

Mobivity Inc. 4/6 project for cs students

SHANMATHI RAJESH | shrajesh@eng.ucsd.edu | 858-265-9151

Problem Definition:

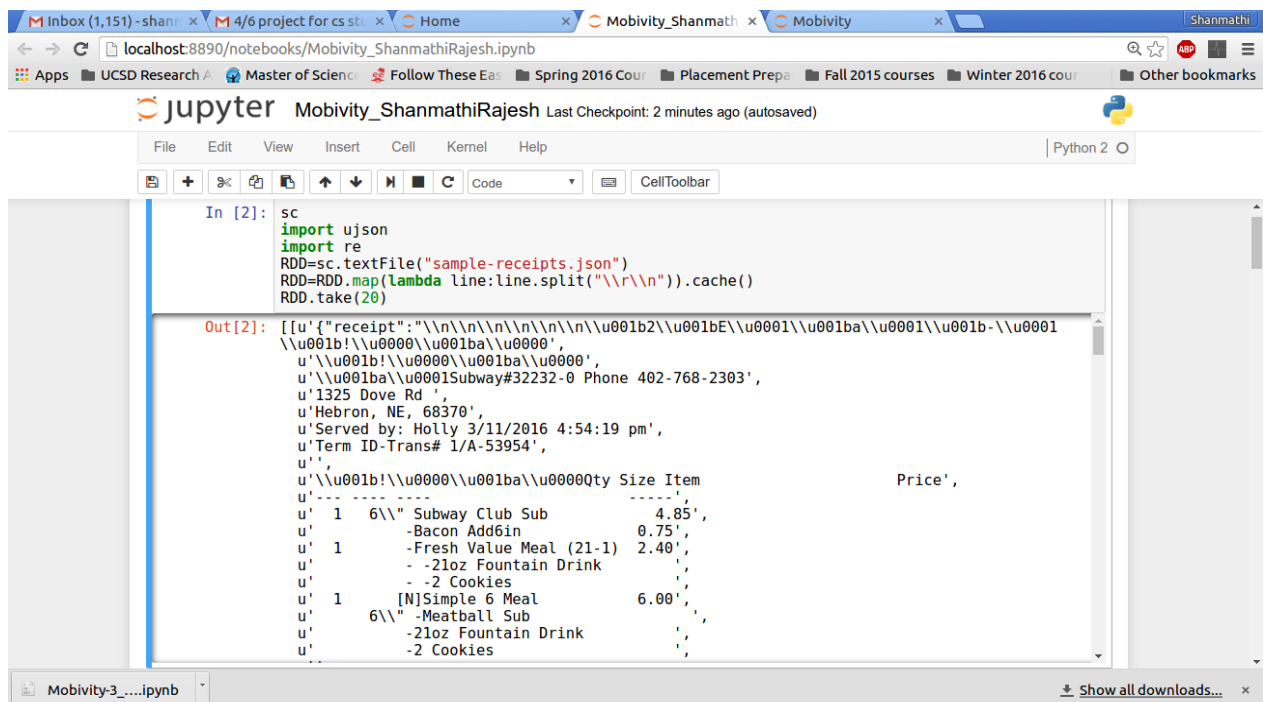
A JSON file containing 250k receipts is given. The goal is to find the number of sandwiches sold, the number of drinks sold and the total length of the sandwiches sold in feet units.

Approach:

As the given problem involves mining a large dataset and extracting information from it, this problem falls under the category of “Big data Analytics”. Such data mining problems can be efficiently solved in Spark environment. **Apache Spark** is an open-source big data processing framework. Its features like speed, ease of use, in-memory distributed computing and sophisticated analytics provides it a great advantage over other traditional map-reduce technology like Hadoop. Spark lets us use Java, Scala and Python to write easy applications. As Python is a powerful, high-level programming language and as it has many libraries for data analytics applications, I have chosen to implement my program using Python in Spark, ie, using **PySpark**.

I have attached my entire code as an **ipython notebook**. Here, I have explained the steps that I have followed with screen-shots of ipython notebook showing the results.

1) Reading the JSON input file and splitting the lines in the file using delimiters and storing it in RDD. An RDD is the basic data structure used in Spark. It represents an immutable, partitioned collection of elements that can be operated on in parallel.



