Brute Force,Forward & Backward Feature Selection Algorithm

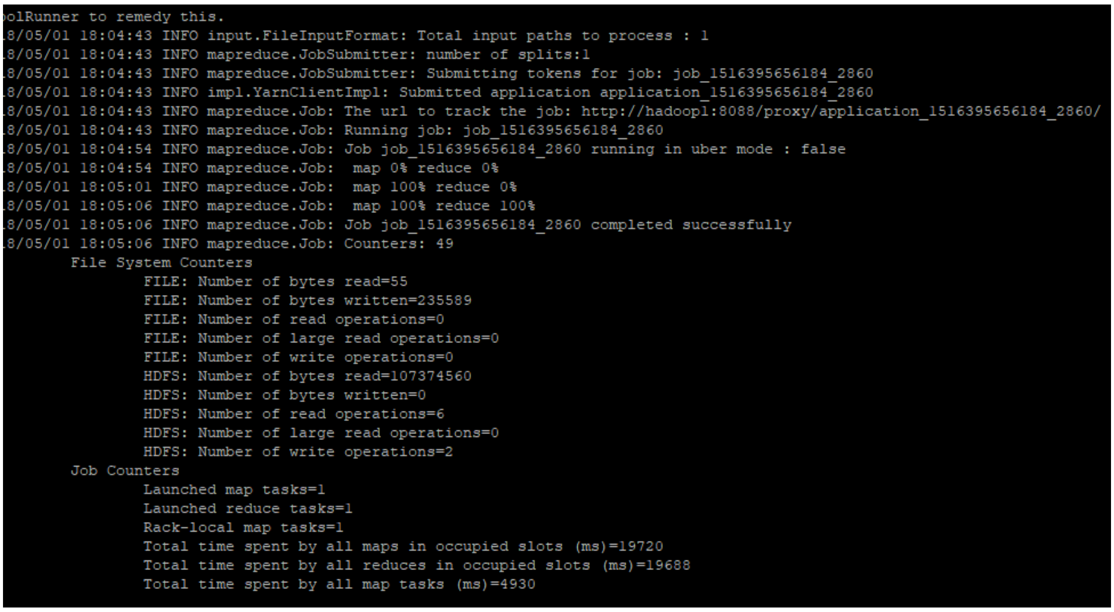
Version	1.0
Document Title	Final Report of Big Data Group Project
Group Members	Tanuj Gyan, Sayanti Ray, Raviprakash Shreeraksha, Babak Shabani
Date of Submission	1-May-2018
Instructor Name	Dr. Johnson P Thomas
TA Name	Vasanthi Mudunuri

Brute Force Feature Selection Algorithm:

This is a trial and error approach where we consider all possible subset of features and find the accuracy of these subsets with the help of Machine Learning Algorithms. All Machine Learning Algorithms used are supervised machine learning algorithms and so we need to have both Test and Train data to calculate accuracy.

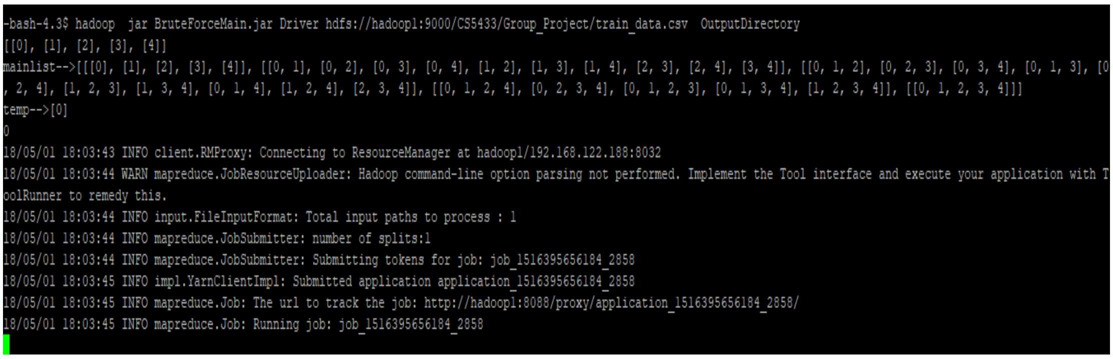
So we have used Map Reduce of Hadoop to implement Brute Force Algorithm.First we will consider the number of features given and then calculate all possible subsets of these features,avoiding the redundancy of the subsets.In the Mapper, we have taken each subset of features and then in the reducer we output the subset.

