# Mobivity Inc. 4/6 project for cs students

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

## Problem Definition:

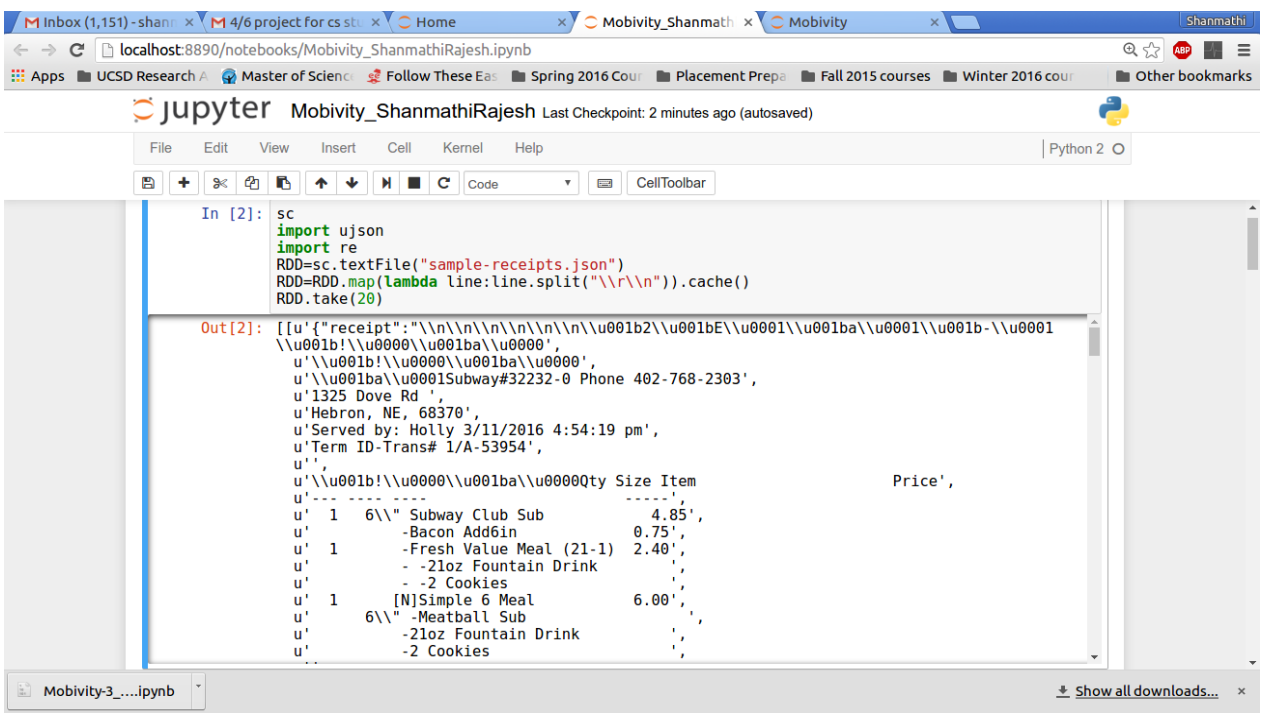
A JSON file containing 250k receipts is given. The goal is to find the number of sandwiches sold, the number of drinks sold and the total length of the sandwiches sold in feet units.