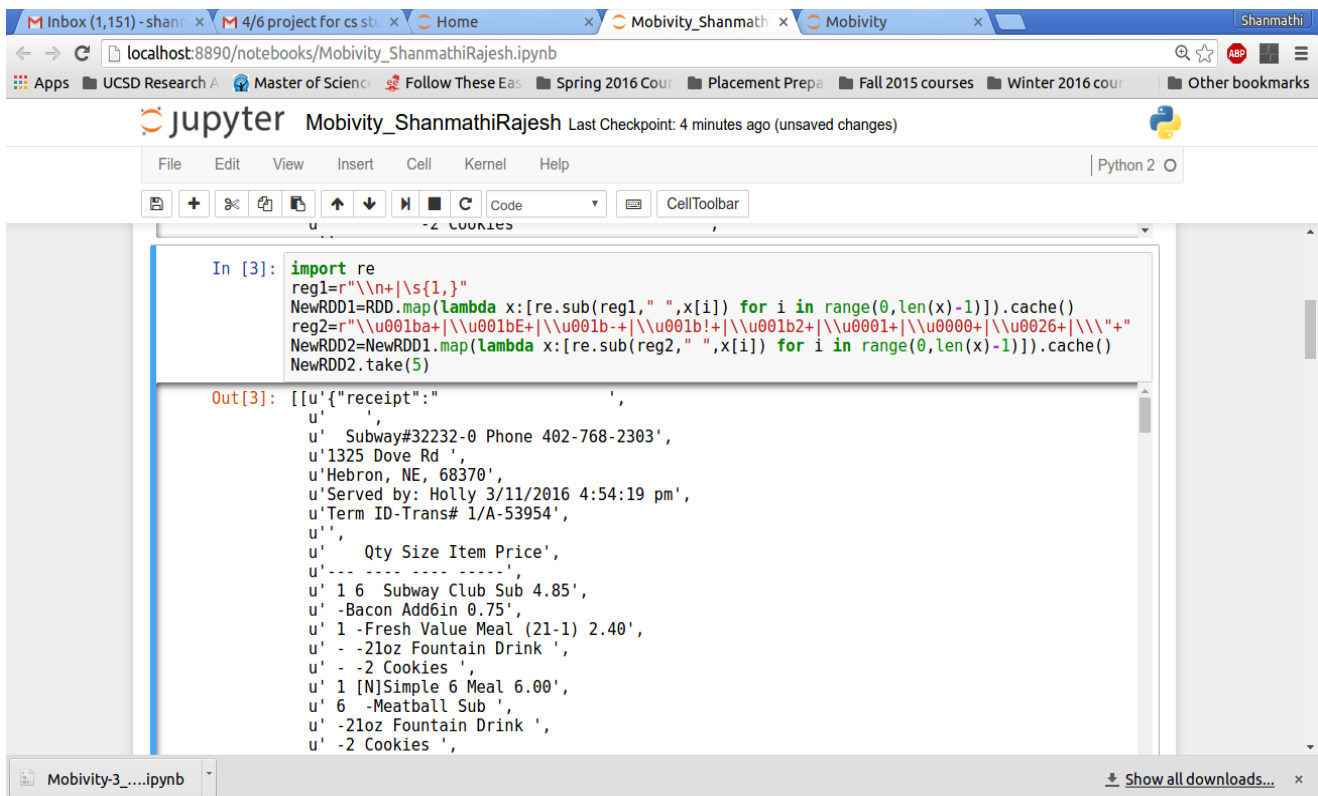
The screenshot shows a Jupyter Notebook interface with the following code in the input cell:

```
In [2]: sc
import ujson
import re
RDD=sc.textFile("sample-receipts.json")
RDD=RDD.map(lambda line:line.split("\r\n")).cache()
RDD.take(20)
```

The output cell displays a list of 20 receipt records. The first record is a dictionary with keys 'receipt' and 'line'. The 'line' key contains a string representing a receipt, which is then formatted into a table. The table has columns for 'Qty', 'Size', 'Item', and 'Price'.

Qty	Size	Item	Price
1	6"	Subway Club Sub	4.85
		-Bacon Add6in	0.75
1		-Fresh Value Meal (21-1)	2.40
		-21oz Fountain Drink	
		-2 Cookies	
1		[N]Simple 6 Meal	6.00
	6"	-Meatball Sub	
		-21oz Fountain Drink	
		-2 Cookies	

2) Formatting the lines using regular expressions to remove unnecessary spaces, escape sequences, unicode characters and storing it in NewRDD.



```
In [3]: import re
reg1=r"\\n+|\\s{1,}"
NewRDD1=RDD.map(lambda x:[re.sub(reg1,"",x[i]) for i in range(0,len(x)-1)]).cache()
reg2=r"\\u001b+|\\u001bE+|\\u001b-+|\\u001b!+|\\u001b2+|\\u0001+|\\u0000+|\\u0026+|\\\\"+"
NewRDD2=NewRDD1.map(lambda x:[re.sub(reg2,"",x[i]) for i in range(0,len(x)-1)]).cache()
NewRDD2.take(5)

Out[3]: [[u{"receipt":",
u'
u' Subway#32232-0 Phone 402-768-2303',
u'1325 Dove Rd ',
u'Hebron, NE, 68370',
u'Served by: Holly 3/11/2016 4:54:19 pm',
u'Term ID-Trans# 1/A-53954',
u'',
u' Qty Size Item Price',
u'-----',
u' 1 6 Subway Club Sub 4.85',
u' -Bacon Add6in 0.75',
u' 1 -Fresh Value Meal (21-1) 2.40',
u' -21oz Fountain Drink ',
u' -2 Cookies ',
u' 1 [N]Simple 6 Meal 6.00',
u' 6 -Meatball Sub ',
u' -21oz Fountain Drink ',
u' -2 Cookies ',
```