Below are the snapshots of Map Reduce program implementing Brute Force Algorithm.



```
olRunner to remedy this.
8/05/01 18:04:43 INFO input.FileInputFormat: Total input paths to process : 1
8/05/01 18:04:43 INFO mapreduce.JobSubmitter: number of splits:1
8/05/01 18:04:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1516395656184_2860
8/05/01 18:04:43 INFO impl.YarnClientImpl: Submitted application application_1516395656184_2860
8/05/01 18:04:43 INFO mapreduce.Job: The url to track the job: http://hadoop1:8088/proxy/application_1516395656184_2860/
8/05/01 18:04:43 INFO mapreduce.Job: Running job: job_1516395656184_2860
8/05/01 18:04:54 INFO mapreduce.Job: Job job_1516395656184_2860 running in uber mode : false
8/05/01 18:04:54 INFO mapreduce.Job:  map 0% reduce 0%
8/05/01 18:05:01 INFO mapreduce.Job:  map 100% reduce 0%
8/05/01 18:05:06 INFO mapreduce.Job:  map 100% reduce 100%
8/05/01 18:05:06 INFO mapreduce.Job: Job job_1516395656184_2860 completed successfully
8/05/01 18:05:06 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=55
                FILE: Number of bytes written=235589
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=107374560
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=19720
                Total time spent by all reduces in occupied slots (ms)=19688
                Total time spent by all map tasks (ms)=4930
```

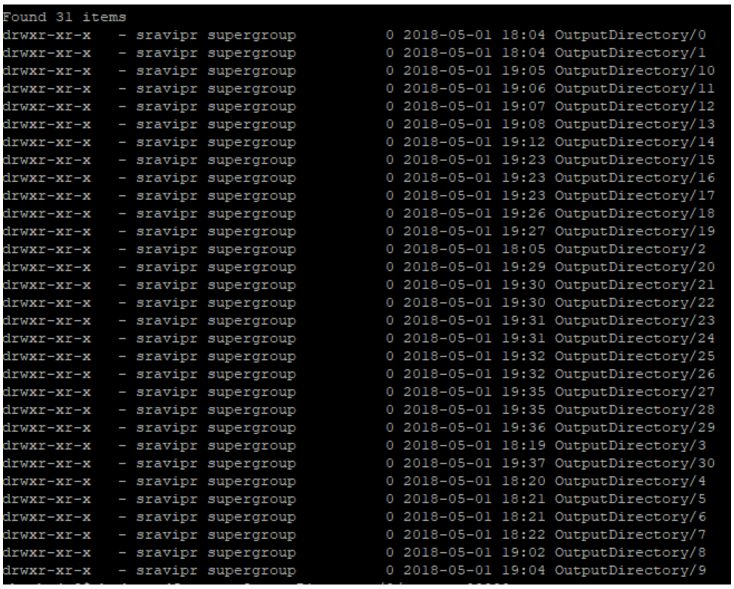
Fig 1: Map Reduce program in execution



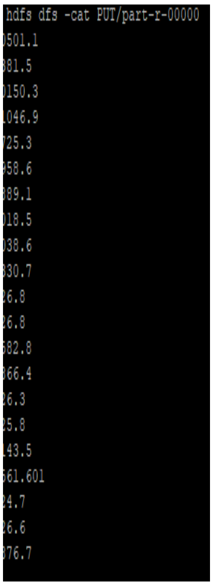
```
-bash-4.3$ hadoop  jar BruteForceMain.jar Driver hdfs://hadoop1:9000/CS5433/Group_Project/train_data.csv  OutputDirectory
[[0], [1], [2], [3], [4]]
mainlist-->[[[0], [1], [2], [3], [4]], [[0, 1], [0, 2], [0, 3], [0, 4], [1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]], [[0, 1, 2], [0, 2, 3], [0, 3, 4], [0, 1, 3], [0
, 2, 4], [1, 2, 3], [1, 3, 4], [0, 1, 4], [1, 2, 4], [2, 3, 4]], [[0, 1, 2, 4], [0, 2, 3, 4], [0, 1, 2, 3], [0, 1, 3, 4], [1, 2, 3, 4]], [[0, 1, 2, 3, 4]]]
temp-->[0]
0
18/05/01 18:03:43 INFO client.RMProxy: Connecting to ResourceManager at hadoop1/192.168.122.188:8032
18/05/01 18:03:44 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with T
oolRunner to remedy this.
18/05/01 18:03:44 INFO input.FileInputFormat: Total input paths to process : 1
18/05/01 18:03:44 INFO mapreduce.JobSubmitter: number of splits:1
18/05/01 18:03:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1516395656184_2858
18/05/01 18:03:45 INFO impl.YarnClientImpl: Submitted application application_1516395656184_2858
18/05/01 18:03:45 INFO mapreduce.Job: The url to track the job: http://hadoop1:8088/proxy/application_1516395656184_2858/
18/05/01 18:03:45 INFO mapreduce.Job: Running job: job_1516395656184_2858
```

