**Description:** This is a list of assignments which is part of Apache Spark L1 training

**Technology**: Apache Spark (RDD API)

**Input Datasets:**

RealEstate.txt

SampleData.txt

Assignment 1. Properties for given location

a. Read the RealEstate.txt file as an RDD

b. List the properties in Location = "Thomas County"

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

q1 = df1.filter(lambda x: "Thomas County" == x[col.index("Location")])

print("No of properties in Thomas County location",q1.count())

q1.collect()



Assignment 2. Display unique locations in command line

a. Read the RealEstate.txt file as an RDD

b. Display the unique locations in command line

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

df1.map(lambda x: x[col.index("Location")]).distinct().collect()



Assignment 3. Compute Price of Property

a. Read the RealEstate.txt file as an RDD

b. Create a RDD having PropertyID, Location, Price (= size \* Price SQ Ft)

c. Display the first 10 records in command line

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

asgn3 = df1.map(lambda x: (x[col.index("Property ID")], x[col.index("Location")], i int(x[col.index("Size")])\*float(x[col.index("Price SQ Ft")])))

asgn3.take(10)



Assignment 4. Display Counts

a. Read the RealEstate.txt file as an RDD

b. Display count of Real estate properties for each location in command line

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

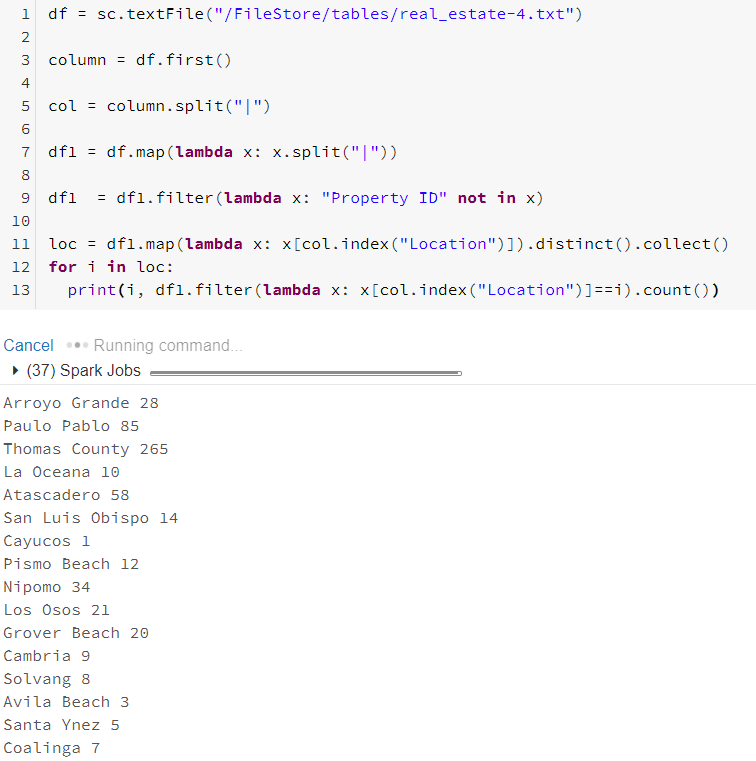
df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

loc = df1.map(lambda x: x[col.index("Location")]).distinct().collect()

for i in loc:

print(i, df1.filter(lambda x: x[col.index("Location")]==i).count())



Assignment 5. Set Operations on RDDs

a. Read the RealEstate.txt file as an RDD

b. Create a RDD of Property IDs having 3 bedrooms

c. Create another RDD of Property IDs having at least 2 bathrooms

d. Create another RDD having 3 bedrooms and at least 2 bathrooms

(Hint: use intersection operator on the above 2 RDDs)

e. save the RDD to a file

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

rdd1 = df1.map(lambda x: (int(x[col.index("Bedrooms")]), x[col.index("Property I ID")])).filter(lambda x: int(x[0])==3).map(lambda x: x[1])

print(rdd1.count())

rdd2 = df1.map(lambda x: (int(x[col.index("Bathrooms")]), x[col.index("Property I ID")])).filter(lambda x: int(x[0])>=2).map(lambda x: x[1])

print(rdd2.count())

rdd3 = rdd1.intersection(rdd2)

print(rdd3.count())

rdd3.saveAsTextFile("./rdd\_assgn\_5")



Assignment 6. Get list of properties satisfying the criteria

a. Read the RealEstate.txt file as an RDD

b. Find the list of properties as per the below criteria and save it to a file

i. Have 3 bedrooms

ii. Have at least 2 bathrooms

iii. Price <= 500000 (Price = size \* Price SQ Ft)

c. Save the RDD to a file

**Code:**

df = sc.textFile("/FileStore/tables/real\_estate-4.txt")

column = df.first()

col = column.split("|")

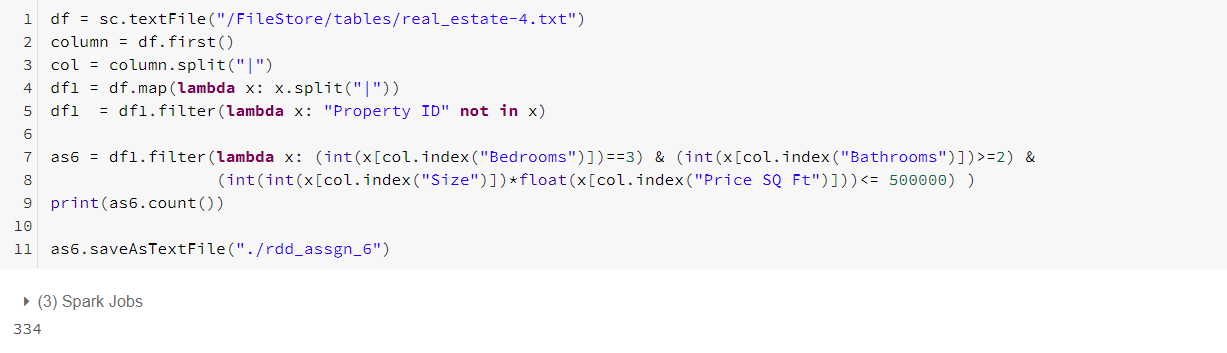
df1 = df.map(lambda x: x.split("|"))

df1 = df1.filter(lambda x: "Property ID" not in x)

as6 = df1.filter(lambda x: (int(x[col.index("Bedrooms")])==3) & ( int(x[col.index("Bathrooms")])>=2) & (int(int(x[col.index("Size")])\*float(x[col.index("Price SQ Ft")]))<= 500000) )

print(as6.count())

as6.saveAsTextFile("./rdd\_assgn\_6")



Assignment 7. Find Word Counts

a. Read the SampleData.txt as an RDD

b. Remove all punctuations marks from the text

c. Remove all numeric values

d. Find the total count of occurrences for each word

**Code:**

df2 = sc.textFile("/FileStore/tables/sample\_data.txt")

df2.take(5)

def cln(x):

# define punctuation

punctuations = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~123456789'''

# remove punctuation from the string

no\_punct = ""

for char in x:

if char not in punctuations:

no\_punct = no\_punct + char

return no\_punct

df2.flatMap(lambda line: line.split(" ")).map(cln).map(lambda word: (word, 1)).reduceByKey(lambda a,b: a+b).collect()

