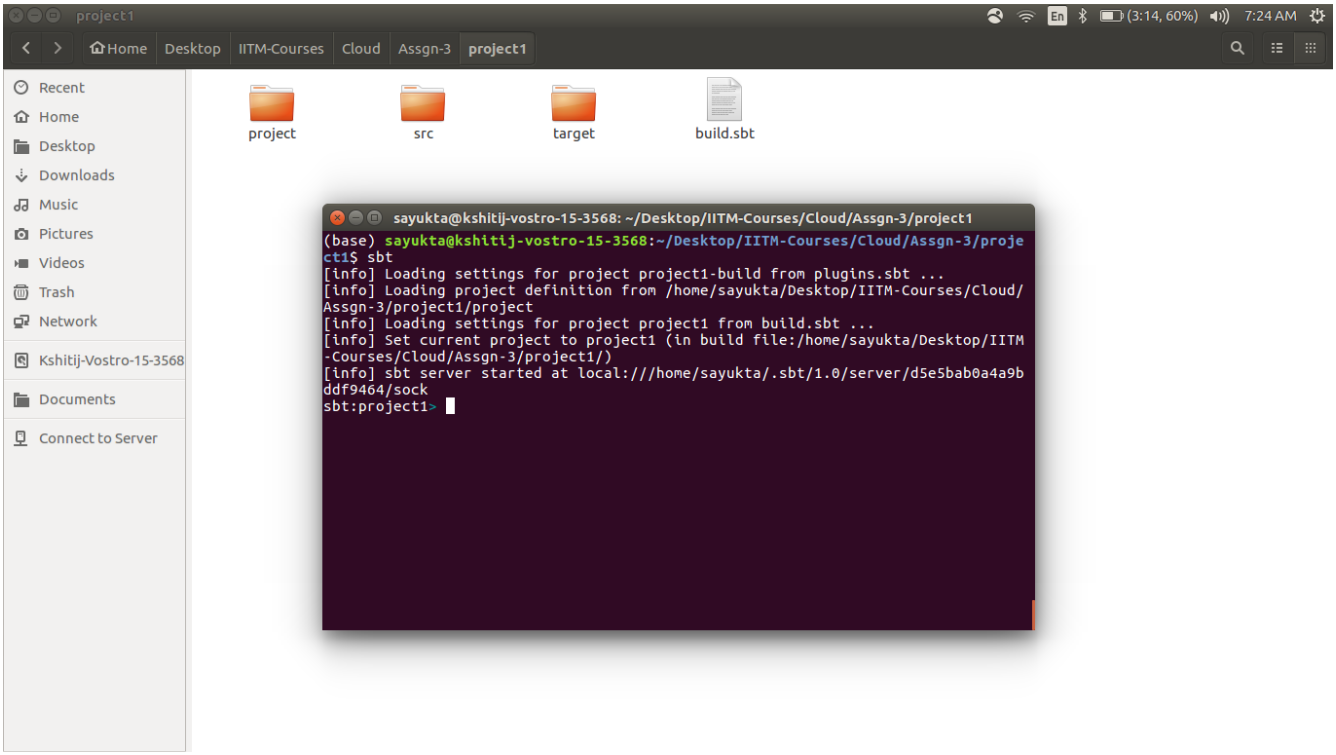
1. ALS:

Setup:

I downloaded Scala-2.13.1 and Spark-3.0.0-preview2. I decided to do this assignment using sbt.
I made a folder called ‘project1’ and made folders: ‘src’, ‘main’ folder inside ‘src’, ‘scala’ folder inside ‘main’. Inside ‘scala’ folder, would be my MovieLensALS.scala in case of ALS, and FPGrowth.scala in case of FPGrowth. I copied source code from mllib of Spark.
Next, with ‘project1’ as root, I opened a terminal and typed sbt.



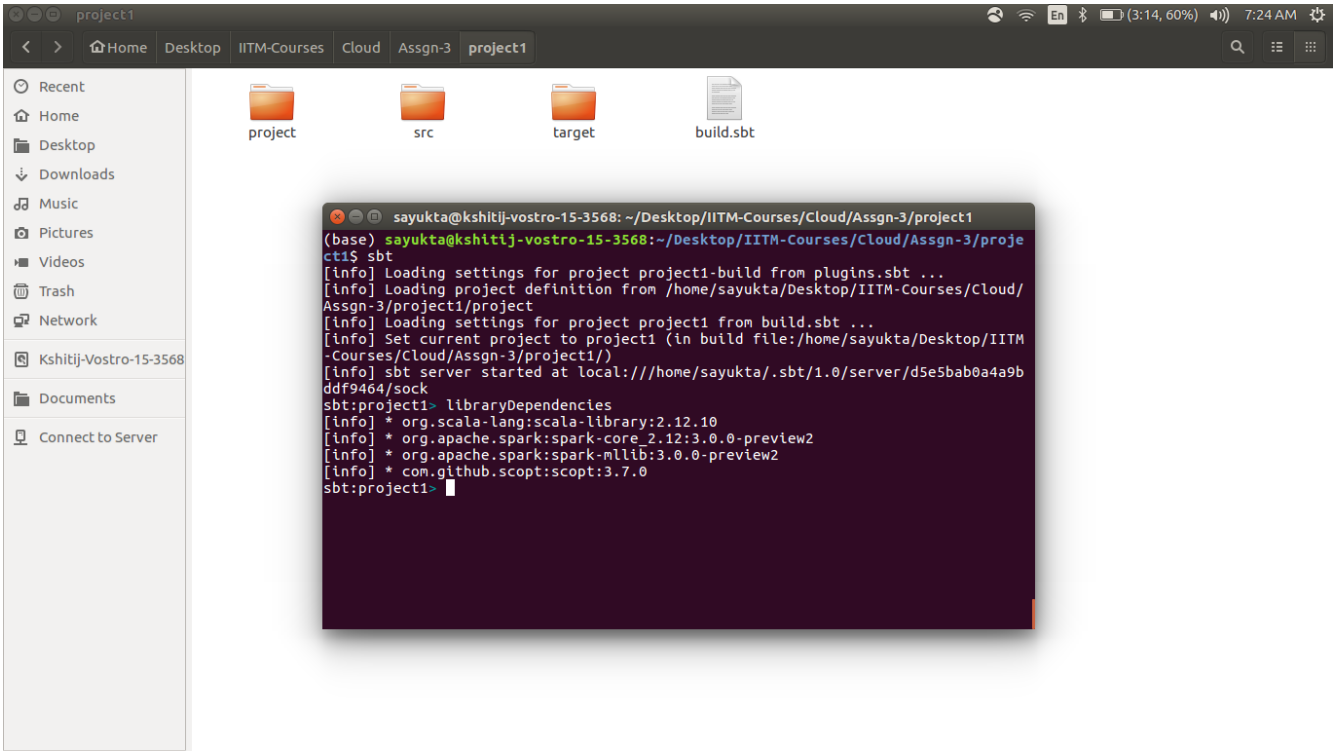
Next, I needed to set the ‘library dependencies’ so that my build.sbt could be built and my code compiled.

I used the libraries:
org.apache.spark:spark-core_2.12_3.0.0-preview2, org.apache.spark:spark-mllib:3.0.0-preview2, com.github.scopt:scopt:3.7.0

Here, 2.12 is the scala version I am using, and 3.0.0-preview2 is the spark version. Central repository for downloading required JAR files corresponding to scala and spark version specified in build.sbt is "Maven".

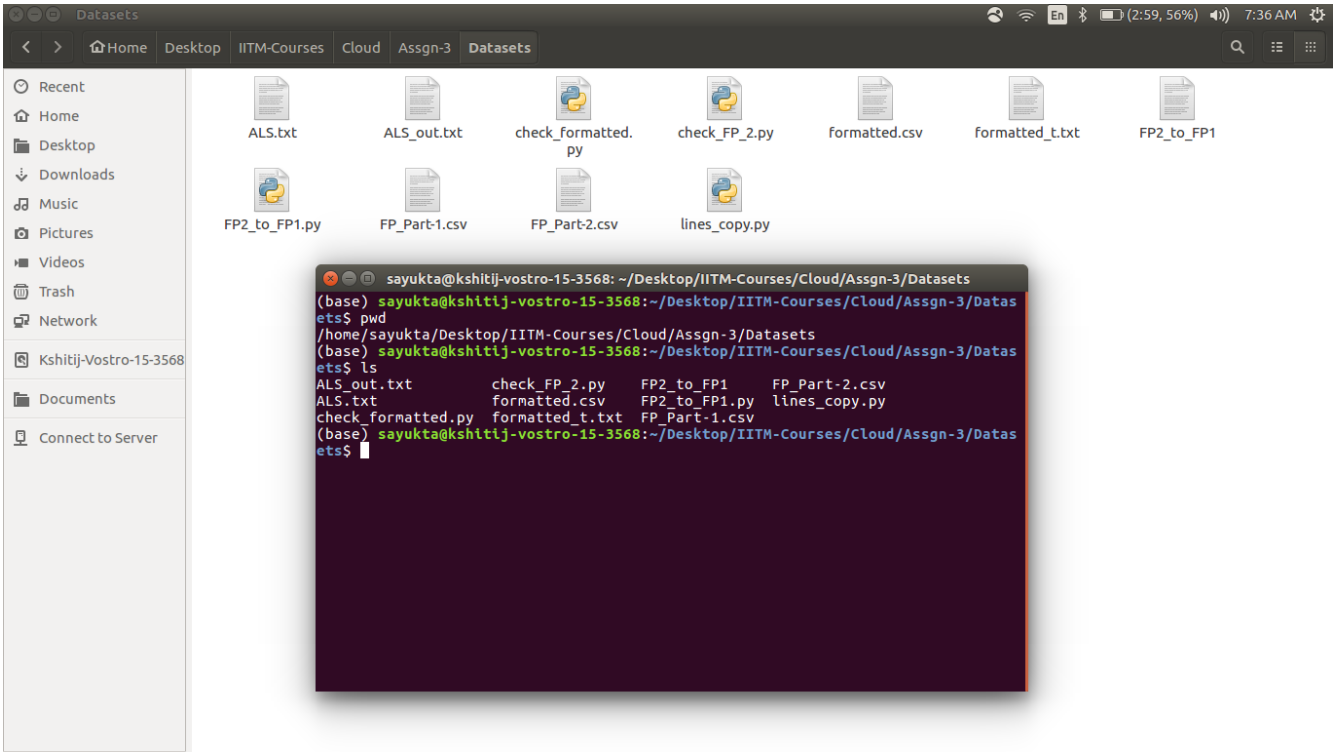
When you develop a Spark project using Scala language, you have to package your project into a jar file.
But I am running my compiled program in sbt itself, and not using ‘assemble’ to create a JAR file of my ALS/FPGrowth program and later running it via spark-submit.
In sbt terrminal, type: ‘session save’, and then ‘reload’ to apply your changes to