## Approach:

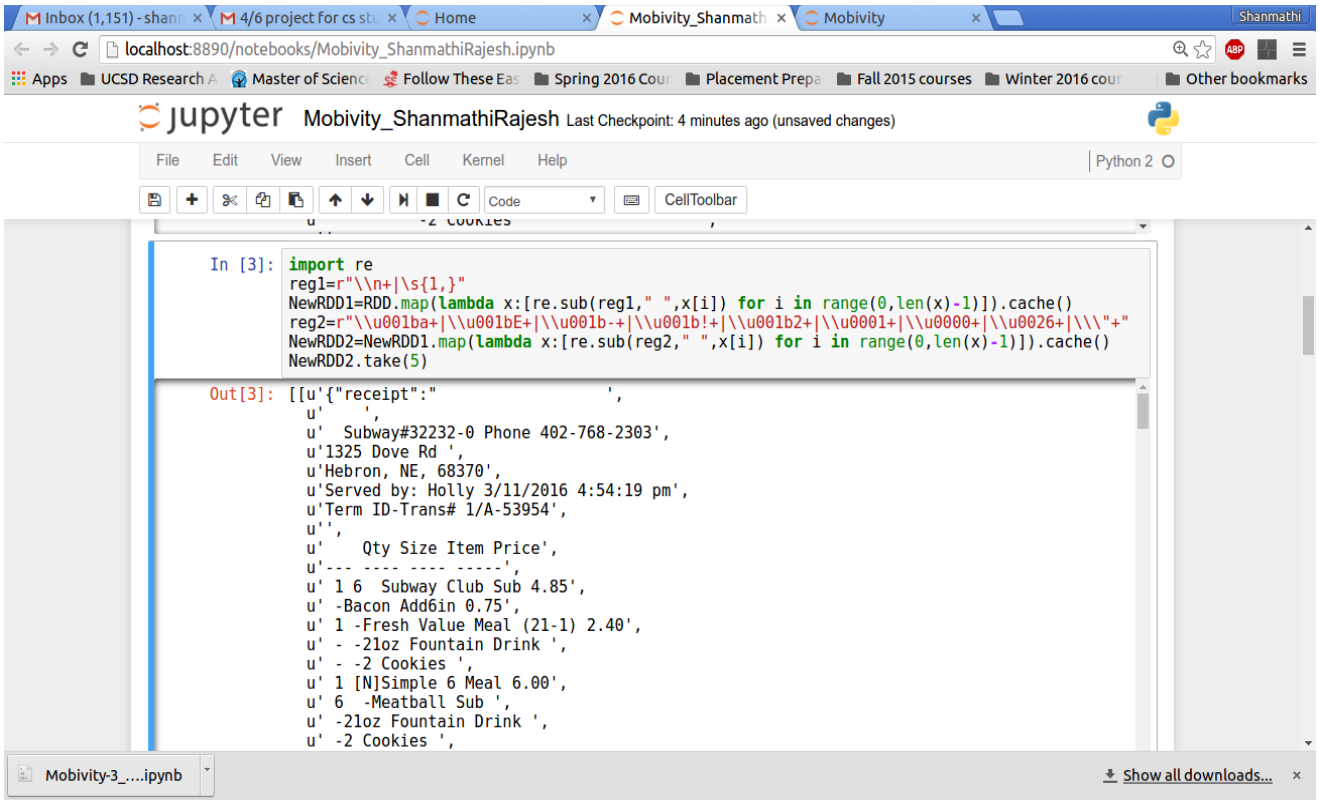
As the given problem involves mining a large dataset and extracting information from it, this problem falls under the category of "Big data Analytics". Such data mining problems can be efficiently solved in Spark environment. **Apache Spark** is an open-source big data processing framework. It's features 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.

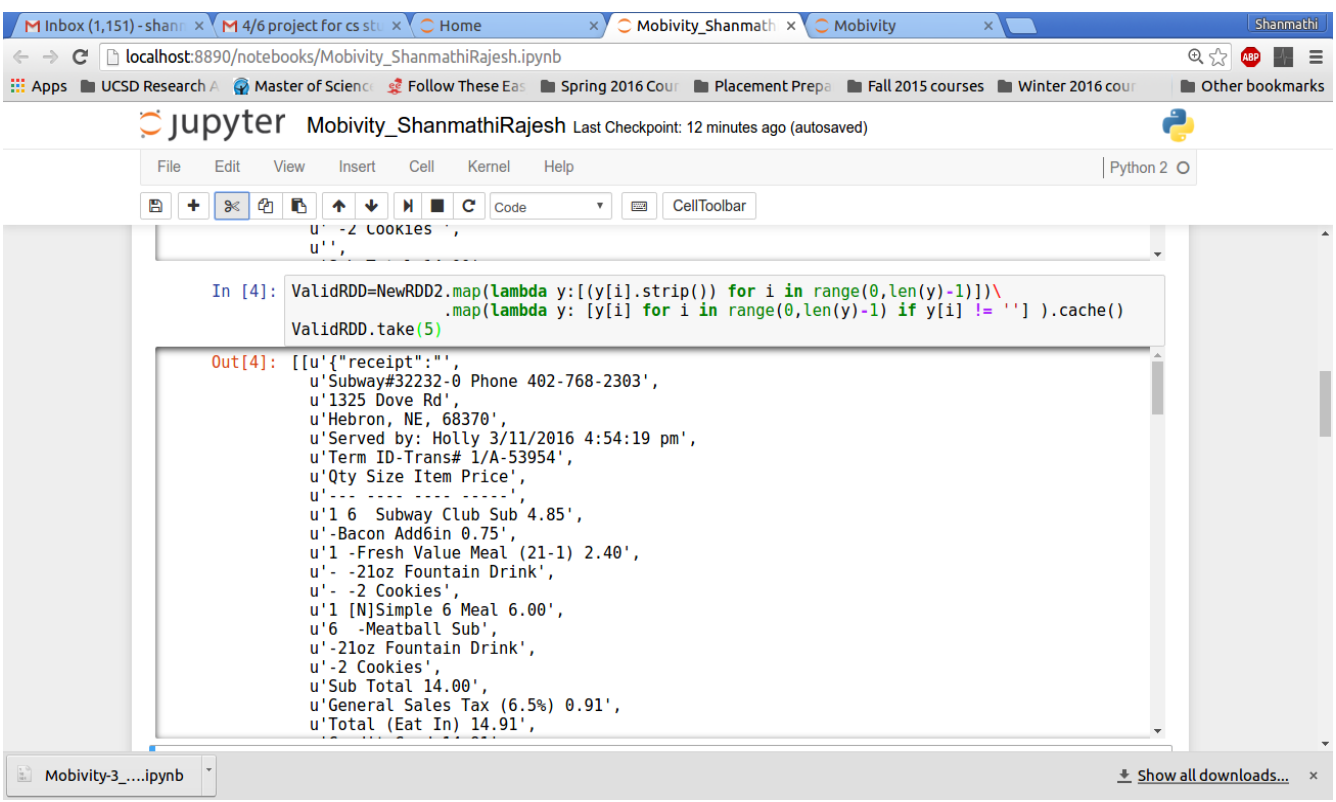
1) Reading the JSON input file and splitting the lines in the file using delimiters and storing it in RDD. An RDD is the basic data structure used in Spark. It represents an immutable, partitioned collection of elements that can be operated on in parallel.