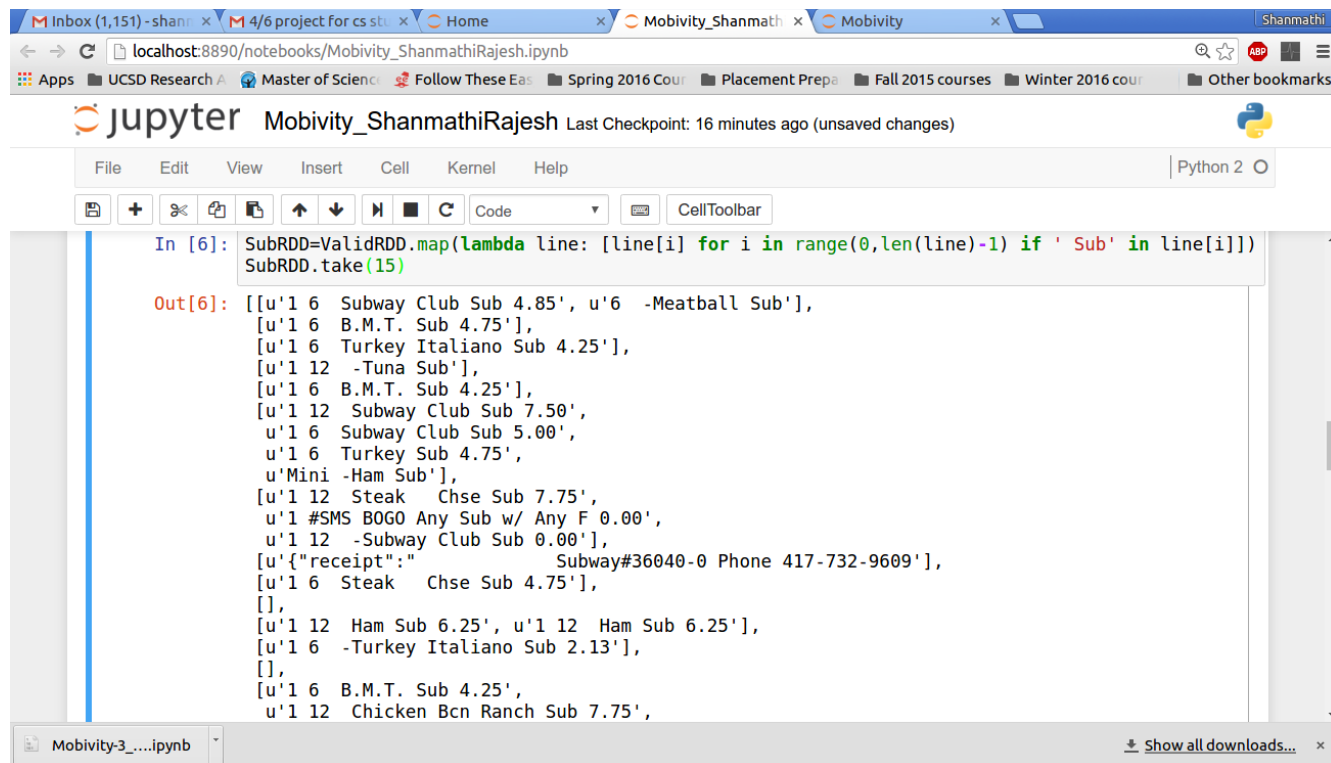
3) The extra spaces in each element is stripped and stored in ValidRDD which contains a list of unicode strings. The empty unicode strings are also filtered out.



```
In [4]: ValidRDD=NewRDD2.map(lambda y:[(y[i].strip()) for i in range(0,len(y)-1)])\
.map(lambda y: [y[i] for i in range(0,len(y)-1) if y[i] != '']).cache()
ValidRDD.take(5)

Out[4]: [[u{"receipt":",
u'Subway#32232-0 Phone 402-768-2303',
u'1325 Dove Rd',
u'Hebron, NE, 68370',
u'Served by: Holly 3/11/2016 4:54:19 pm',
u'Term ID-Trans# 1/A-53954',
u'Qty Size Item Price',
u'-----',
u'1 6 Subway Club Sub 4.85',
u'-Bacon Add6in 0.75',
u'1 -Fresh Value Meal (21-1) 2.40',
u'-21oz Fountain Drink',
u'-2 Cookies',
u'1 [N]Simple 6 Meal 6.00',
u'6 -Meatball Sub',
u'-21oz Fountain Drink',
u'-2 Cookies',
u'Sub Total 14.00',
u'General Sales Tax (6.5%) 0.91',
u'Total (Eat In) 14.91',
```

4) From each element of the ValidRDD only those lines containing the word “Sub” are filtered out so as to find the quantity and the size of each sandwich sold. This list of strings is stored in SubRDD.