build.sbt.



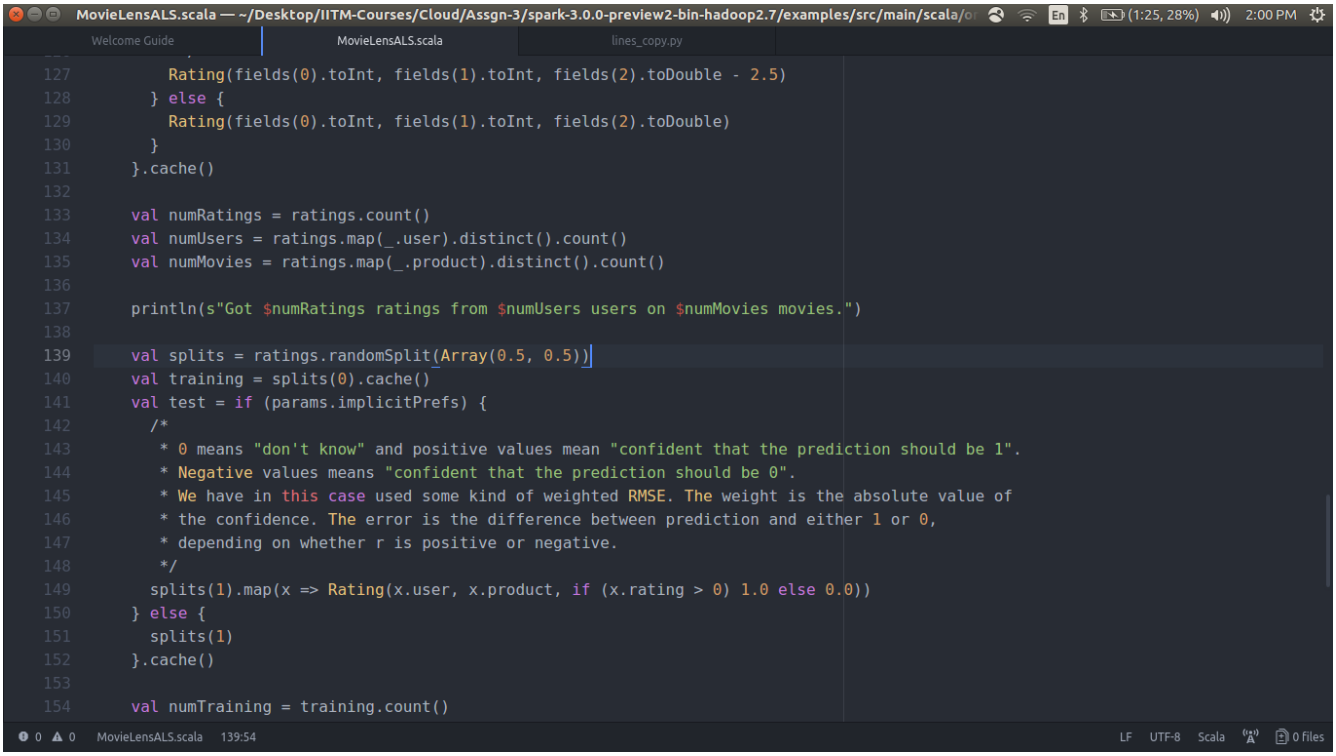
The data sets I got from Archive.zip of Question are here:

(So ‘project1’ and ‘Datasets’ are 2 different folders in Assgn-3 folder that can be seen in terminal)



Type ‘compile’ in sbt terminal to compile MovieLensALS.scala.

By default in MovieLensALS.scala, ratio of training data size: test data size is 80:20, rank=5, regularization parameter(lambda)=1.
To change ratio of training data size: test data size, change line 139 here:

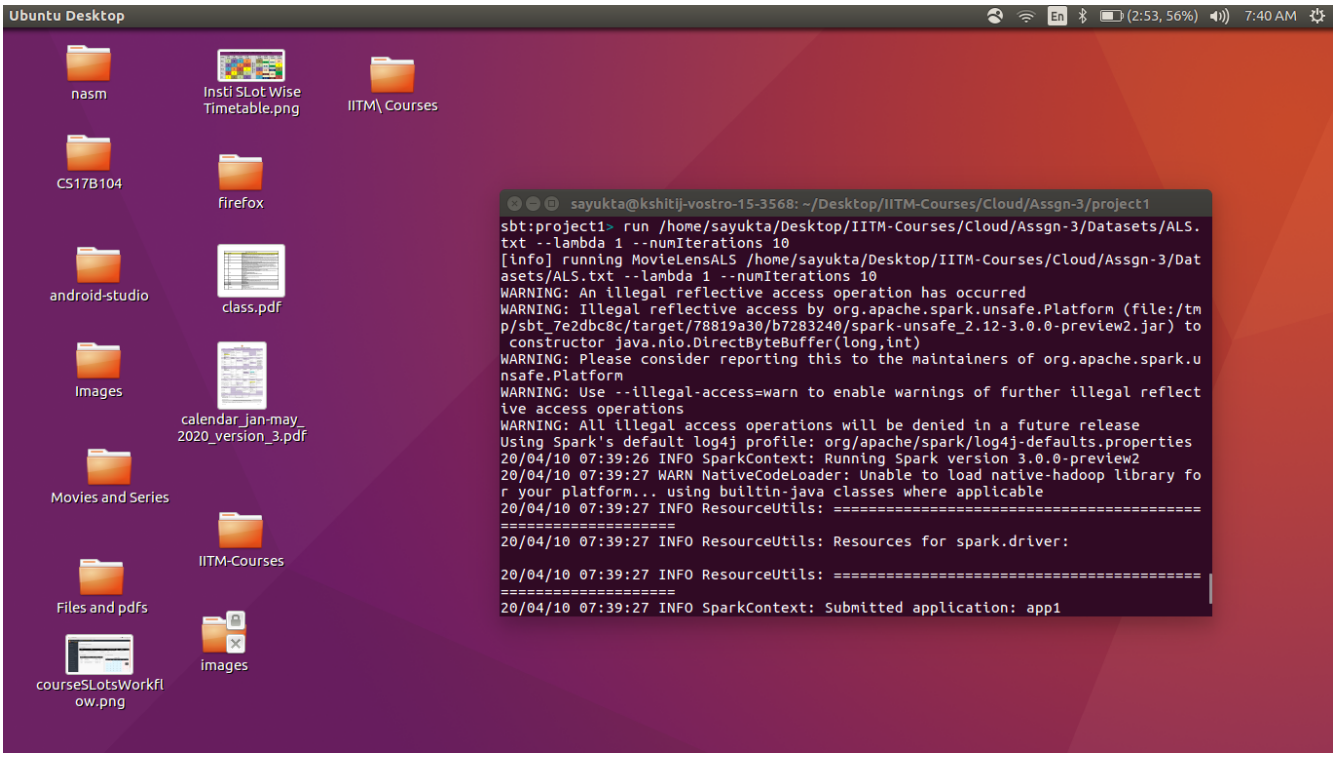


I changed this to make training size:test size, as 90:10, then 80:20, then 70:30, 60:40, 50:50,...,10:90.