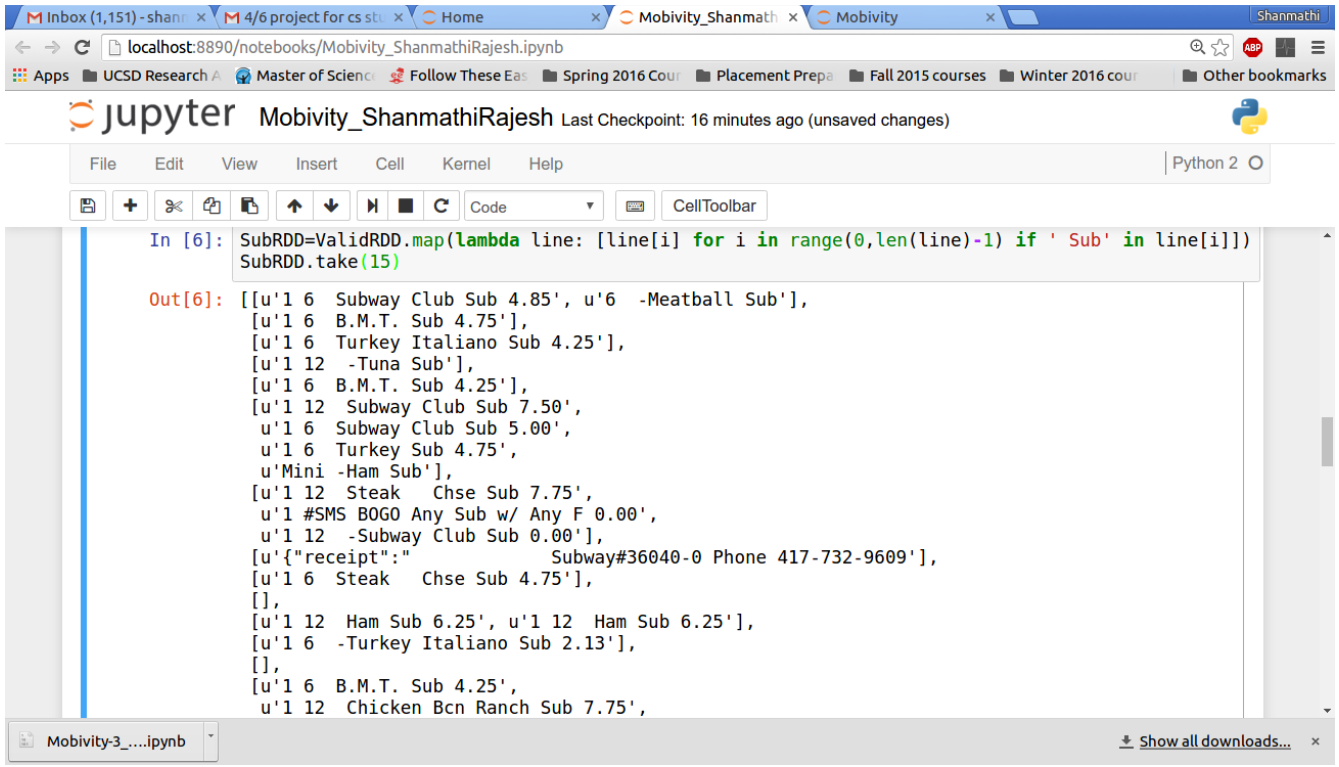
2) Formatting the lines using regular expressions to remove unnecessary spaces, escape sequences, unicode characters and storing it in NewRDD.




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.