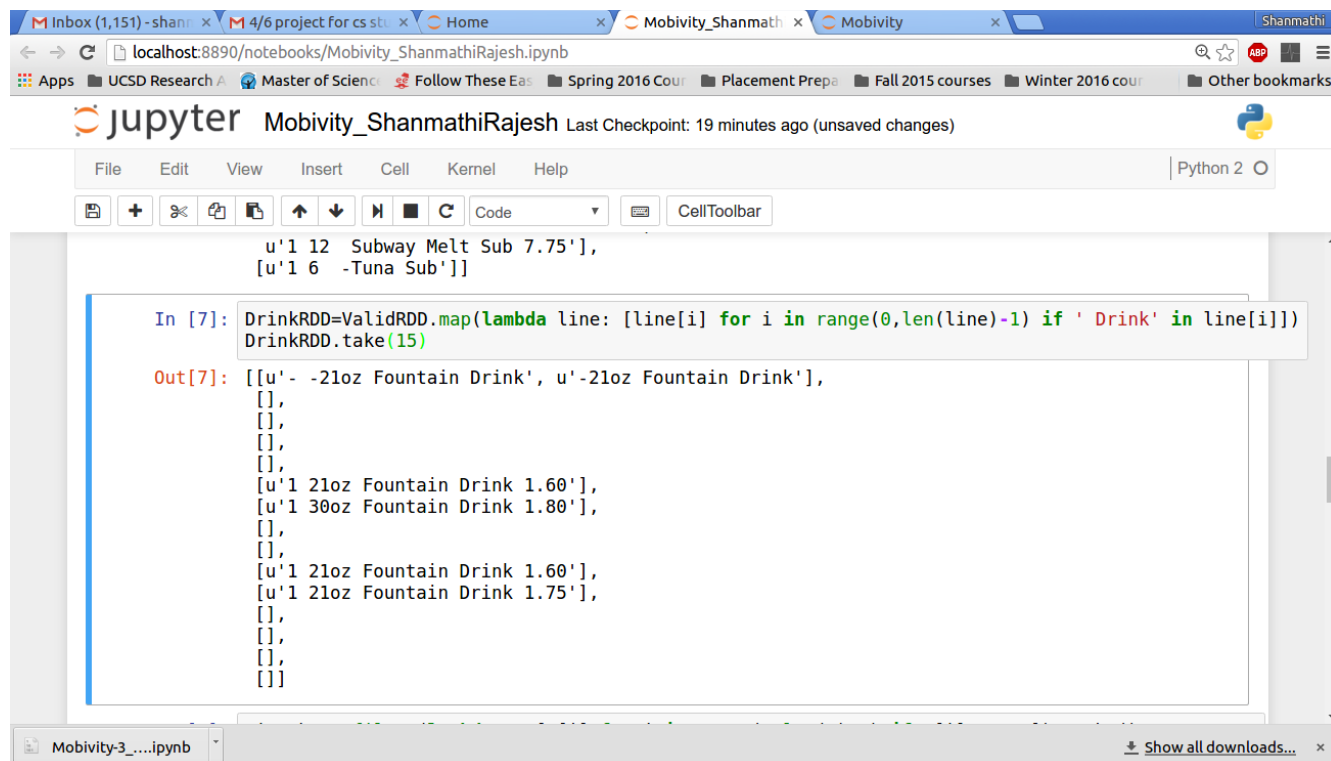


The screenshot shows a Jupyter Notebook interface with the following content:

```
In [6]: SubRDD=ValidRDD.map(lambda line: [line[i] for i in range(0,len(line)-1) if ' Sub' in line[i]])
SubRDD.take(15)
```

```
Out[6]: [[u'1 6 Subway Club Sub 4.85', u'6 -Meatball Sub'],
[u'1 6 B.M.T. Sub 4.75'],
[u'1 6 Turkey Italiano Sub 4.25'],
[u'1 12 -Tuna Sub'],
[u'1 6 B.M.T. Sub 4.25'],
[u'1 12 Subway Club Sub 7.50'],
[u'1 6 Subway Club Sub 5.00'],
[u'1 6 Turkey Sub 4.75'],
[u'Mini -Ham Sub'],
[u'1 12 Steak Chse Sub 7.75'],
[u'1 #SMS BOGO Any Sub w/ Any F 0.00'],
[u'1 12 -Subway Club Sub 0.00'],
[u'{"receipt": "Subway#36040-0 Phone 417-732-9609"}'],
[u'1 6 Steak Chse Sub 4.75'],
[],
[u'1 12 Ham Sub 6.25', u'1 12 Ham Sub 6.25'],
[u'1 6 -Turkey Italiano Sub 2.13'],
[],
[u'1 6 B.M.T. Sub 4.25'],
[u'1 12 Chicken Bcn Ranch Sub 7.75'],
```

5) Similarly, to find the number of drinks sold, the lines of the receipt containing the word “Drink” are filtered out separately and stored in DrinkRDD.