For all these, I ran ALS for all combinations of lambda={0.2,1,5} and number of iterations={10,20,40}

The command to run this is:

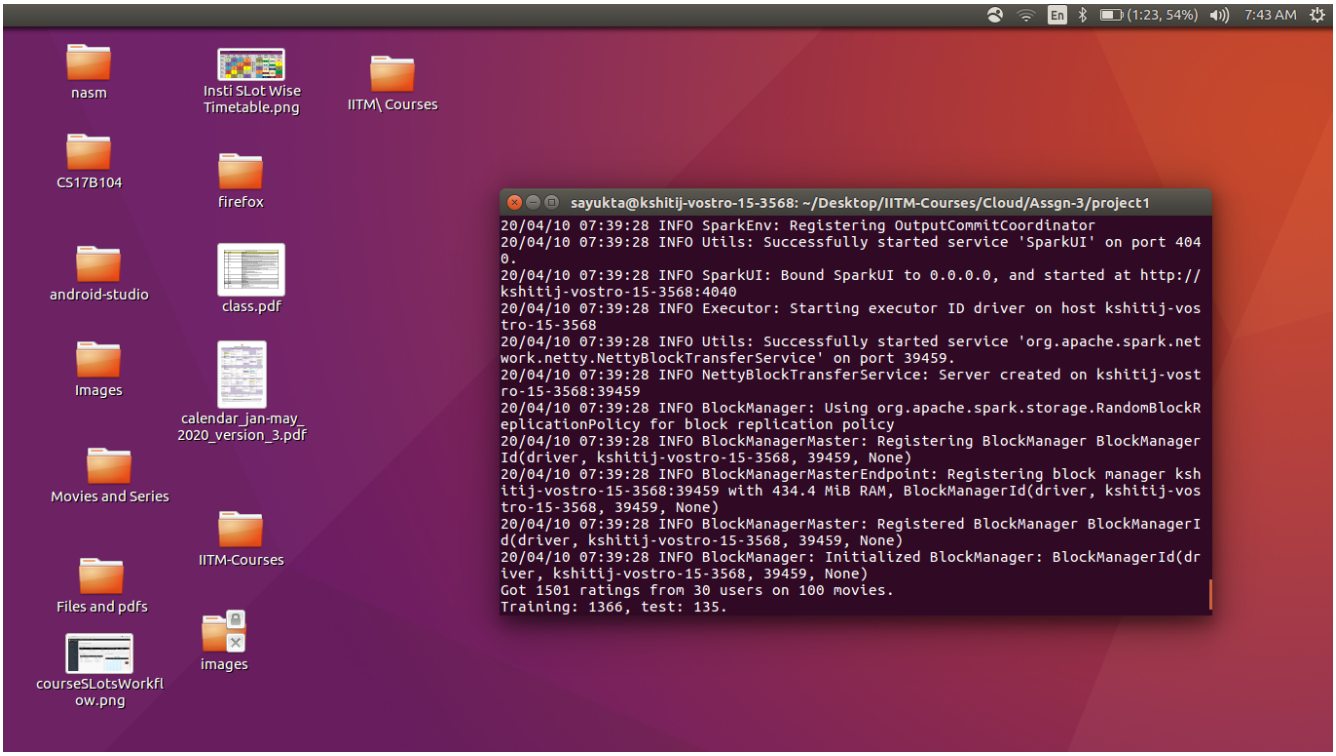
run /home/sayukta/Desktop/IITM-Courses/Cloud/Assgn-3/Datasets/ALS.txt –numIterations 20 –-rank 5 –-lambda 0.1



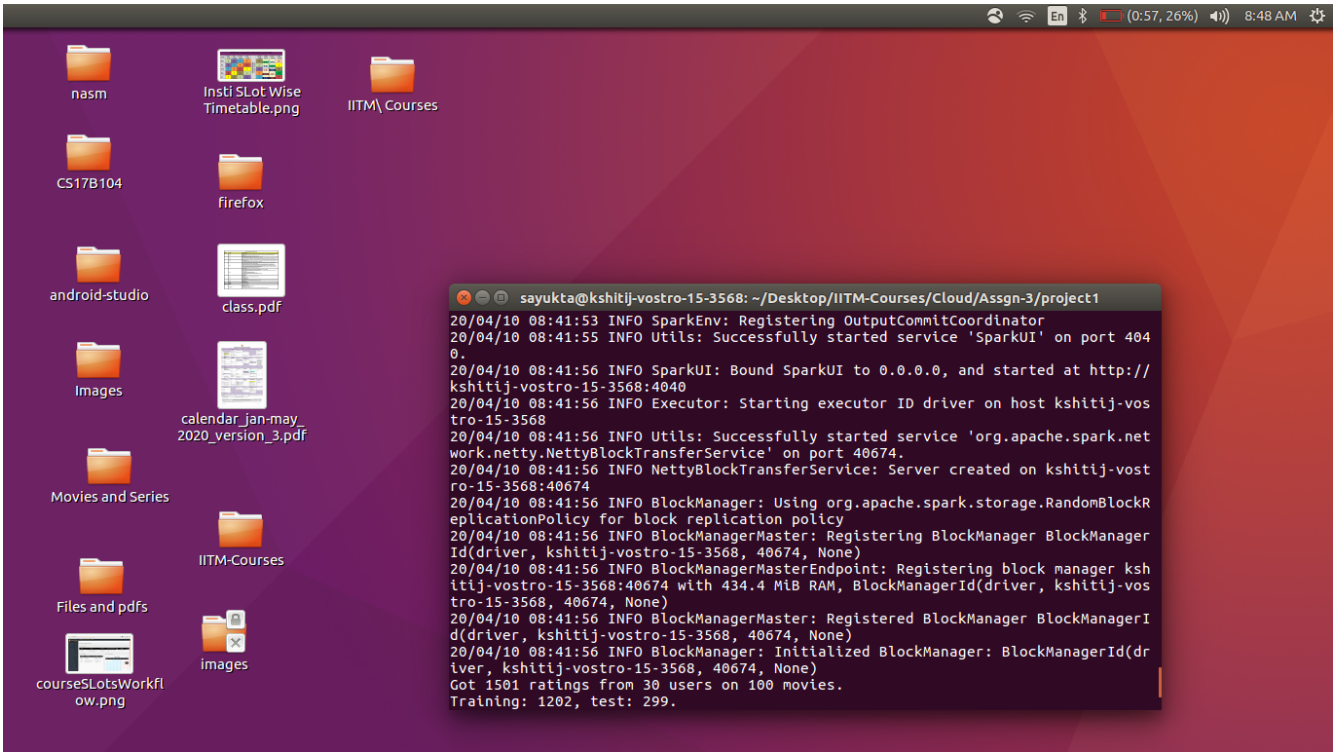
Don’t mention the parameters if you want to run the default values.

Here are the screenshots of almost all of the 1st times I ran this program for all ratios. They show what sizes the train and test data have been split into.

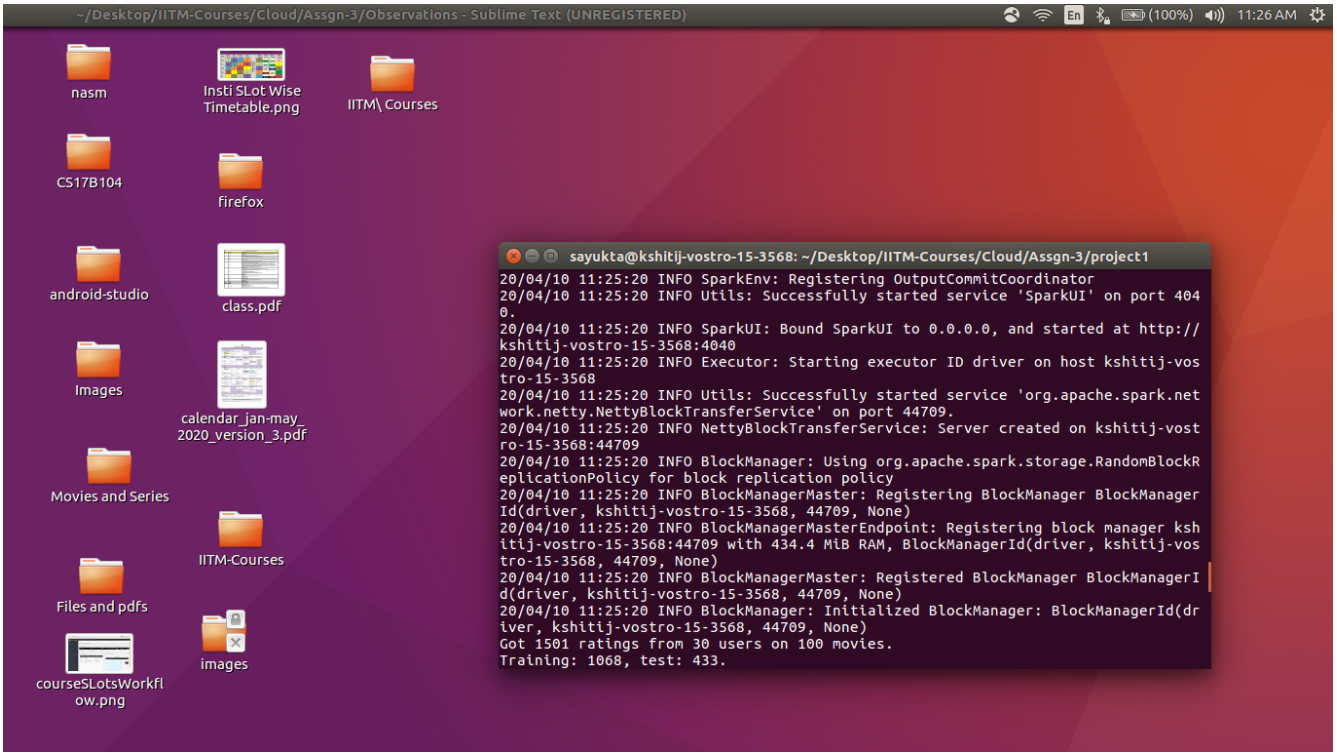
90:10-


(1:23, 54%)
7:43 AM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 07:39:28 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 07:39:28 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 07:39:28 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 07:39:28 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 07:39:28 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 39459.
20/04/10 07:39:28 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:39459
20/04/10 07:39:28 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 07:39:28 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 39459, None)
20/04/10 07:39:28 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:39459 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 39459, None)
20/04/10 07:39:28 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 39459, None)
20/04/10 07:39:28 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 39459, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 1366, test: 135.