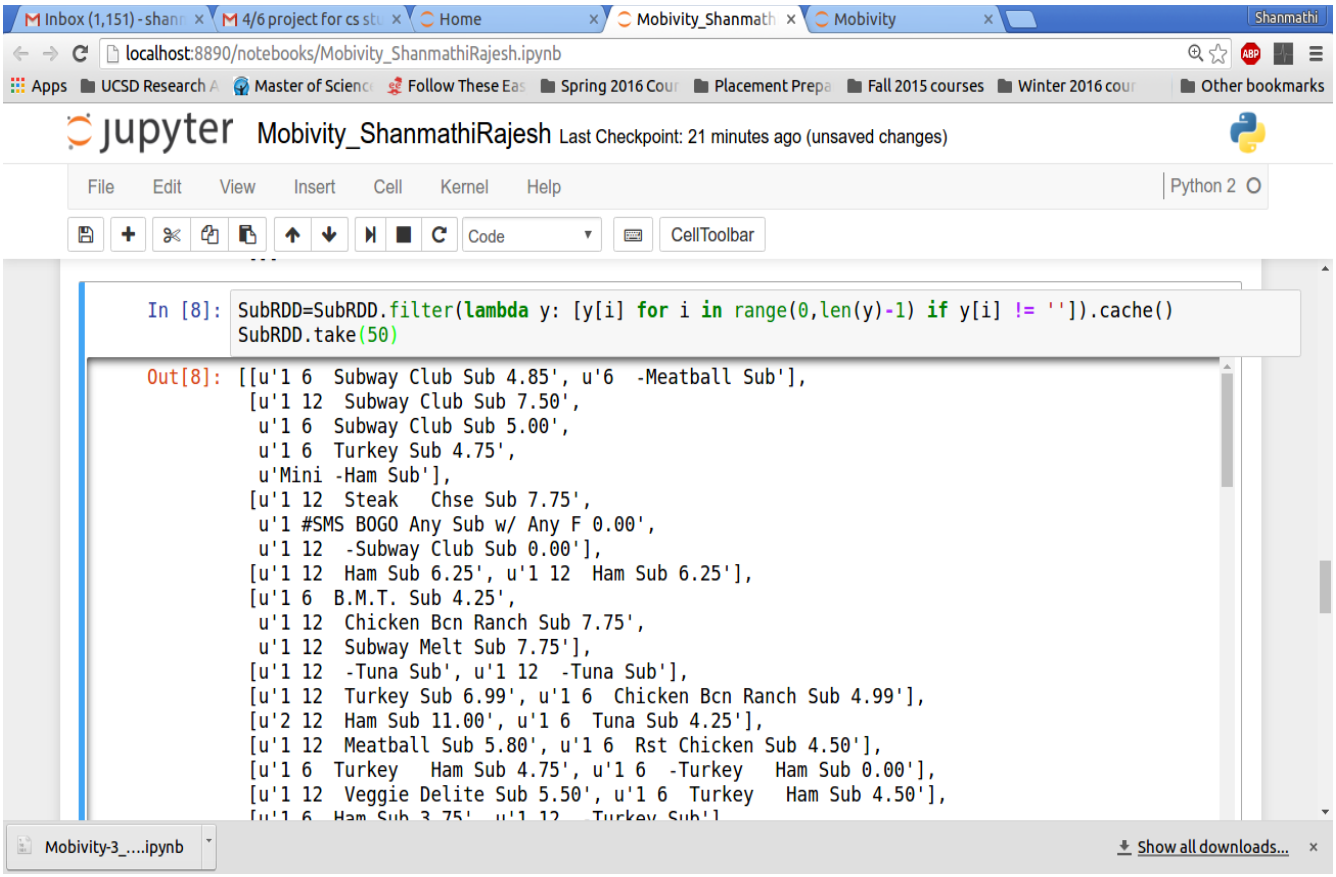


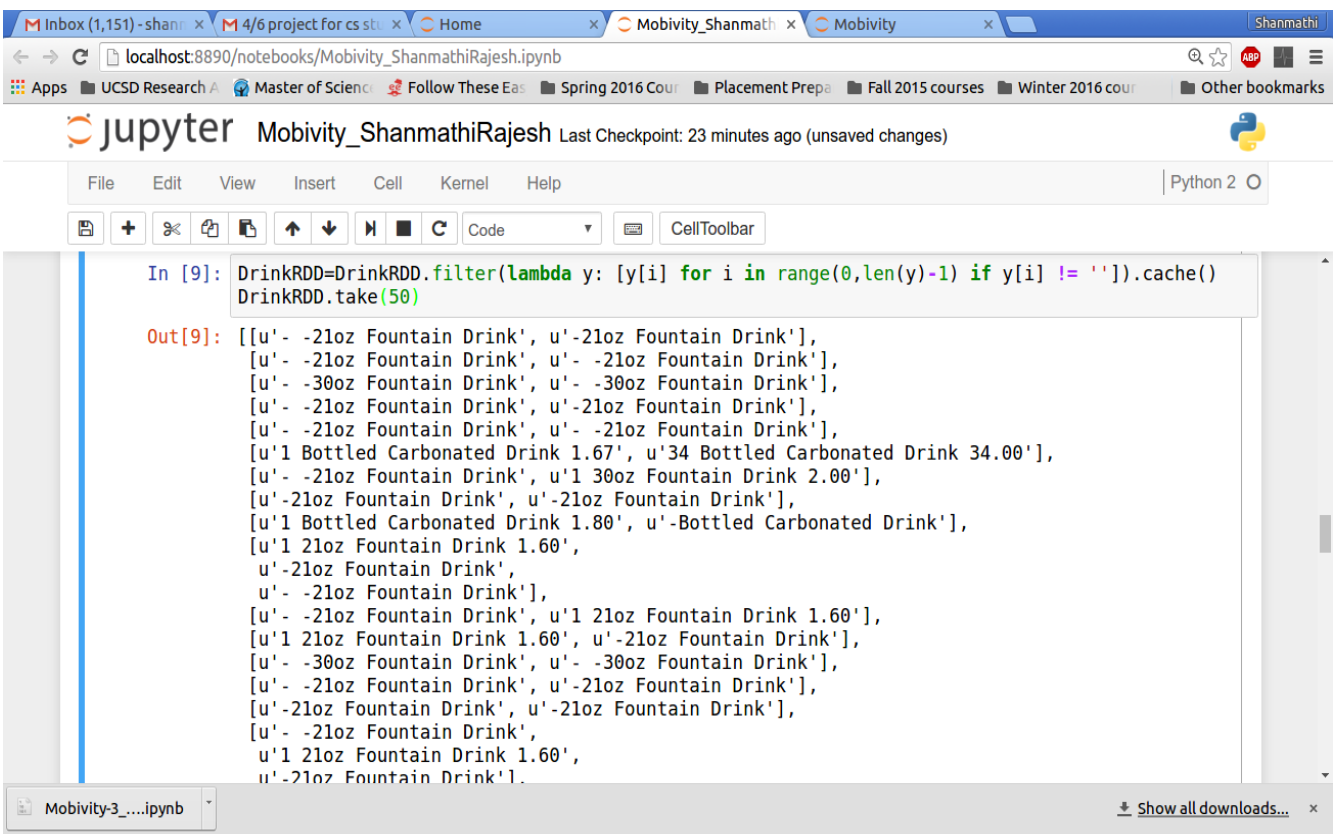
4) From each element of the ValidRDD only those lines containing the word “Sub” are filtered out so as to find the quantity and the size of each sandwich sold. This list of strings is stored in SubRDD.