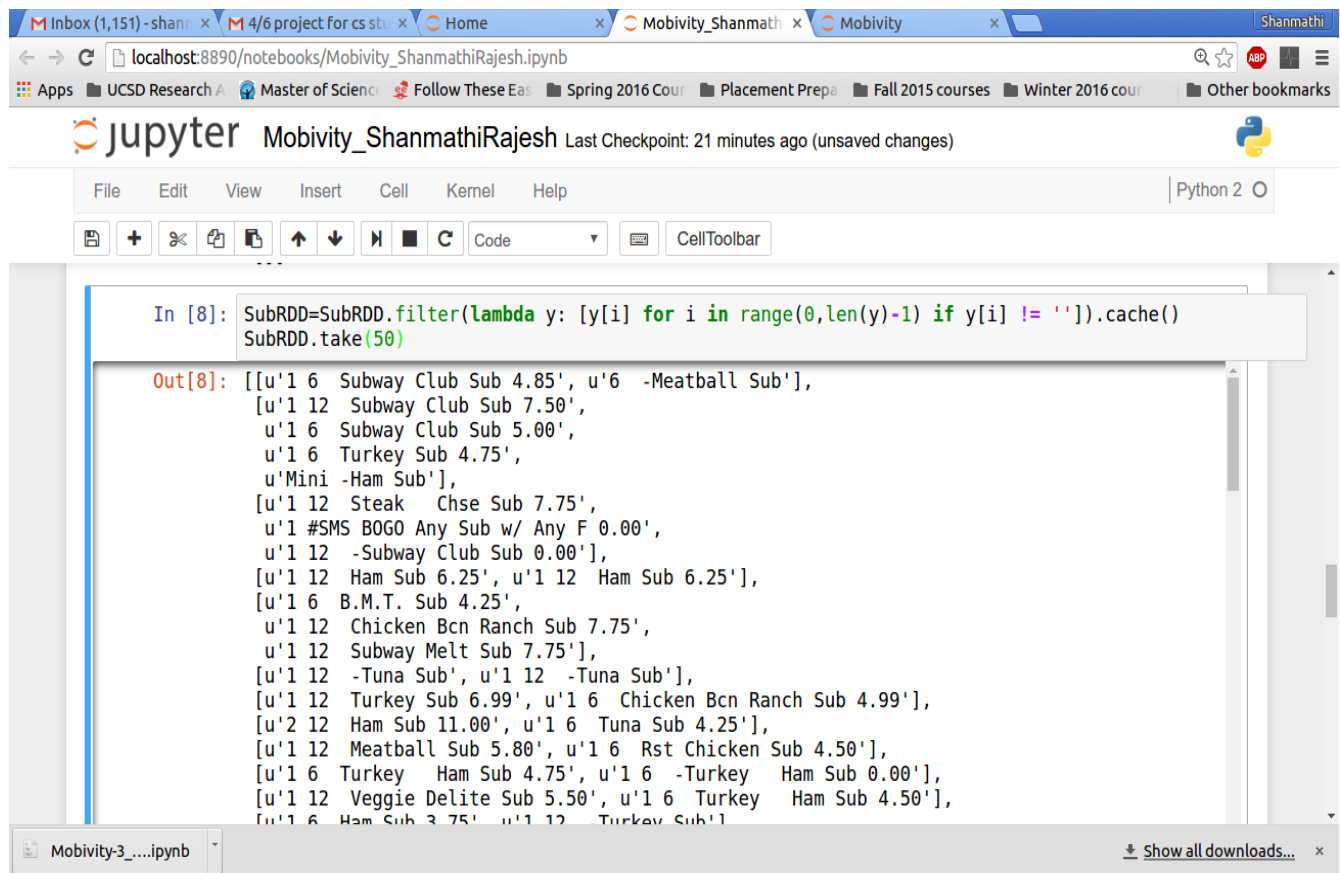
The screenshot shows a Jupyter Notebook interface with the following content:

```
u'1 12 Subway Melt Sub 7.75'],
[u'1 6 -Tuna Sub']]
```

```
In [7]: DrinkRDD=ValidRDD.map(lambda line: [line[i] for i in range(0,len(line)-1) if ' Drink' in line[i]])
DrinkRDD.take(15)
```

```
Out[7]: [[u'- 21oz Fountain Drink', u'-21oz Fountain Drink'],
[],
[],
[],
[],
[u'1 21oz Fountain Drink 1.60'],
[u'1 30oz Fountain Drink 1.80'],
[],
[],
[u'1 21oz Fountain Drink 1.60'],
[u'1 21oz Fountain Drink 1.75'],
[],
[],
[],
[]]
```

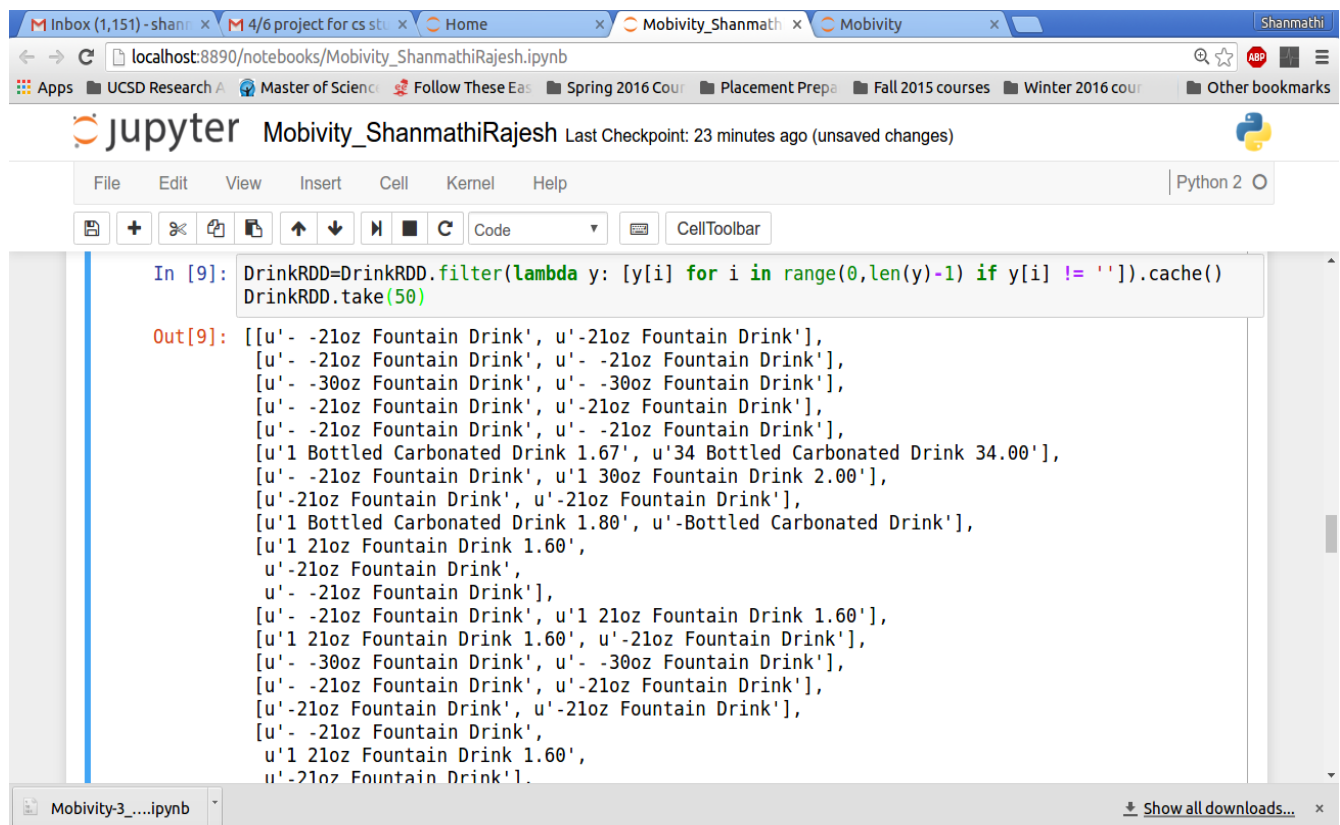
6) Empty lists and null strings are deleted from SubRDD and DrinkRDD as shown below.