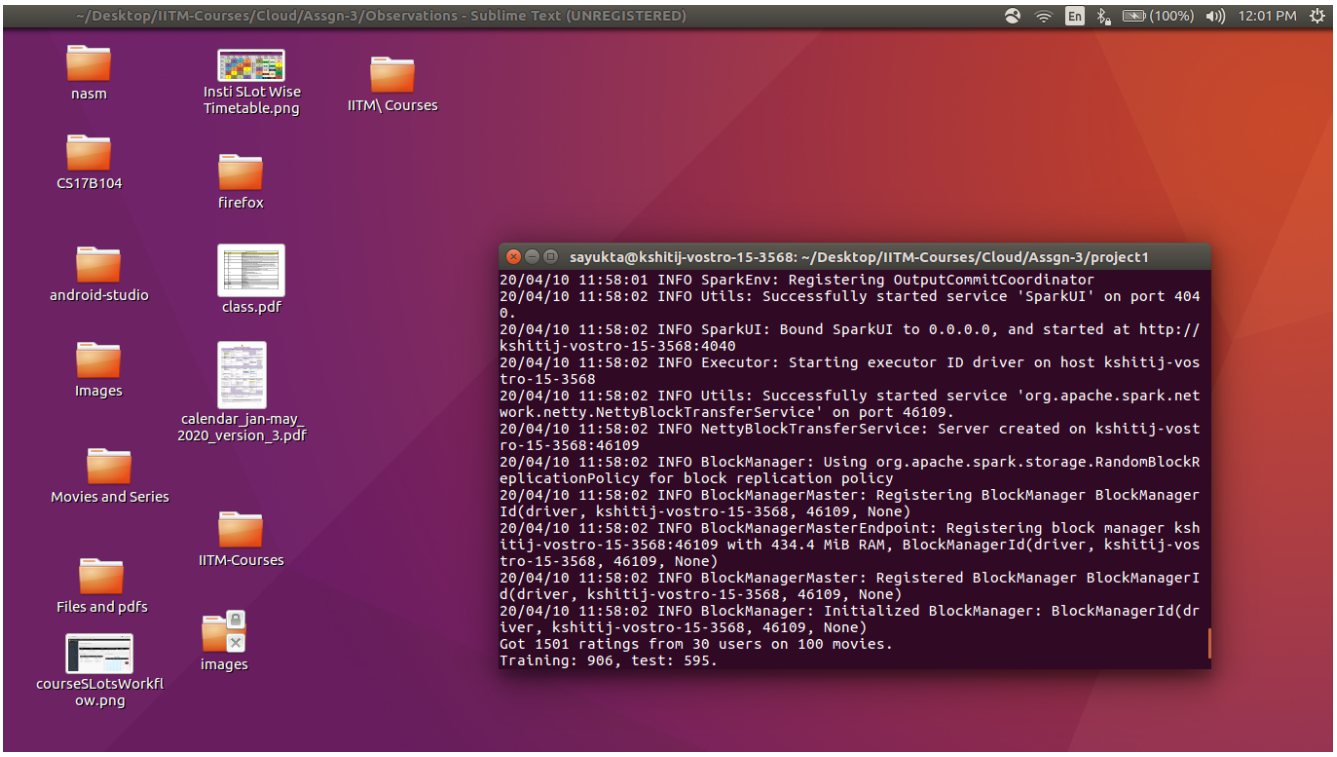
80:20-


(0:57, 26%)
8:48 AM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 08:41:53 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 08:41:55 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 08:41:56 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 08:41:56 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 08:41:56 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 40674.
20/04/10 08:41:56 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:40674
20/04/10 08:41:56 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 08:41:56 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 40674, None)
20/04/10 08:41:56 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:40674 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 40674, None)
20/04/10 08:41:56 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 40674, None)
20/04/10 08:41:56 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 40674, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 1202, test: 299.

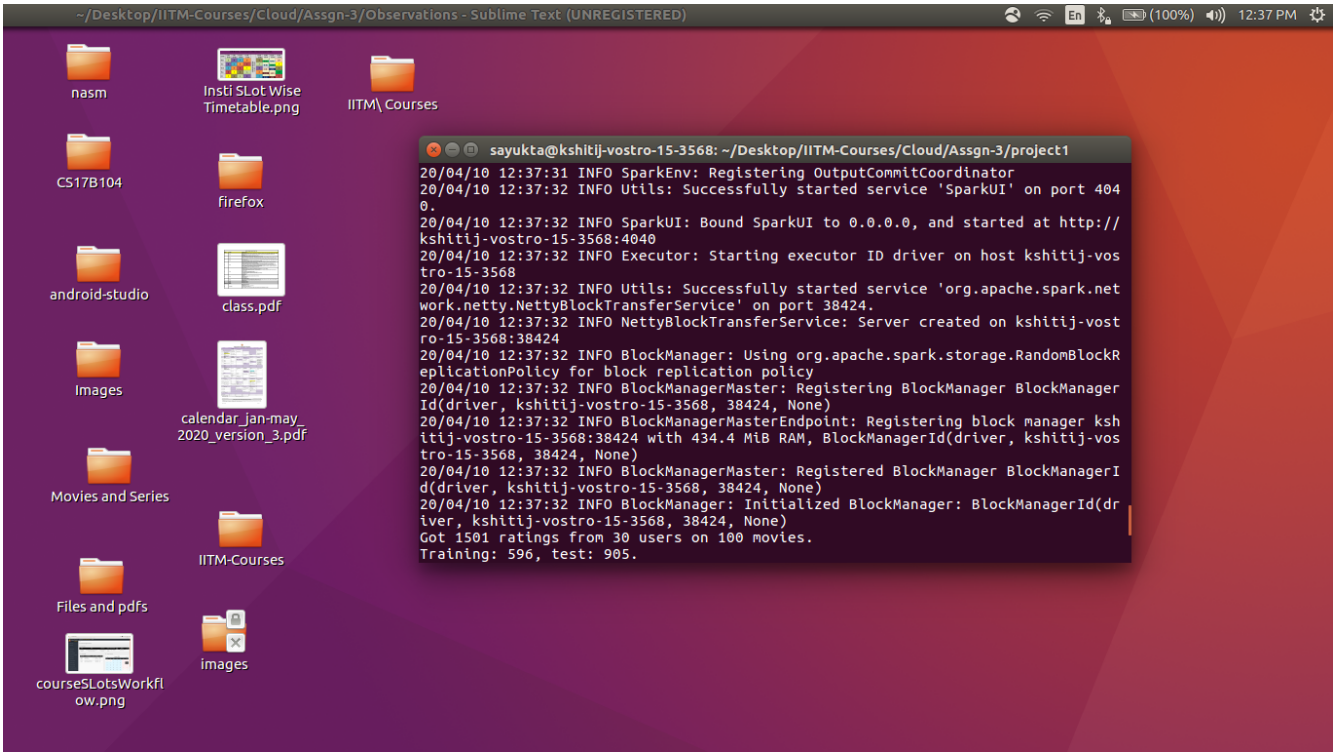
70:30-


~/Desktop/IITM-Courses/Cloud/Assgn-3/Observations - Sublime Text (UNREGISTERED)
11:26 AM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 11:25:20 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 11:25:20 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 11:25:20 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 11:25:20 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 11:25:20 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 44709.
20/04/10 11:25:20 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:44709
20/04/10 11:25:20 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 11:25:20 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 44709, None)
20/04/10 11:25:20 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:44709 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 44709, None)
20/04/10 11:25:20 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 44709, None)
20/04/10 11:25:20 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 44709, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 1068, test: 433.