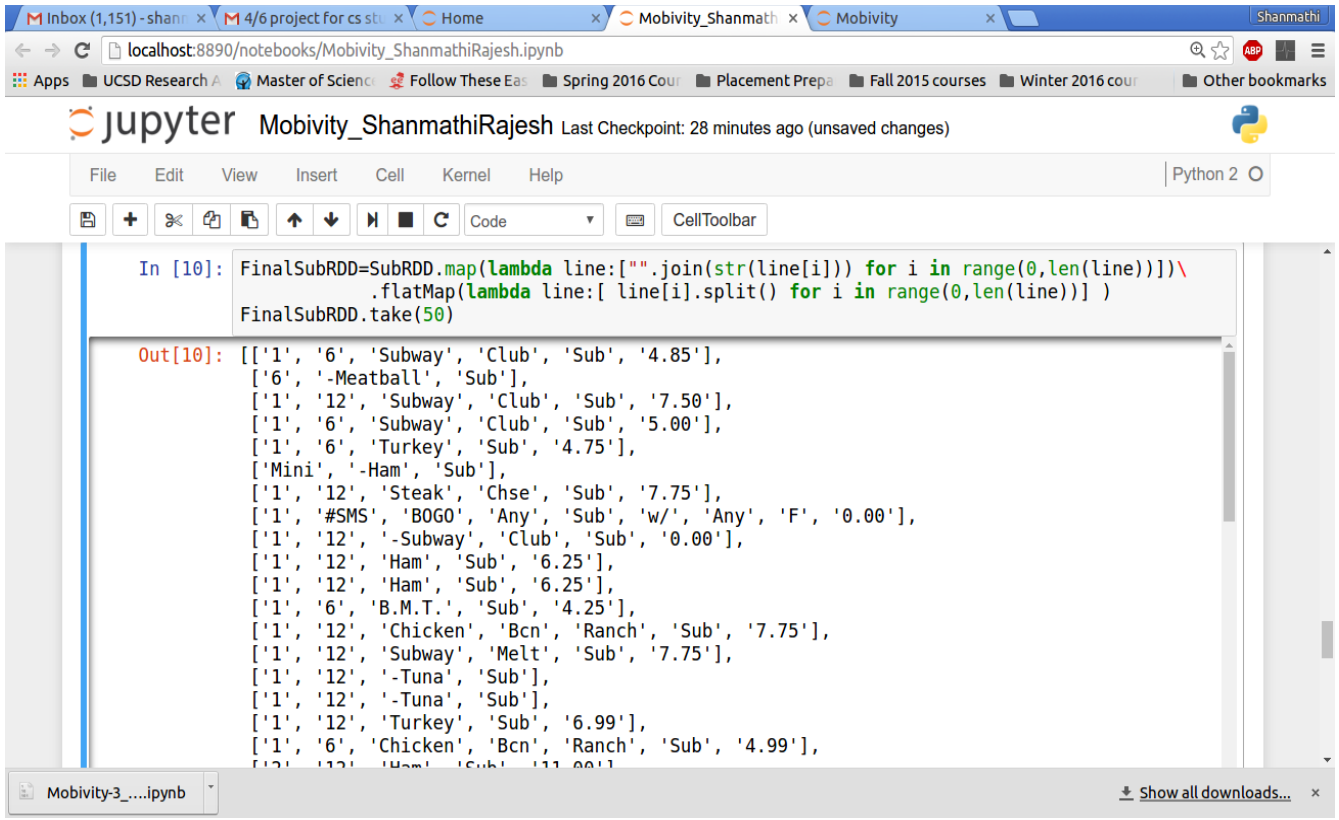

5) Similary, to find the number of drinks sold, the lines of the receipt containing the word “Drink” are filtered out separately and stored in DrinkRDD.

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

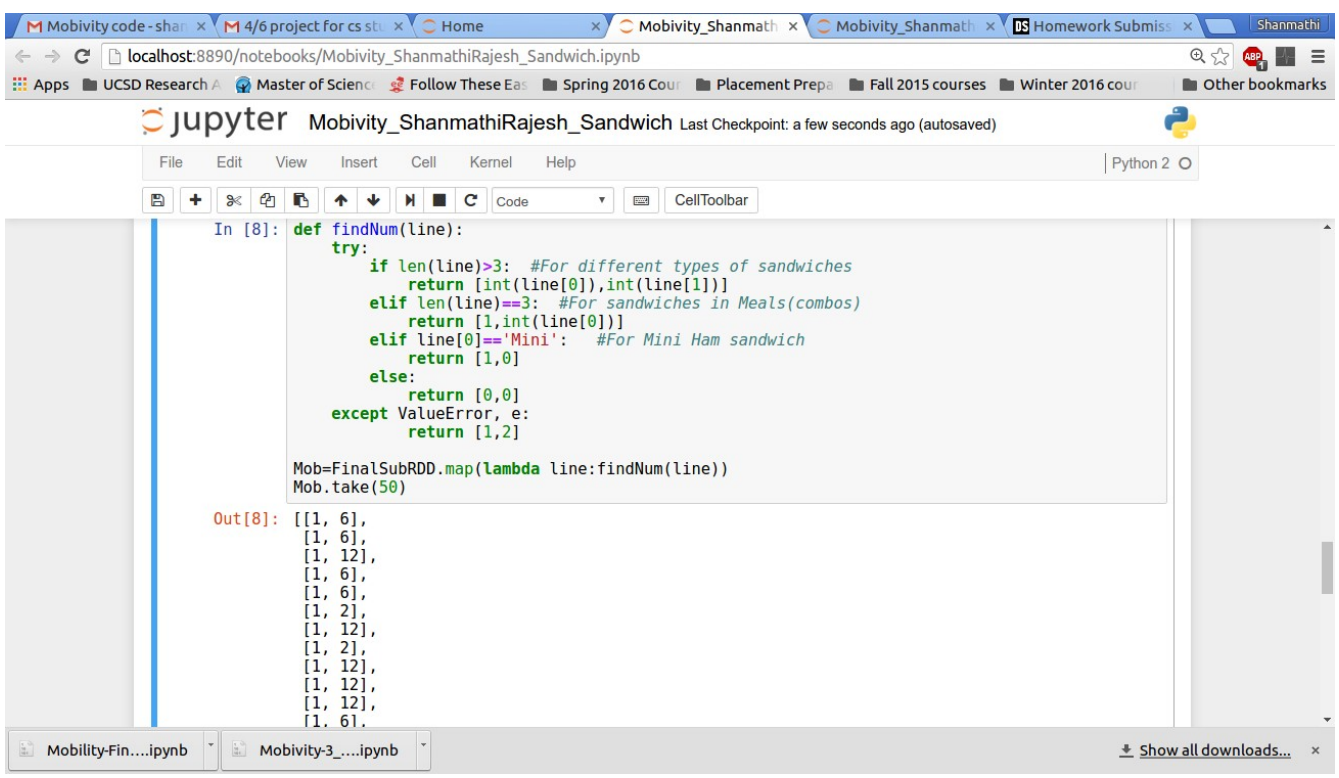


7) The elements of SubRDD are joined together and converted from unicode strings to normal strings and stored in FinalSubRDD. In the FinalSubRDD list, the first column contains the quantity of sandwich sold ans the second column contains the size of the sandwich sold.