The screenshot shows a Jupyter Notebook interface with the title 'Mobivity_ShanmathiRajesh'. The browser address bar indicates the notebook is running on localhost:8890. The notebook contains a code cell with the following input and output:

```
In [8]: SubRDD=SubRDD.filter(lambda y: [y[i] for i in range(0,len(y)-1) if y[i] != '']).cache()
SubRDD.take(50)
```

```
Out[8]: [[u'1 6 Subway Club Sub 4.85', u'6 -Meatball Sub'],
[u'1 12 Subway Club Sub 7.50',
u'1 6 Subway Club Sub 5.00',
u'1 6 Turkey Sub 4.75',
u'Mini -Ham Sub'],
[u'1 12 Steak Chse Sub 7.75',
u'1 #SMS BOGO Any Sub w/ Any F 0.00',
u'1 12 -Subway Club Sub 0.00'],
[u'1 12 Ham Sub 6.25', u'1 12 Ham Sub 6.25'],
[u'1 6 B.M.T. Sub 4.25',
u'1 12 Chicken Bcn Ranch Sub 7.75',
u'1 12 Subway Melt Sub 7.75'],
[u'1 12 -Tuna Sub', u'1 12 -Tuna Sub'],
[u'1 12 Turkey Sub 6.99', u'1 6 Chicken Bcn Ranch Sub 4.99'],
[u'2 12 Ham Sub 11.00', u'1 6 Tuna Sub 4.25'],
[u'1 12 Meatball Sub 5.80', u'1 6 Rst Chicken Sub 4.50'],
[u'1 6 Turkey Ham Sub 4.75', u'1 6 -Turkey Ham Sub 0.00'],
[u'1 12 Veggie Delite Sub 5.50', u'1 6 Turkey Ham Sub 4.50'],
[u'1 6 Ham Sub 3.75', u'1 12 Turkey Sub']]
```

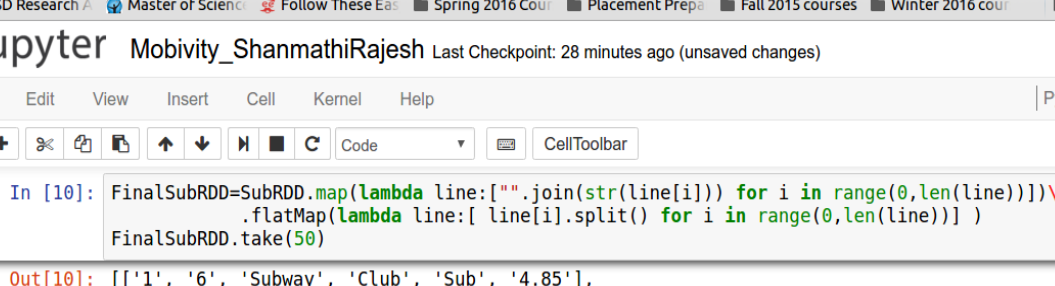


The screenshot shows a Jupyter Notebook interface with the title 'Mobivity_ShanmathiRajesh'. The browser address bar indicates the notebook is running on localhost:8890. The notebook contains a code cell with the following input and output:

```
In [9]: DrinkRDD=DrinkRDD.filter(lambda y: [y[i] for i in range(0,len(y)-1) if y[i] != '']).cache()
DrinkRDD.take(50)
```

```
Out[9]: [[u'- 21oz Fountain Drink', u'-21oz Fountain Drink'],
[u'- 21oz Fountain Drink', u'- 21oz Fountain Drink'],
[u'- 30oz Fountain Drink', u'- 30oz Fountain Drink'],
[u'- 21oz Fountain Drink', u'-21oz Fountain Drink'],
[u'- 21oz Fountain Drink', u'- 21oz Fountain Drink'],
[u'1 Bottled Carbonated Drink 1.67', u'34 Bottled Carbonated Drink 34.00'],
[u'- 21oz Fountain Drink', u'1 30oz Fountain Drink 2.00'],
[u'-21oz Fountain Drink', u'-21oz Fountain Drink'],
[u'1 Bottled Carbonated Drink 1.80', u'-Bottled Carbonated Drink'],
[u'1 21oz Fountain Drink 1.60',
u'-21oz Fountain Drink',
u'- 21oz Fountain Drink'],
[u'- 21oz Fountain Drink', u'1 21oz Fountain Drink 1.60'],
[u'1 21oz Fountain Drink 1.60', u'-21oz Fountain Drink'],
[u'- 30oz Fountain Drink', u'- 30oz Fountain Drink'],
[u'- 21oz Fountain Drink', u'-21oz Fountain Drink'],
[u'-21oz Fountain Drink', u'-21oz Fountain Drink'],
[u'- 21oz Fountain Drink',
u'1 21oz Fountain Drink 1.60',
u'-21oz Fountain Drink']]
```