Fig 2: Creation of Subsets

Above screenshots display the execution of MapReduce program and creation of subsets for 5 features.



```
Found 31 items
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:04 OutputDirectory/0
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:04 OutputDirectory/1
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:05 OutputDirectory/10
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:06 OutputDirectory/11
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:07 OutputDirectory/12
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:08 OutputDirectory/13
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:12 OutputDirectory/14
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:23 OutputDirectory/15
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:23 OutputDirectory/16
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:23 OutputDirectory/17
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:26 OutputDirectory/18
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:27 OutputDirectory/19
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:05 OutputDirectory/2
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:29 OutputDirectory/20
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:30 OutputDirectory/21
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:30 OutputDirectory/22
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:31 OutputDirectory/23
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:31 OutputDirectory/24
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:32 OutputDirectory/25
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:32 OutputDirectory/26
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:35 OutputDirectory/27
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:35 OutputDirectory/28
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:36 OutputDirectory/29
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:19 OutputDirectory/3
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:37 OutputDirectory/30
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:20 OutputDirectory/4
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:21 OutputDirectory/5
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:21 OutputDirectory/6
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 18:22 OutputDirectory/7
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:02 OutputDirectory/8
drwxr-xr-x   - sravipr supergroup          0 2018-05-01 19:04 OutputDirectory/9
```

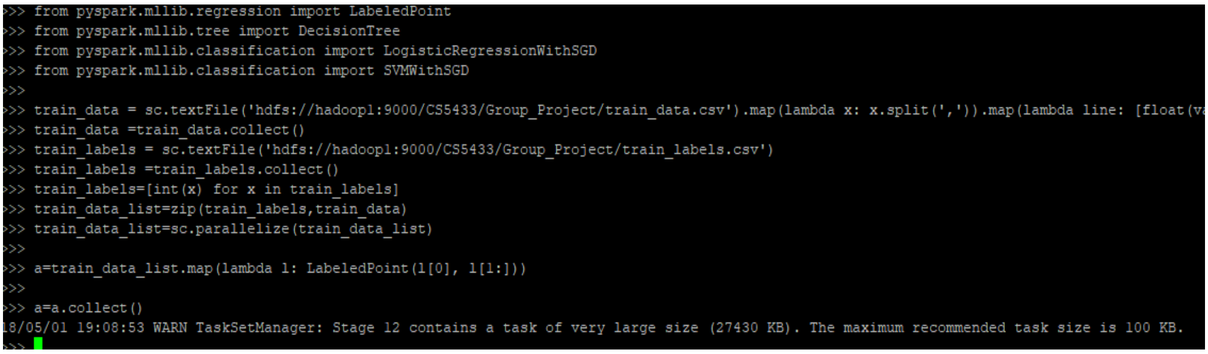


```
hdfs dfs -cat PUT/part-r-00000
501.1
81.5
150.3
046.9
25.3
58.6
89.1
18.5
38.6
30.7
6.8
6.8
82.8
66.4
6.3
5.8
43.5
61.601
4.7
6.6
76.7
```

Fig 3: Output

Challenges Faced :

- There are total 760 features and we had to create all possible subsets ,but the data is large and it was not feasible to run Brute Force algorithm as the system would crash.
- When implementing Decision tree ,after Stage 8 or 12 the data contained for that task was very large and exceeded the task size.Please refer the below snapshot.



```
>>> from pyspark.mllib.regression import LabeledPoint
>>> from pyspark.mllib.tree import DecisionTree
>>> from pyspark.mllib.classification import LogisticRegressionWithSGD
>>> from pyspark.mllib.classification import SVMWithSGD
>>>
>>> train_data = sc.textFile('hdfs://hadoop1:9000/CS5433/Group_Project/train_data.csv').map(lambda x: x.split(',')).map(lambda line: [float(va
>>> train_data =train_data.collect()
>>> train_labels = sc.textFile('hdfs://hadoop1:9000/CS5433/Group_Project/train_labels.csv')
>>> train_labels =train_labels.collect()
>>> train_labels=[int(x) for x in train_labels]
>>> train_data_list=zip(train_labels,train_data)
>>> train_data_list=sc.parallelize(train_data_list)
>>>
>>> a=train_data_list.map(lambda l: LabeledPoint(l[0], l[1:]))
>>>
>>> a=a.collect()
18/05/01 19:08:53 WARN TaskSetManager: Stage 12 contains a task of very large size (27430 KB). The maximum recommended task size is 100 KB.
>>>
```

Forward Algorithm :

Initially the feature set is subjected to be empty and then in FFS we keep adding the features as we progress.

Here we used a threshold of 0.03 to maintain the accuracy within a limit, without that value obtained from subset of features will give the maximum accuracy which is equivalent to the original accuracy of the machine learning algorithm.

Backward Algorithm:

Initially the feature set is subjected to be full and then in BFS we keep subtracting the features as we progress.

Both Forward and Backward Algorithm use feature importance to select the important features in the feature set.In the Forward algorithm we keep on adding the important feature based on feature importance and in the Backward algorithm we keep on removing the unimportant feature.