csv")

4.Display the products with atleast 2 occurences of 'a'. (Using spark)



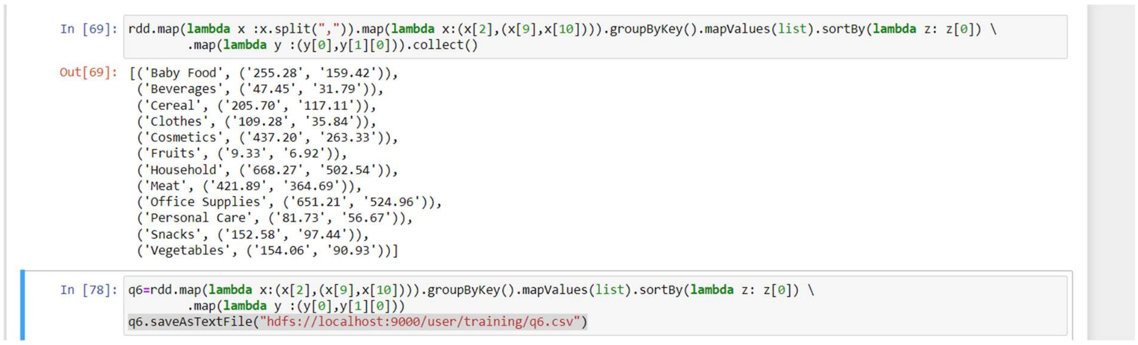
```
AttributeError: 'tuple' object has no attribute 'saveAsTextFile'

In [24]: p3=rdd.map(lambda x:x[2]).filter(lambda x : x.count('a')>=2)

In [26]: p3.distinct().collect()

Out[26]: ['Personal Care']
```

5.Display country in each region with highest units sold.



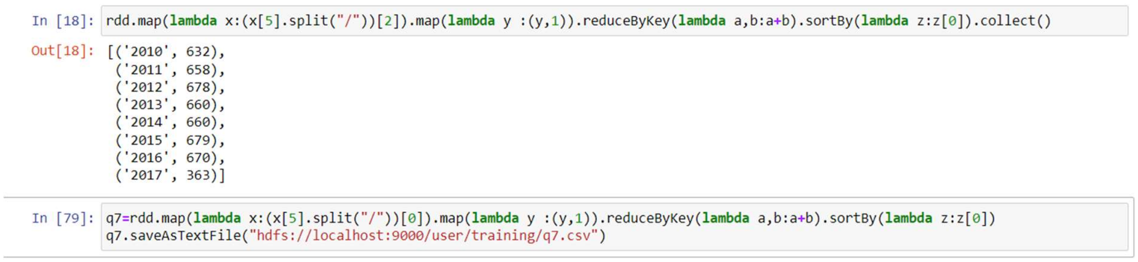
```
In [ ]:

In [90]: rdd.map(lambda z :((z[0],z[1]),int(z[8]))).reduceByKey(lambda a,b:a+b) \
                     .sortBy(lambda cols: (cols[0][0],cols[1]),False).groupBy(lambda p :p[0][0]).mapValues(list) \
                     .map(lambda k:(k[0],k[1][0][0][1],k[1][0][1])).collect()

Out[90]: [('Middle East and North Africa', 'Somalia', 193065),
          ('Australia and Oceania', 'Australia', 183909),
          ('Asia', 'Myanmar', 199967),
          ('Sub-Saharan Africa', 'Equatorial Guinea', 197767),
          ('North America', 'United States of America', 159519),
          ('Europe', 'Macedonia', 203078),
          ('Central America and the Caribbean', 'Grenada', 205943)]

In [67]: q5=rdd.map(lambda z :((z[0],z[1]),int(z[8]))).reduceByKey(lambda a,b:a+b) \
                     .sortBy(lambda cols: (cols[0][0],cols[1]),False).groupBy(lambda p :p[0][0]).mapValues(list) \
                     .map(lambda k:(k[0],k[1][0][0][1],k[1][0][1])).collect()

In [68]: q5.saveAsTextFile("hdfs://localhost:9000/user/training/q5.csv")
```

6.Display the unit price and unit cost of each item in ascending order.



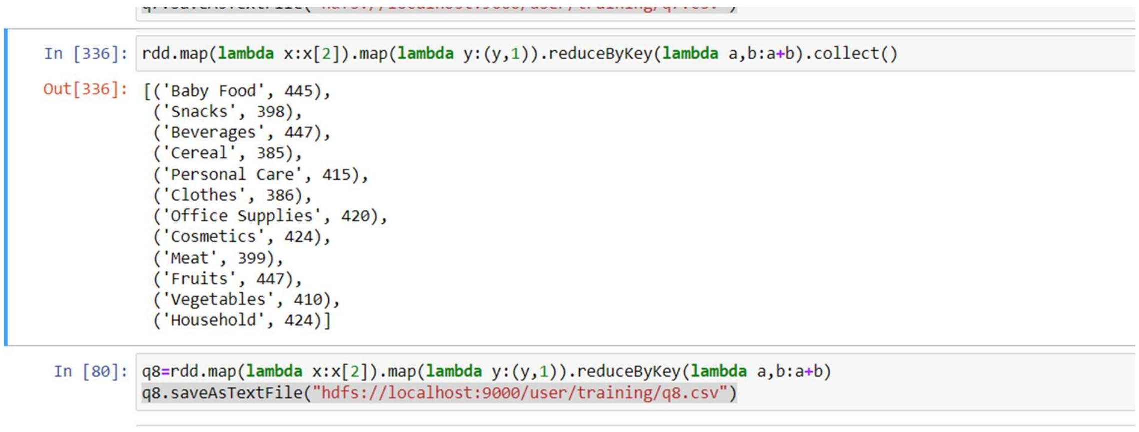
```
In [69]: rdd.map(lambda x :x.split(",")).map(lambda x:(x[2],(x[9],x[10]))).groupByKey().mapValues(list).sortBy(lambda z: z[0]) \
                 .map(lambda y :(y[0],y[1][0])).collect()

Out[69]: [('Baby Food', ('255.28', '159.42')),
          ('Beverages', ('47.45', '31.79')),
          ('Cereal', ('205.70', '117.11')),
          ('Clothes', ('109.28', '35.84')),
          ('Cosmetics', ('437.20', '263.33')),
          ('Fruits', ('9.33', '6.92')),
          ('Household', ('668.27', '502.54')),
          ('Meat', ('421.89', '364.69')),
          ('Office Supplies', ('651.21', '524.96')),
          ('Personal Care', ('81.73', '56.67')),
          ('Snacks', ('152.58', '97.44')),
          ('Vegetables', ('154.06', '90.93'))]

In [78]: q6=rdd.map(lambda x:(x[2],(x[9],x[10]))).groupByKey().mapValues(list).sortBy(lambda z: z[0]) \
                 .map(lambda y :(y[0],y[1][0]))
         q6.saveAsTextFile("hdfs://localhost:9000/user/training/q6.csv")
```