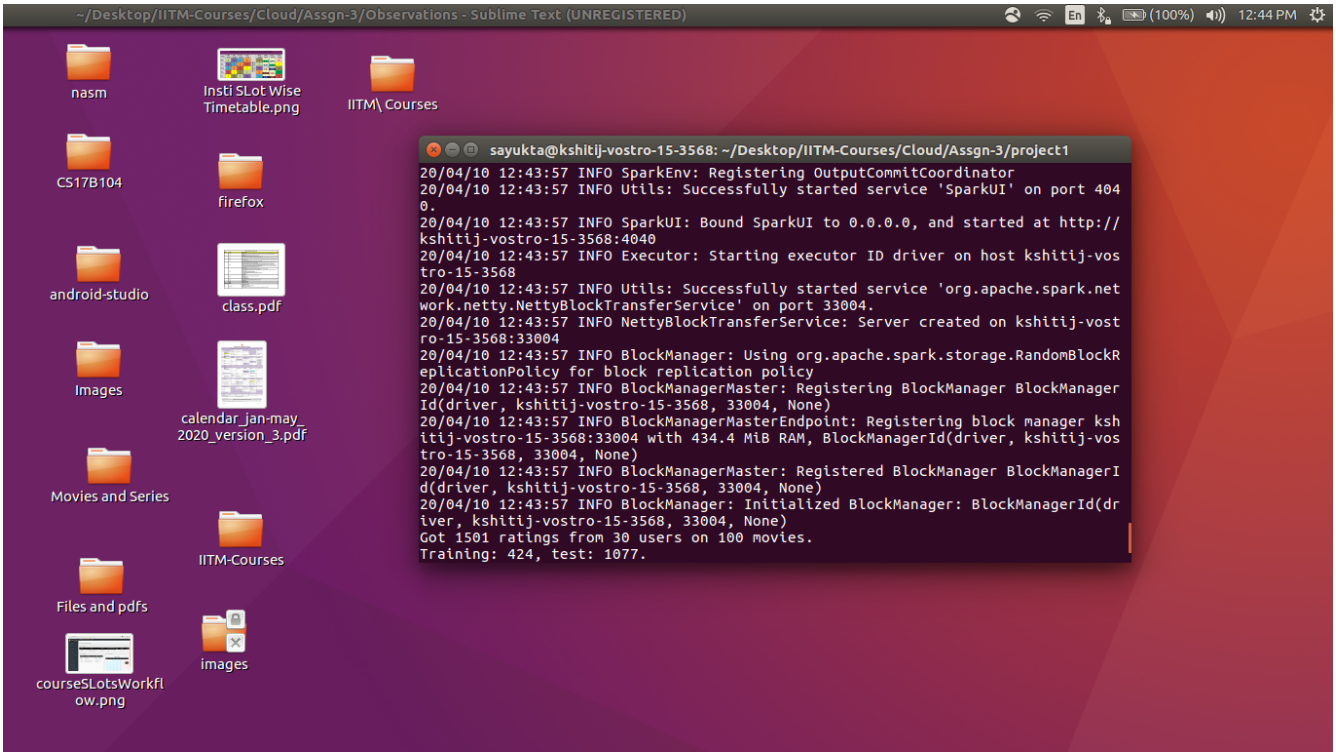
60:40-


~/Desktop/IITM-Courses/Cloud/Assgn-3/Observations - Sublime Text (UNREGISTERED)
12:01 PM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 11:58:01 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 11:58:02 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 11:58:02 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 11:58:02 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 11:58:02 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 46109.
20/04/10 11:58:02 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:46109
20/04/10 11:58:02 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 11:58:02 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 46109, None)
20/04/10 11:58:02 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:46109 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 46109, None)
20/04/10 11:58:02 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 46109, None)
20/04/10 11:58:02 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 46109, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 906, test: 595.

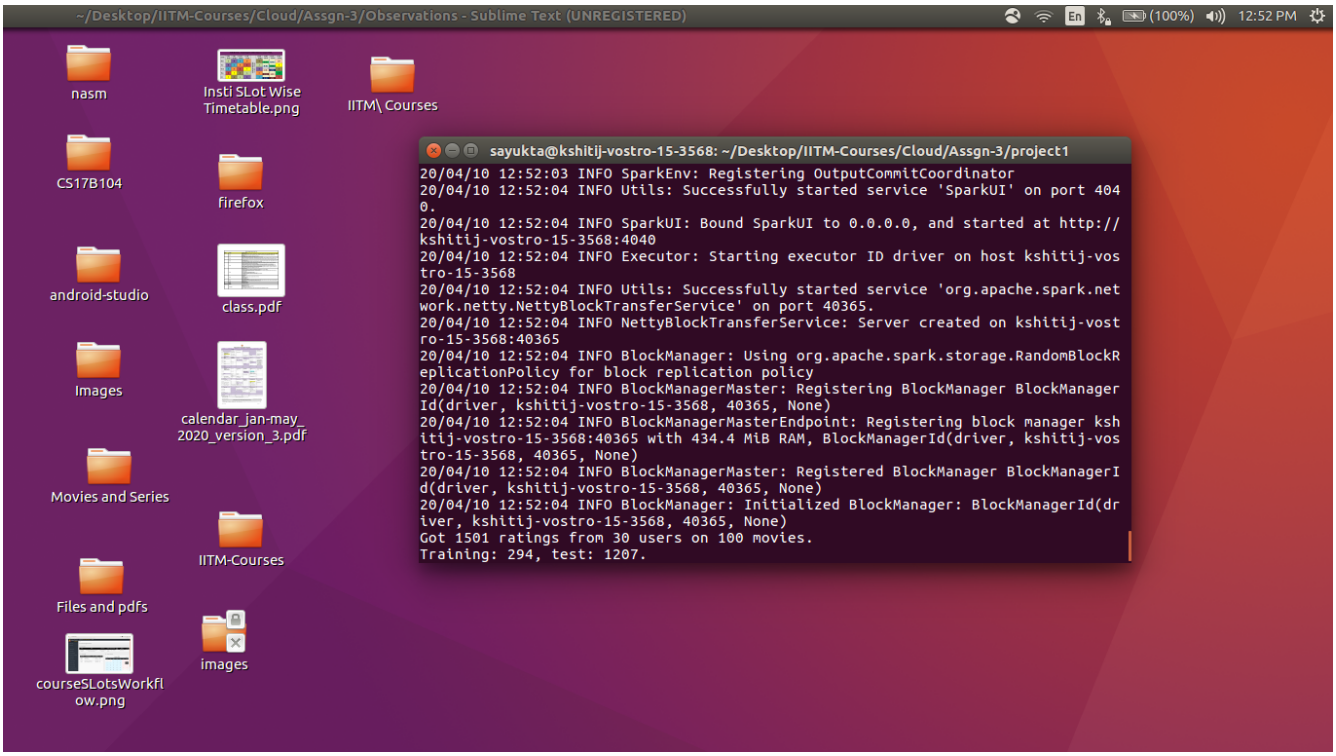
40:60-


~/Desktop/IITM-Courses/Cloud/Assgn-3/Observations - Sublime Text (UNREGISTERED)
(100%)
12:37 PM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 12:37:31 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 12:37:32 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 12:37:32 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 12:37:32 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 12:37:32 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 38424.
20/04/10 12:37:32 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:38424
20/04/10 12:37:32 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 12:37:32 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 38424, None)
20/04/10 12:37:32 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:38424 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 38424, None)
20/04/10 12:37:32 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 38424, None)
20/04/10 12:37:32 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 38424, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 596, test: 905.

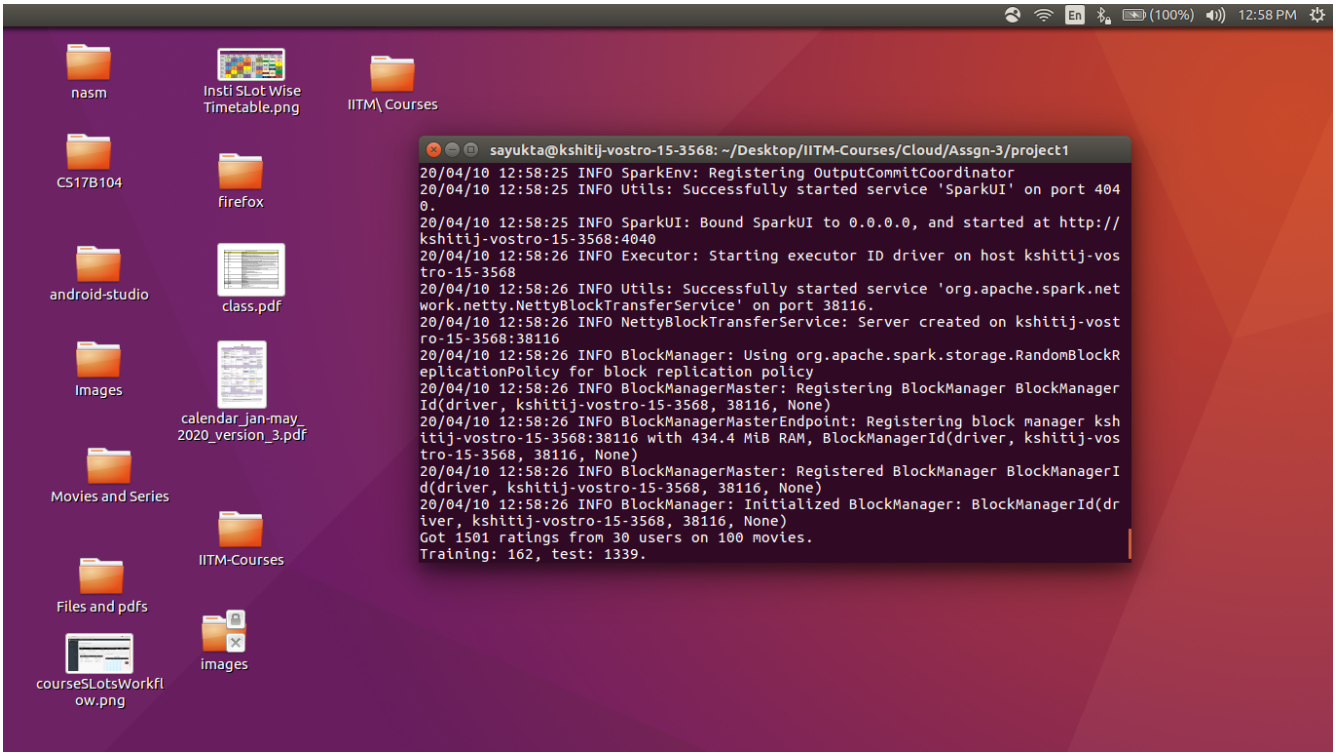
30:70-


~/Desktop/IITM-Courses/Cloud/Assgn-3/Observations - Sublime Text (UNREGISTERED)
12:44 PM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 12:43:57 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 12:43:57 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 12:43:57 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 12:43:57 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 12:43:57 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 33004.
20/04/10 12:43:57 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:33004
20/04/10 12:43:57 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 12:43:57 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 33004, None)
20/04/10 12:43:57 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:33004 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 33004, None)
20/04/10 12:43:57 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 33004, None)
20/04/10 12:43:57 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 33004, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 424, test: 1077.