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

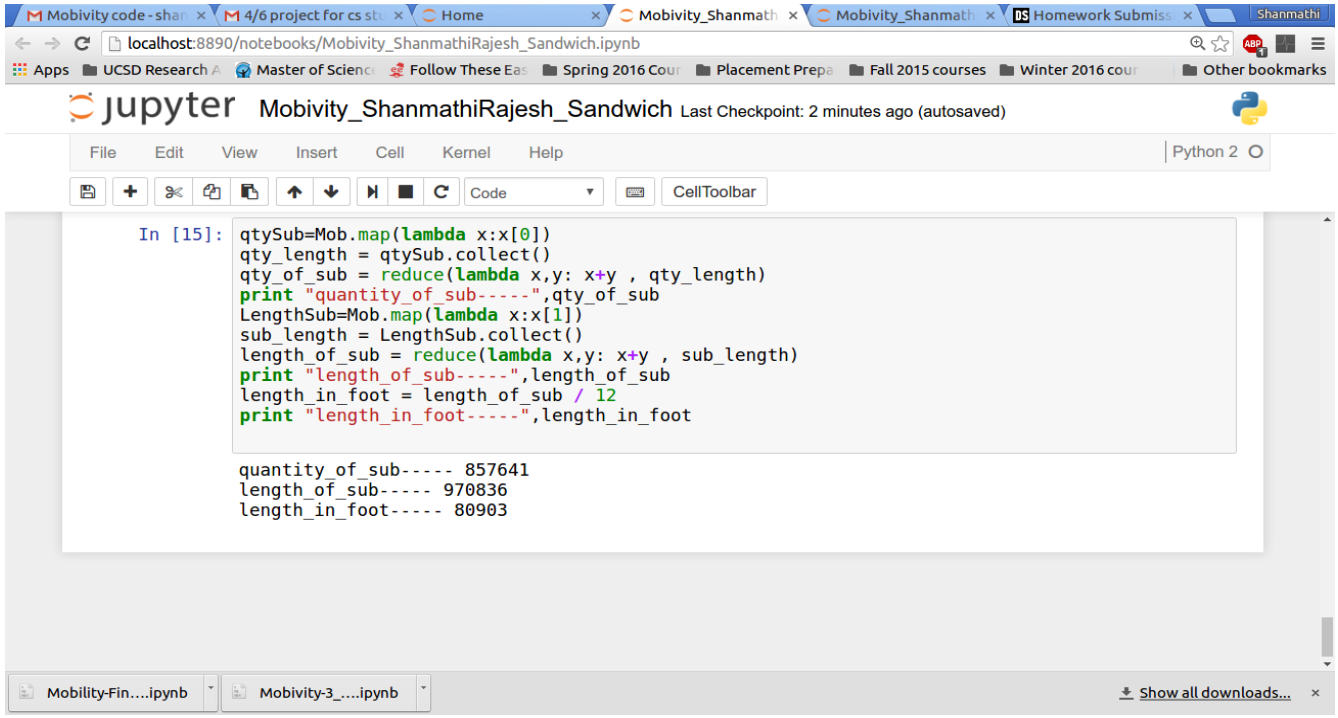


9) All the quantities are collected as a separate list in qty_length and all the values are added using reduce function and stored in qty_of_sub.