7) The elements of SubRDD are joined together and converted from unicode strings to normal strings and stored in FinalSubRDD. In the FinalSubRDD list, the first column contains the quantity of sandwich sold and the second column contains the size of the sandwich sold.



The screenshot shows a Jupyter Notebook window titled 'Mobivity_ShanmathiRajesh'. The interface includes a top bar with browser tabs, a file explorer, and a menu bar (File, Edit, View, Insert, Cell, Kernel, Help). Below the menu bar is a toolbar with icons for saving, adding, deleting, and running code. The main area contains a code cell with the following Scala code:

```
In [10]: FinalSubRDD=SubRDD.map(lambda line:["".join(str(line[i])) for i in range(0,len(line))]\
                                .flatMap(lambda line:[ line[i].split() for i in range(0,len(line)) ] )
                                FinalSubRDD.take(50)
```

The output of the code is displayed below the cell:

```
Out[10]: [['1', '6', 'Subway', 'Club', 'Sub', '4.85'],
           ['6', '-Meatball', 'Sub'],
           ['1', '12', 'Subway', 'Club', 'Sub', '7.50'],
           ['1', '6', 'Subway', 'Club', 'Sub', '5.00'],
           ['1', '6', 'Turkey', 'Sub', '4.75'],
           ['Mini', '-Ham', 'Sub'],
           ['1', '12', 'Steak', 'Chse', 'Sub', '7.75'],
           ['1', '#SMS', 'BOGO', 'Any', 'Sub', 'w/', 'Any', 'F', '0.00'],
           ['1', '12', '-Subway', 'Club', 'Sub', '0.00'],
           ['1', '12', 'Ham', 'Sub', '6.25'],
           ['1', '12', 'Ham', 'Sub', '6.25'],
           ['1', '6', 'B.M.T.', 'Sub', '4.25'],
           ['1', '12', 'Chicken', 'Bcn', 'Ranch', 'Sub', '7.75'],
           ['1', '12', 'Subway', 'Melt', 'Sub', '7.75'],
           ['1', '12', '-Tuna', 'Sub'],
           ['1', '12', '-Tuna', 'Sub'],
           ['1', '12', 'Turkey', 'Sub', '6.99'],
           ['1', '6', 'Chicken', 'Bcn', 'Ranch', 'Sub', '4.99'],
           ['1', '12', 'Ham', 'Sub', '11.00']]
```

8) Each line of the FinalSubRDD is passed to the function findNum which outputs the [Quantity, Length] of each sandwich.

Mobivity code - sha... x 4/6 project for cs st... x Mobivity_Shanmath... x Mobivity_Shanmath... x DS Homework Submis... x Shanmathi...

localhost:8890/notebooks/Mobivity_ShanmathiRajesh_Sandwich.ipynb

Apps UCSD Research A Master of Science Follow These Eas Spring 2016 Cour Placement Prepa Fall 2015 courses Winter 2016 cour Other bookmarks

Jupyter Mobivity_ShanmathiRajesh_Sandwich Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

```
In [8]: def findNum(line):
        try:
            if len(line)>3: #For different types of sandwiches
                return [int(line[0]),int(line[1])]
            elif len(line)==3: #For sandwiches in Meals(combos)
                return [1,int(line[0])]
            elif line[0]=='Mini': #For Mini Ham sandwich
                return [1,0]
            else:
                return [0,0]
        except ValueError, e:
            return [1,2]

        Mob=FinalSubRDD.map(lambda line:findNum(line))
        Mob.take(50)
```

```
Out[8]: [[1, 6],
          [1, 6],
          [1, 12],
          [1, 6],
          [1, 6],
          [1, 2],
          [1, 12],
          [1, 2],
          [1, 12],
          [1, 12],
          [1, 12],
          [1, 12],
          [1, 6]]
```

Mobility-Fin....ipynb Mobivity-3_...ipynb

Show all downloads...

Number of Sandwiches sold = 857641

Length of Sandwiches sold = 970836 inches = 80903 feet

10) The elements of DrinkRDD are joined together as strings and stored in a RDD called FinalDrink which contains each drink as an element in the list.