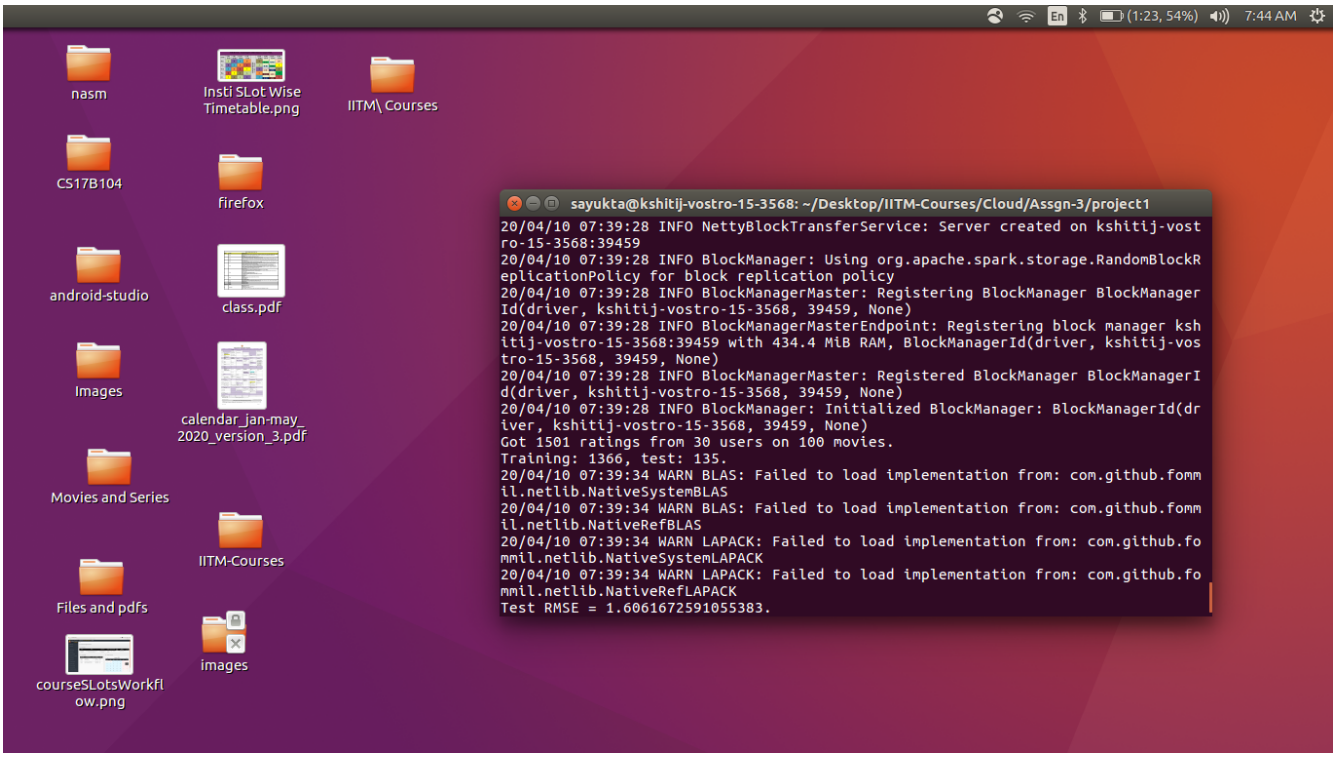
20:80-


~/Desktop/IITM-Courses/Cloud/Assgn-3/Observations - Sublime Text (UNREGISTERED)
12:52 PM
nasm
Insti SLot Wise Timetable.png
IITM\ Courses
CS17B104
firefox
android-studio
class.pdf
Images
calendar_jan-may_2020_version_3.pdf
Movies and Series
IITM-Courses
Files and pdfs
images
courseSLotsWorkflow.png
sayukta@kshitij-vostro-15-3568: ~/Desktop/IITM-Courses/Cloud/Assgn-3/project1
20/04/10 12:52:03 INFO SparkEnv: Registering OutputCommitCoordinator
20/04/10 12:52:04 INFO Utils: Successfully started service 'SparkUI' on port 4040.
20/04/10 12:52:04 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://kshitij-vostro-15-3568:4040
20/04/10 12:52:04 INFO Executor: Starting executor ID driver on host kshitij-vostro-15-3568
20/04/10 12:52:04 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 40365.
20/04/10 12:52:04 INFO NettyBlockTransferService: Server created on kshitij-vostro-15-3568:40365
20/04/10 12:52:04 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
20/04/10 12:52:04 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 40365, None)
20/04/10 12:52:04 INFO BlockManagerMasterEndpoint: Registering block manager kshitij-vostro-15-3568:40365 with 434.4 MiB RAM, BlockManagerId(driver, kshitij-vostro-15-3568, 40365, None)
20/04/10 12:52:04 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, kshitij-vostro-15-3568, 40365, None)
20/04/10 12:52:04 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, kshitij-vostro-15-3568, 40365, None)
Got 1501 ratings from 30 users on 100 movies.
Training: 294, test: 1207.

10:90-



This is how the final result of 1 such execution looked:



Results:

1st column: Regularization parameter 2nd column: # of iterations 3rd column: Test RMSE error

```
Test RMS errors for train_data:test_data= 90 : 10
 [[ 0.2        10.          0.88353792]
 [ 1.         10.          0.90417007]
 [ 5.         10.          0.9613902 ]
 [ 0.2        20.          1.60616726]
 [ 1.         20.          1.61504794]
 [ 5.         20.          1.49140474]
 [ 0.2        40.          2.09794198]
 [ 1.         40.          1.98284569]
 [ 5.         40.          2.23325354]]
Test RMS errors for train_data:test_data= 80 : 20
 [[ 0.2        10.          1.02957151]
 [ 1.         10.          1.03096771]
 [ 5.         10.          1.10813583]
 [ 0.2        20.          1.43403387]
 [ 1.         20.          1.60788534]
 [ 5.         20.          1.54648357]
 [ 0.2        40.          2.20279543]
 [ 1.         40.          2.14128615]
 [ 5.         40.          2.17762473]]
Test RMS errors for train_data:test_data= 70 : 30
 [[ 0.2        10.          1.06373663]
 [ 1.         10.          1.10748153]
 [ 5.         10.          1.06862574]
 [ 0.2        20.          1.51199456]
 [ 1.         20.          1.43685131]
 [ 5.         20.          1.5464546 ]
 [ 0.2        40.          2.10224201]
 [ 1.         40.          2.17892323]
 [ 5.         40.          2.05740273]]
Test RMS errors for train_data:test_data= 60 : 40
 [[ 0.2        10.          1.02508173]
 [ 1.         10.          1.08840263]
 [ 5.         10.          1.13868564]
 [ 0.2        20.          1.51123961]
 [ 1.         20.          1.60677845]
 [ 5.         20.          1.49133515]
 [ 0.2        40.          2.1563481 ]
 [ 1.         40.          2.17964257]
 [ 5.         40.          2.16623221]]
Test RMS errors for train_data:test_data= 50 : 50
 [[ 0.2        10.          1.11721683]
 [ 1.         10.          1.21799336]
 [ 5.         10.          1.1315039 ]
 [ 0.2        20.          1.45851439]
 [ 1.         20.          1.50572964]
 [ 5.         20.          1.58248119]
 [ 0.2        40.          2.09252996]
 [ 1.         40.          2.10575652]
 [ 5.         40.          2.1202274 ]]
Test RMS errors for train_data:test_data= 40 : 60
 [[ 0.2        10.          1.18730461]
 [ 1.         10.          1.24012897]
 [ 5.         10.          1.20491116]
```

```
 [ 0.2        20.          1.56539213]
 [ 1.         20.          1.52754403]
 [ 5.         20.          1.52009261]
 [ 0.2        40.          2.14521368]
 [ 1.         40.          2.16917237]
 [ 5.         40.          2.12972165]]
Test RMS errors for train_data:test_data= 30 : 70
 [[ 0.2        10.          1.4067198 ]
 [ 1.         10.          1.33970171]
 [ 5.         10.          1.31692408]
 [ 0.2        20.          1.55605547]
 [ 1.         20.          1.55033343]
 [ 5.         20.          1.53773046]
 [ 0.2        40.          2.16210787]
 [ 1.         40.          2.08228596]
 [ 5.         40.          2.12841339]]
Test RMS errors for train_data:test_data= 20 : 80
 [[ 0.2        10.          1.53861237]
 [ 1.         10.          1.48407088]
 [ 5.         10.          1.4531197 ]
 [ 0.2        20.          1.5816078 ]
 [ 1.         20.          1.57483072]
 [ 5.         20.          1.51522952]
 [ 0.2        40.          2.17705008]
 [ 1.         40.          2.1595567 ]
 [ 5.         40.          2.09935275]]
Test RMS errors for train_data:test_data= 10 : 90
 [[ 0.2        10.          1.88407489]
 [ 1.         10.          1.67212956]
 [ 5.         10.          1.69563679]
 [ 0.2        20.          1.8483204 ]
 [ 1.         20.          1.56104108]
 [ 5.         20.          1.63303345]
 [ 0.2        40.          2.17321109]
 [ 1.         40.          2.11833467]
 [ 5.         40.          2.19292165]]
```

Inference:

I have kept rank=5 fixed.