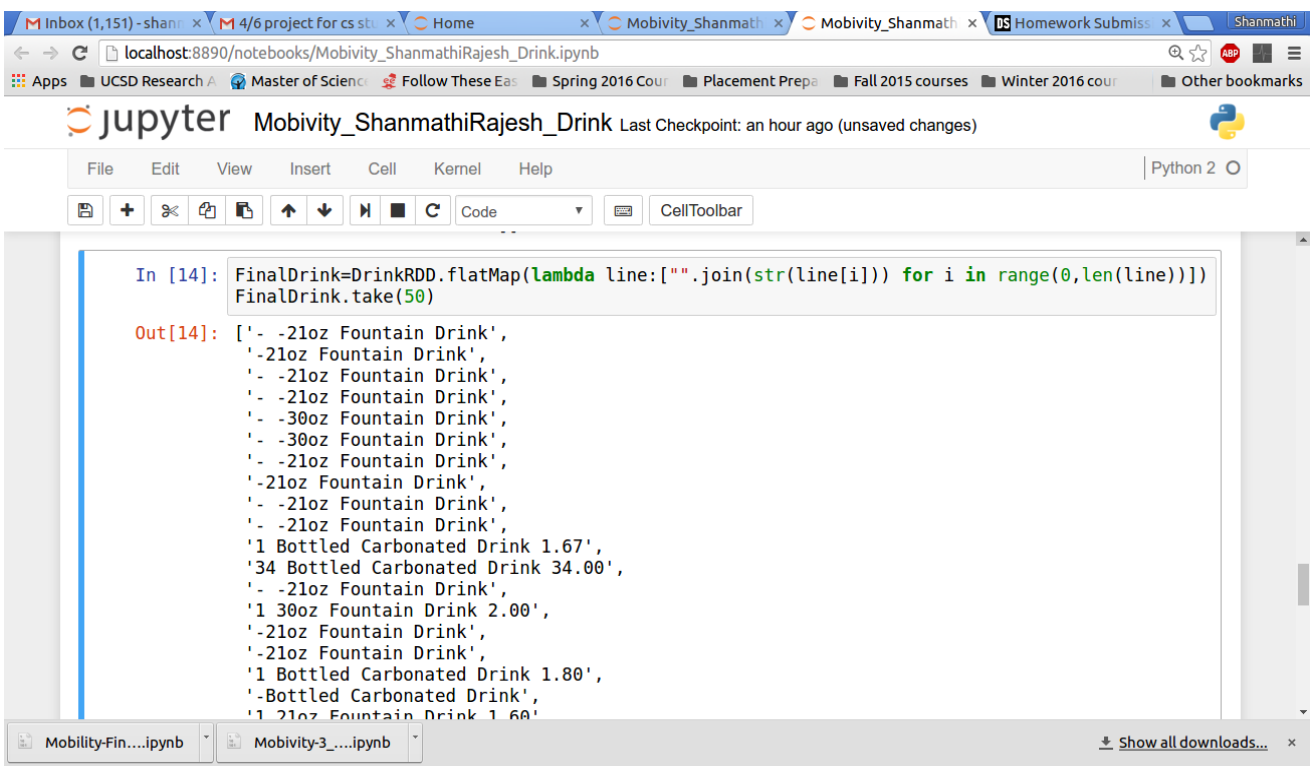
**Number of Sandwiches sold = 857641**

Similarly, the length of the sandwiches are added and stored in length_of_sub. To calculate in feet this value is divided by 12 (As 12 inch = 1 foot).

**Length of Sandwiches sold = 970836 inches = 80903 feet**

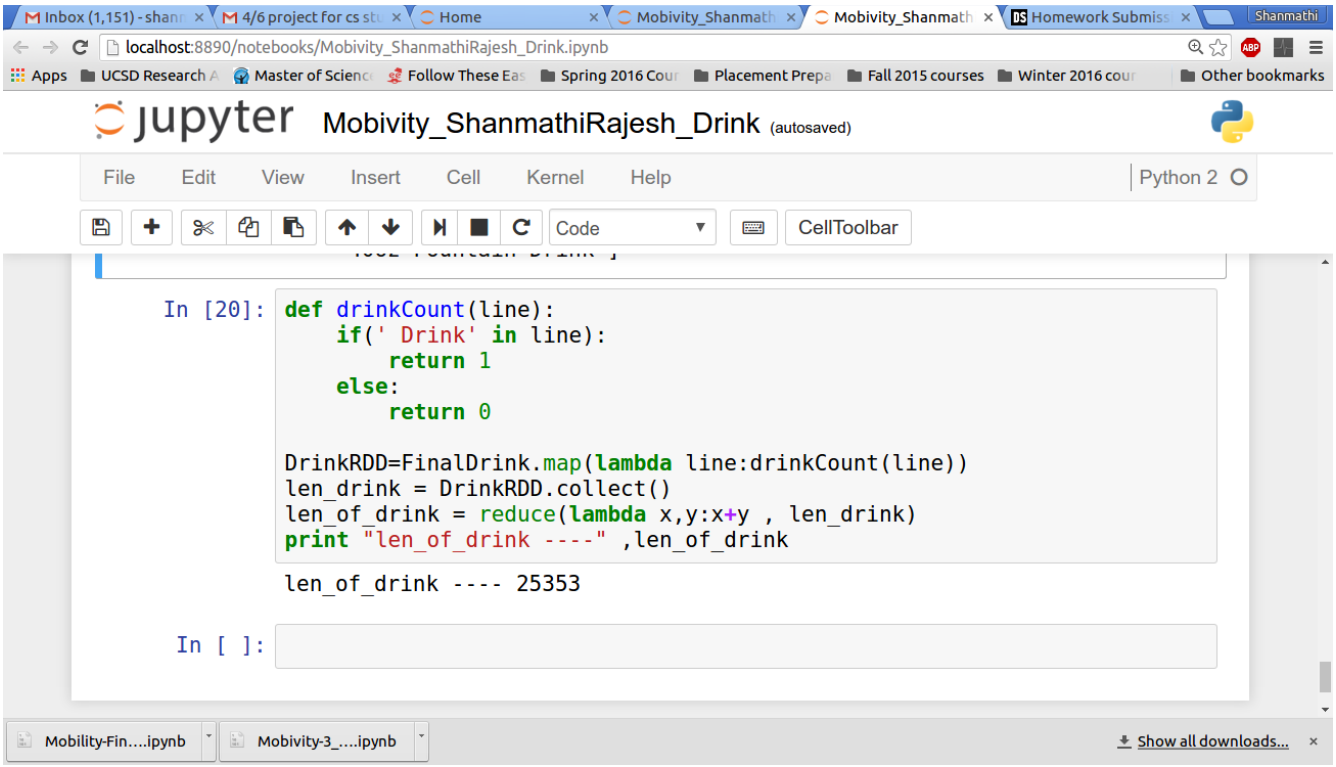



10) The elements of DrinkRDD are joined together as strings and stored in a RDD called FinalDrink which contains each drink as an element in the list.



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**




# RESULTS:

**Number of Sandwiches sold = 857641**

**Length of Sandwiches sold = 970836 inches = 80903 feet**

**The number of drinks sold is 25353**