Inbox (1,151) - shan...xM 4/6 project for cs stu...xHome

Mobivity_Shanmathi xMobivity_Shanmathi xDS Homework Submiss...xShanmathi

localhost:8890/notebooks/Mobivity_ShanmathiRajesh_Drink.ipynb

AppsUCSD ResearchAMaster of ScienceFollow These EaSpring 2016 CourPlacement PrepaFall 2015 coursesWinter 2016 courOther bookmarks

jupyter Mobivity_ShanmathiRajesh_Drink Last Checkpoint: an hour ago (unsaved changes)

Python 2

FileEditViewInsertCellKernelHelp

CodeCellToolBar

```
In [14]: FinalDrink=DrinkRDD.flatMap(lambda line:["".join(str(line[i])) for i in range(0,len(line))])
FinalDrink.take(50)

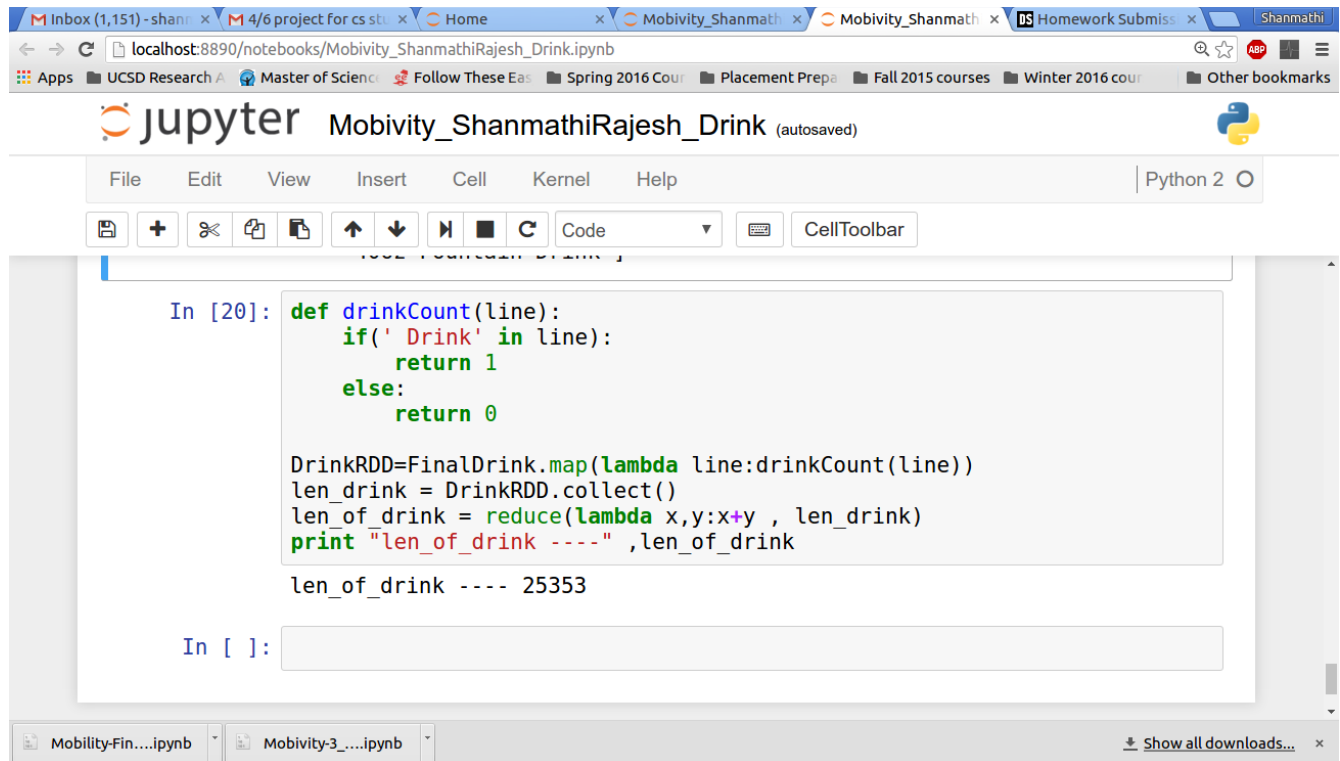
Out[14]: ['- 21oz Fountain Drink',
'-21oz Fountain Drink',
'- 21oz Fountain Drink',
'- 21oz Fountain Drink',
'- 30oz Fountain Drink',
'- 30oz Fountain Drink',
'- 21oz Fountain Drink',
'-21oz Fountain Drink',
'- 21oz Fountain Drink',
'- 21oz Fountain Drink',
'1 Bottled Carbonated Drink 1.67',
'34 Bottled Carbonated Drink 34.00',
'- 21oz Fountain Drink',
'1 30oz Fountain Drink 2.00',
'-21oz Fountain Drink',
'-21oz Fountain Drink',
'1 Bottled Carbonated Drink 1.80',
'-Bottled Carbonated Drink',
'1 21oz Fountain Drink 1.60']
```

Mobivity-Fin....ipynbMobivity-3....ipynb

Show all downloads...

11) The number of drinks sold is calculated by counting the number of elements in the DrinkRDD, adding all the values using reduce function and stored in len_of_drink.

The number of drinks sold is 25353



The screenshot shows a Jupyter Notebook titled 'Mobivity_ShanmathiRajesh_Drink' running on a local host. The code in the cell is as follows:

```
In [20]: def drinkCount(line):
          if(' Drink' in line):
              return 1
          else:
              return 0

          DrinkRDD=FinalDrink.map(lambda line:drinkCount(line))
          len_drink = DrinkRDD.collect()
          len_of_drink = reduce(lambda x,y:x+y , len_drink)
          print "len_of_drink ----" ,len_of_drink

          len_of_drink ---- 25353
```

The output of the code is 'len_of_drink ---- 25353'.

RESULTS:

Number of Sandwiches sold = 857641

Length of Sandwiches sold = 970836 inches = 80903 feet

The number of drinks sold is 25353