As the ratio of train_data size: test_data size decreases, you can see a visible difference in test RMSE values for observation with lower regularization parameter; for Train:Test=90:10, the max_error for lambda=0.2 is 0.96, whereas Train:Test=10:90, the min_error is 1.67.
Thus clearly, the less size of train data factor in the 10:90 case outweighs the presence of a low lambda in the overfitted case of 90:10 in its contribution to the test RMSE.

On the other hand, for lambda=5, the test RMSE is a constant high of ~2.1 across all data size ratios. This may be because even in the overfitted case of 90:10 (train data:test data) case, lambda=5 is actually too large a value to prevent overfitting; rather it is showing its presence in the form of a high error.

In fact, I think that even lambda=0.2 is too high a lambda for stopping overfitting, as if this were a low value for lambda, we would have gotten high errors for the overfitted case of 90:10 (train data:test data) with lambda=0.2;

instead we are getting the least errors for this across all the observations. This shows that lambda=0.2 is in fact better at stopping overfitting than lambda=5, which is just too large a lambda.

In general, the average error is increasing as we are decreasing the training data size, across all the lambdas.

After 50:50, when the test_data size surpasses train_data size, the increase in number of iterations is in general, decreasing test RMSE, which is expected. However, above 50:50, there is no clear boundary for no. of iterations as such.

This can be partially due to very low erros for large training data size and low lambda, such that no. of iterations are not causing much effect.

2) FPGrowth:

FP_Part1:

Instead of MovieLensALS.scala, put FPGrowthExample.scala.
Note that both in MovieLensALS.scala and FPGrowthExample.scala, I put the source code of class AbstractParams, which I got from github code of apache spark.

For FP_Part1.csv, I found that in some lines, some items were been repeated. FPGrowth requires unique items to be in 1 line, so data had to be cleaned.

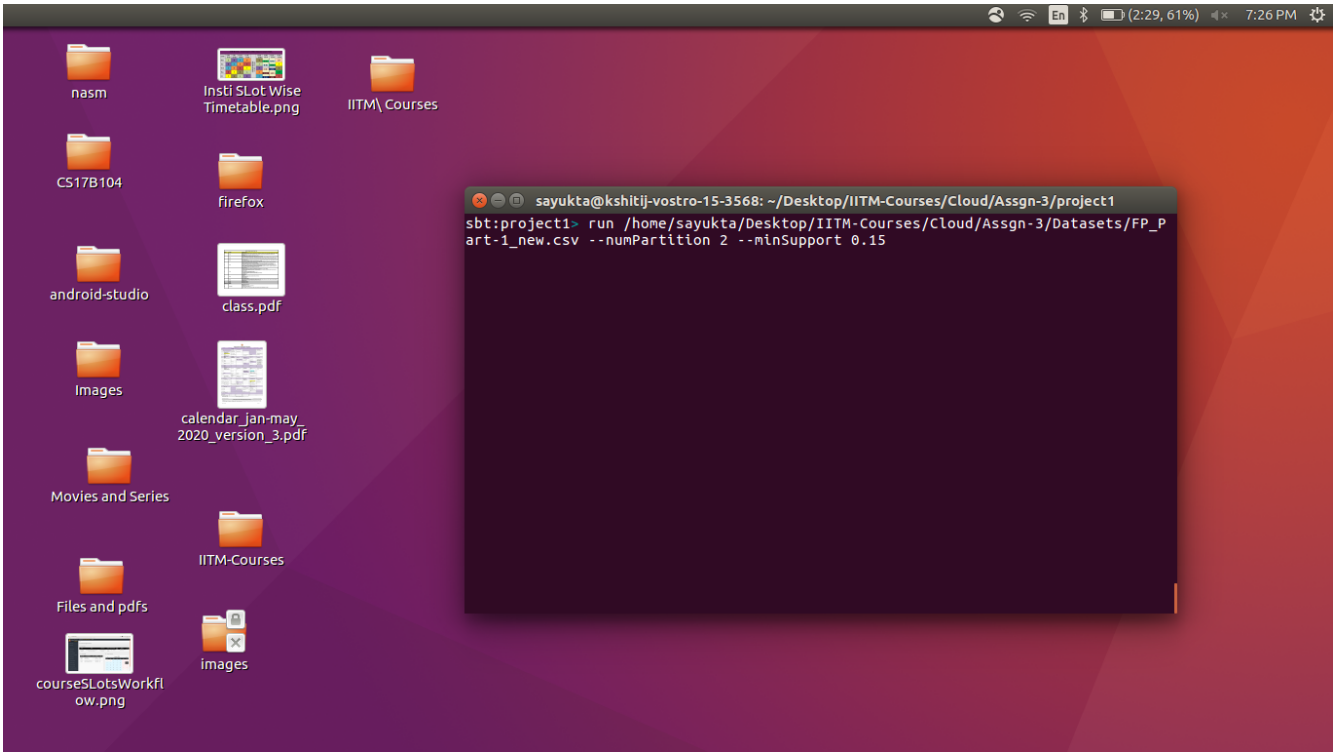
For this, I used, ‘format1.py’. This is my data cleaning program.
This opens the file ‘ FP_Part1.csv’, goes through each line and sorts the items alphabetically. Then, we just have to go through each of the sorted lines, and take unique items; this can be done in O(n) time.
Sorting is the bottleneck.

Since efficiency of data cleaning program was asked, I used Quick Sort.

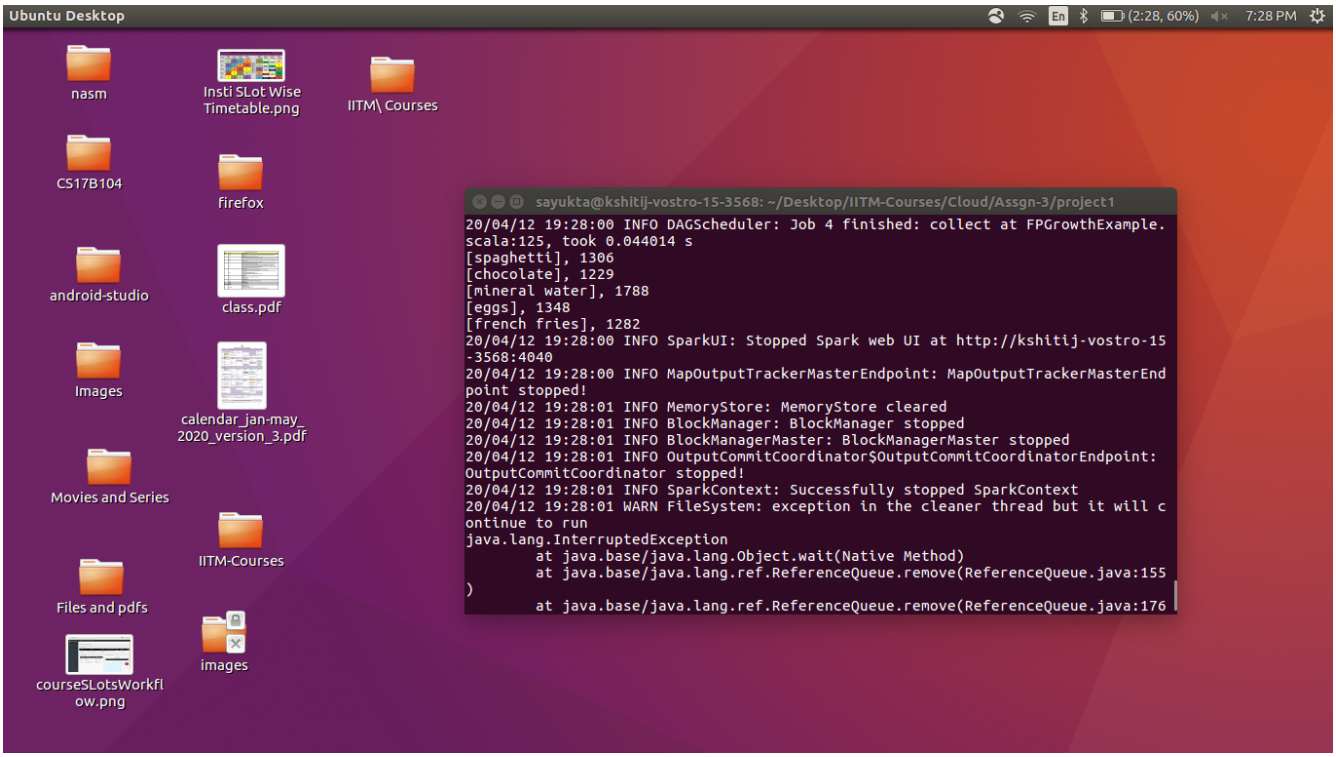
So, my ‘format1.py’ runs in O(nlogn) time.