7.Display the number of sales yearwise. (Using pyspark)



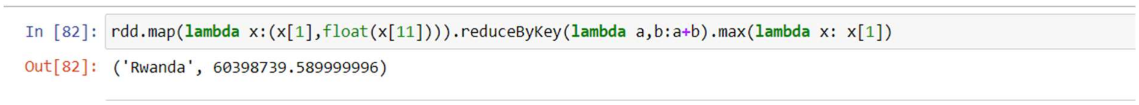
```
In [18]: rdd.map(lambda x:(x[5].split("/"))[2]).map(lambda y :(y,1)).reduceByKey(lambda a,b:a+b).sortBy(lambda z:z[0]).collect()

Out[18]: [('2010', 632),
          ('2011', 658),
          ('2012', 678),
          ('2013', 660),
          ('2014', 660),
          ('2015', 679),
          ('2016', 670),
          ('2017', 363)]

In [79]: q7=rdd.map(lambda x:(x[5].split("/"))[0]).map(lambda y :(y,1)).reduceByKey(lambda a,b:a+b).sortBy(lambda z:z[0])
         q7.saveAsTextFile("hdfs://localhost:9000/user/training/q7.csv")
```

8.Display the number of orders for each item



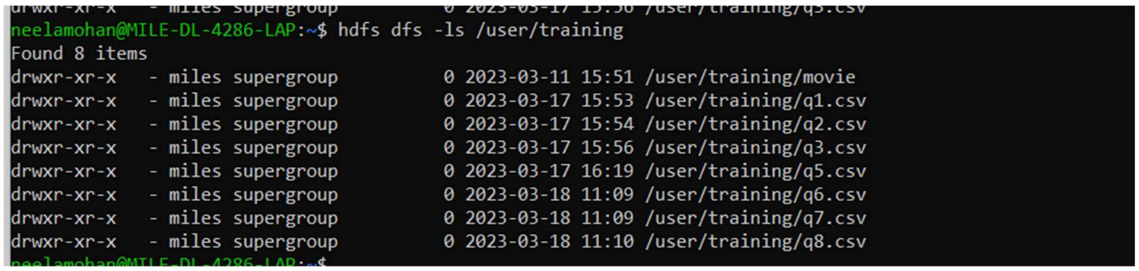
```
In [336]: rdd.map(lambda x:x[2]).map(lambda y:(y,1)).reduceByKey(lambda a,b:a+b).collect()

Out[336]: [('Baby Food', 445),
           ('Snacks', 398),
           ('Beverages', 447),
           ('Cereal', 385),
           ('Personal Care', 415),
           ('Clothes', 386),
           ('Office Supplies', 420),
           ('Cosmetics', 424),
           ('Meat', 399),
           ('Fruits', 447),
           ('Vegetables', 410),
           ('Household', 424)]

In [80]: q8=rdd.map(lambda x:x[2]).map(lambda y:(y,1)).reduceByKey(lambda a,b:a+b)
         q8.saveAsTextFile("hdfs://localhost:9000/user/training/q8.csv")
```

9.Display the country with highest sale



```
In [82]: rdd.map(lambda x:(x[1],float(x[11]))).reduceByKey(lambda a,b:a+b).max(lambda x: x[1])

Out[82]: ('Rwanda', 60398739.589999996)
```

All the files moved to hdfs:



```
neelamohan@MILE-DL-4286-LAP:~$ hdfs dfs -ls /user/training
Found 8 items
drwxr-xr-x   - miles supergroup          0 2023-03-11 15:51 /user/training/movie
drwxr-xr-x   - miles supergroup          0 2023-03-17 15:53 /user/training/q1.csv
drwxr-xr-x   - miles supergroup          0 2023-03-17 15:54 /user/training/q2.csv
drwxr-xr-x   - miles supergroup          0 2023-03-17 15:56 /user/training/q3.csv
drwxr-xr-x   - miles supergroup          0 2023-03-17 16:19 /user/training/q5.csv
drwxr-xr-x   - miles supergroup          0 2023-03-18 11:09 /user/training/q6.csv
drwxr-xr-x   - miles supergroup          0 2023-03-18 11:09 /user/training/q7.csv
drwxr-xr-x   - miles supergroup          0 2023-03-18 11:10 /user/training/q8.csv
neelamohan@MILE-DL-4286-LAP:~$
```