Steps for running

$SPARK_HOME/bin/spark-submit Devansh_Sharma_SON.py <input_file> <support_threshold>

Spark version: 2.3.1
Python version: 2.7

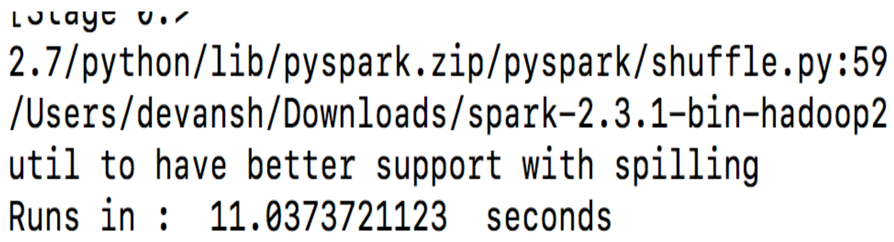
Method Used:

- According to the question, I am using the SON algorithm. As SON applies naturally to the Map-Reduce method, I am taking advantage of that to find the frequent itemsets, by first finding the candidate itemsets.
- I have used A-priori Algorithm to find the frequent itemsets of sizes 1, 2, … so on, for each chunk or partition, during the first Map.
- Then, for the map, I am applying reduce, such that (itemset, 1) pairs are formed
- For the second map, each map task gives the number of occurrences of each of the candidate itemsets among the baskets in the portion of the dataset that it was assigned.
- Based on the output of the second reduce function, the candidate items which are below the threshold and their counts are returned

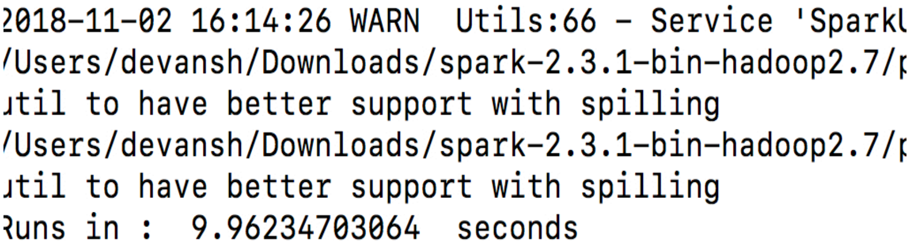
Problem 1:

Support Threshold	Execution Time
30	9.96234703064 seconds
40	11.0373721123 seconds

Screenshots:



```
[Stage 0:>
2.7/python/lib/pyspark.zip/pyspark/shuffle.py:59
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2
util to have better support with spilling
Runs in :  11.0373721123  seconds
```

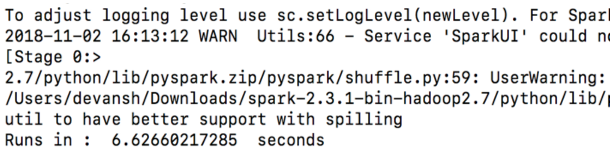


```
2018-11-02 16:14:26 WARN  Utils:66 - Service 'Spark
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/
util to have better support with spilling
Runs in :  9.96234703064  seconds
```

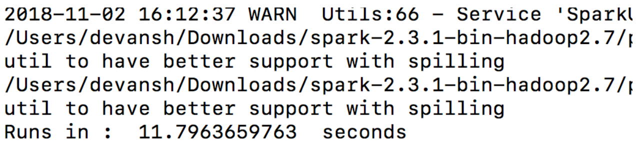
Problem 2:

Support Threshold	Execution Time
500	11.7963659763 seconds
1000	6.62660217285 seconds

Screenshots:



```
To adjust logging level use sc.setLogLevel(newLevel). For Spar
2018-11-02 16:13:12 WARN  Utils:66 - Service 'SparkUI' could n
[Stage 0:>
2.7/python/lib/pyspark.zip/pyspark/shuffle.py:59: UserWarning:
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/python/lib/
util to have better support with spilling
Runs in :  6.62660217285  seconds
```

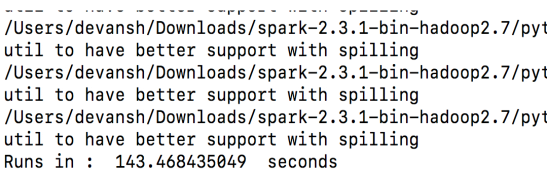


```
2018-11-02 16:12:37 WARN  Utils:66 - Service 'Spark
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/
util to have better support with spilling
Runs in :  11.7963659763  seconds
```

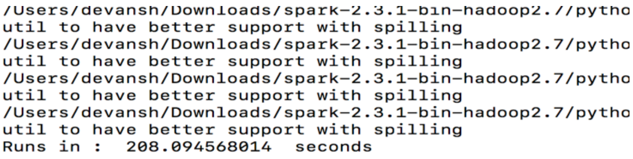
Problem 3:

Support Threshold	Execution Time
100000	208.094568014 seconds
120000	143.468435049 seconds

Screenshots:



```
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pyt
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pyt
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pyt
util to have better support with spilling
Runs in :  143.468435049  seconds
```



```
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pytho
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pytho
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pytho
util to have better support with spilling
/Users/devansh/Downloads/spark-2.3.1-bin-hadoop2.7/pytho
util to have better support with spilling
Runs in :  208.094568014  seconds
```

Bonus (5pts): Describe why did we need to use such a large support threshold and where do you think there could be a bottleneck that could result in a slow execution for your implementation, if any.

We need such a large support threshold because if we have a low threshold, the dataset being so large, the number of candidate itemsets will increase exponentially. The larger the threshold, the smaller candidate itemsets we have to consider in the next Map step.

The second Map function mentioned above in the Method Used will prove to be a bottleneck, because if the threshold is small, the Map function will be applied to many more candidate itemsets and time taken for the step as well as the next Reduce step will exponentially increase.