I wrote each line thus got, to, ‘FP_Part-1.csv’.

Next, I ran this file, from sbt inside ‘project1’, thus:



Output got was:



So, top 5 most frequent items are:
1) Mineral water- 1788 times
2) Eggs- 1348 times

3) Spaghetti- 1306 times
4) French fries- 1282 times
5) Chocolate- 1229 times

Note that I used minSupport as 0.15. Increasing this reduces the number of items displayed.

I made ‘format2.py’, which had the same code as ‘format1.py’ except that in the former

FP_Part2:

I made FP2_to_FP1.py, which prints out all the items with the same InVoice_Number on the same line. All items of a single order will have a single unique InVoice_Number.
I used:
python FP2_to_FP1 > FP_Part-2_new.csv

to redirect each new_line order in a file ‘FP_Part-2_new.csv’.

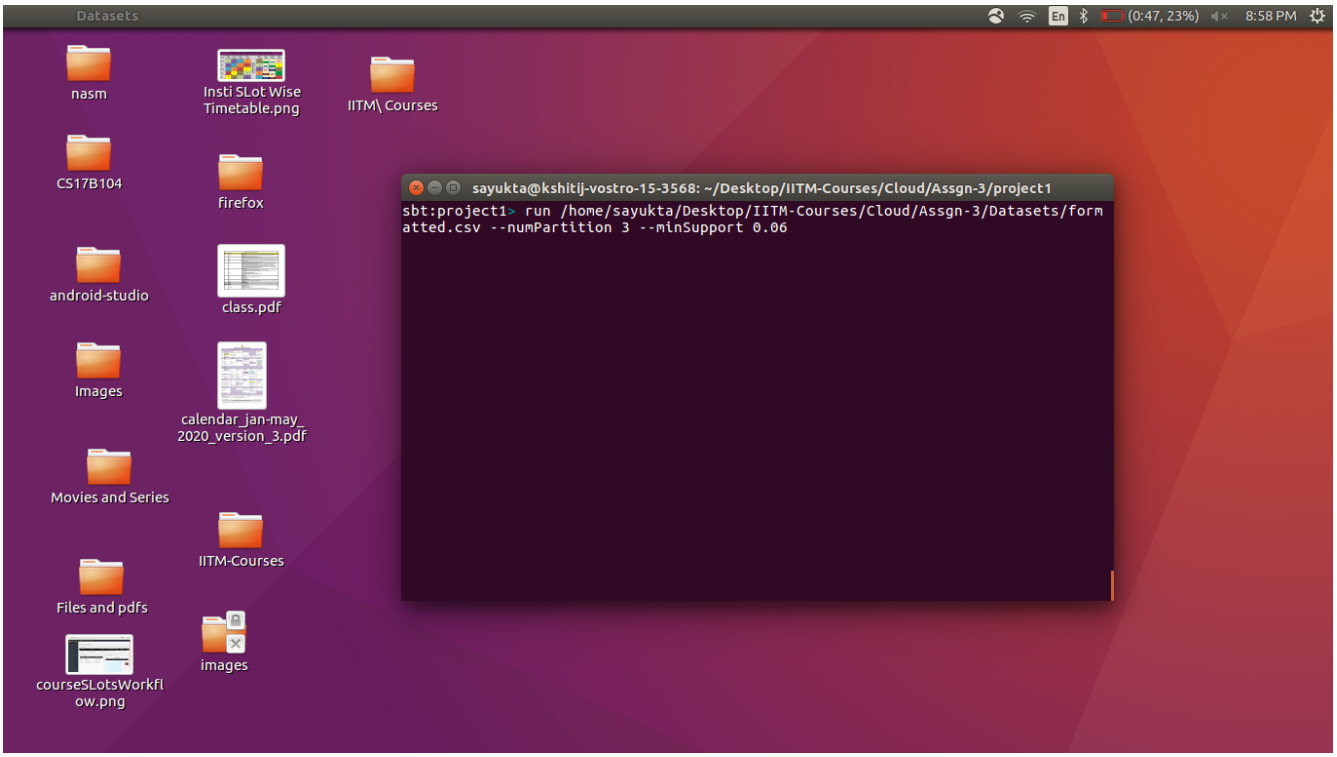
Then, again, we face same problem as FP_Part1, where multiple items may be on the same line.

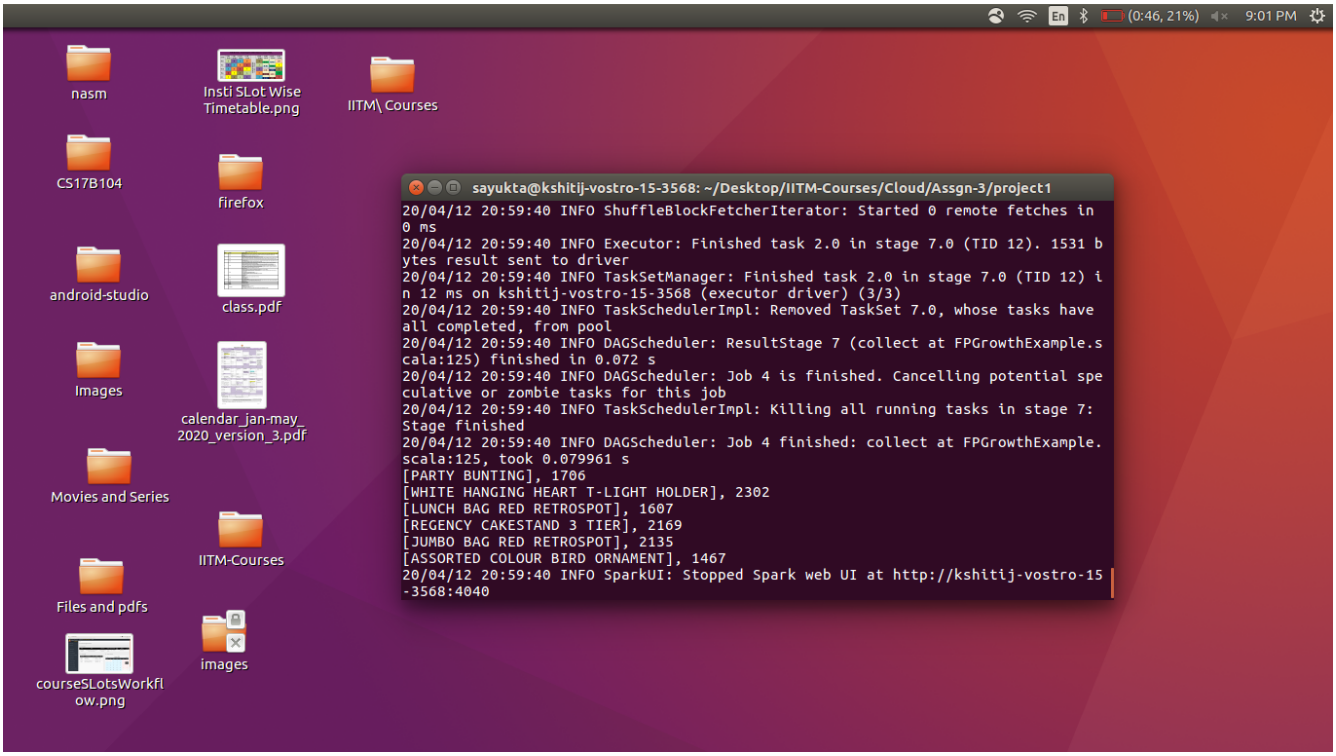
So, I made a file ‘format2.py’, which had the same code as ‘format1.py’, except that it read ‘FP_Part-2_new.csv’, and wrote to ‘FP_Part-2_real_new.csv'.

Now, ‘FP_Part-2_real_new.csv' had the clean data.

I copied this to ‘formatted.csv’, as required in the probelm.

I ran this as:





Thus, 5 most frequent items are:
1) White hanging heart T-light holder- 2302 times
2) Regency cakestand 3 tier- 2169 times
3) Jumbo Bag Red Retrospot- 2135 times
4) Party Bunting- 1706 times
5) Lunch bag red retrospect- 1607 times

NOTE:
1) I have not used HDFS as I had accidently deleted my VDI image of BOSS and I can't download 15 GB file at my home.

2) I have written Python Programs for data cleaning of FP_Part1 and FP_Part2.

3) I have used sbt to run programs instead of spark-submit.

4) For the 3 different programs for ALS, FPGrowth, Data Cleaning asked:

format1.py is the ALS code, FP2_to_FP1.py is the FPGrowth code, format2.py is the Data Cleaning code.