Machine Learning with Pyspark

Part 1: Create a databricks account.

Step 1: Open the link:

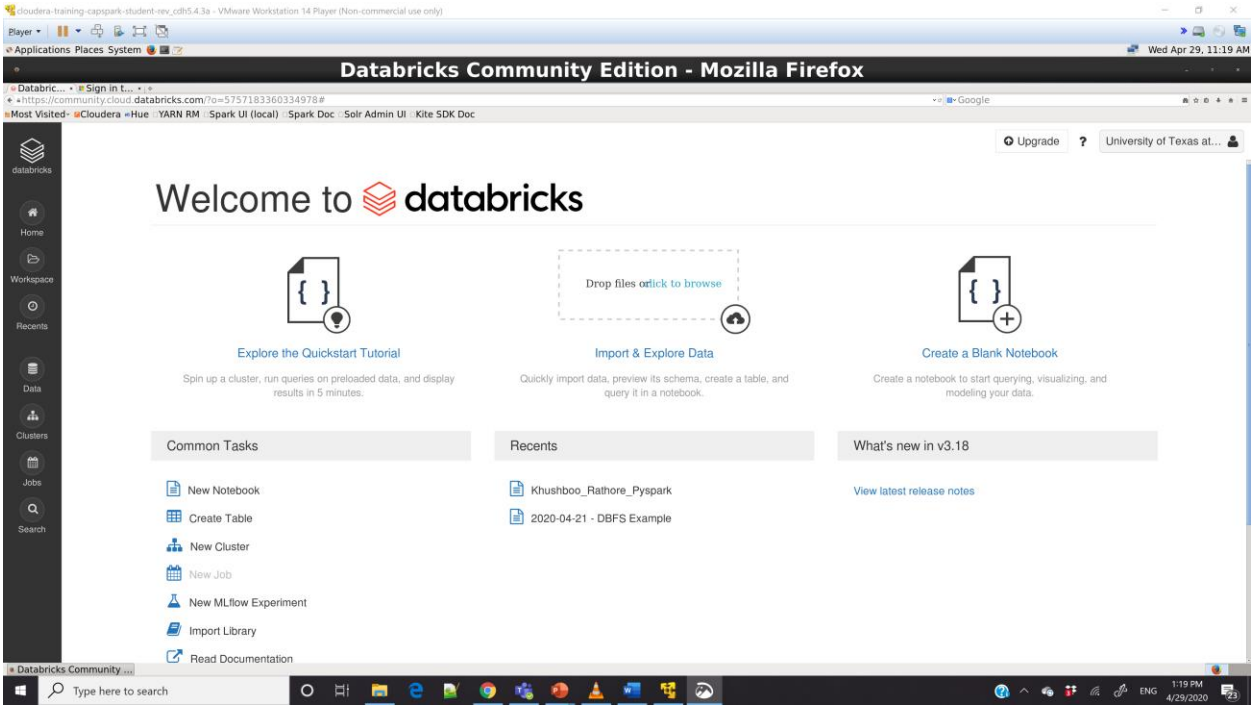
https://community.cloud.databricks.com/login.html

Click on sign up.

Step 2: Verify and confirm the databricks account from the email you have received from Databricks.

Go to the link again and click on “Sign In”.

Q. Login to your databricks account and take a screenshot of the Home page output and paste it below.

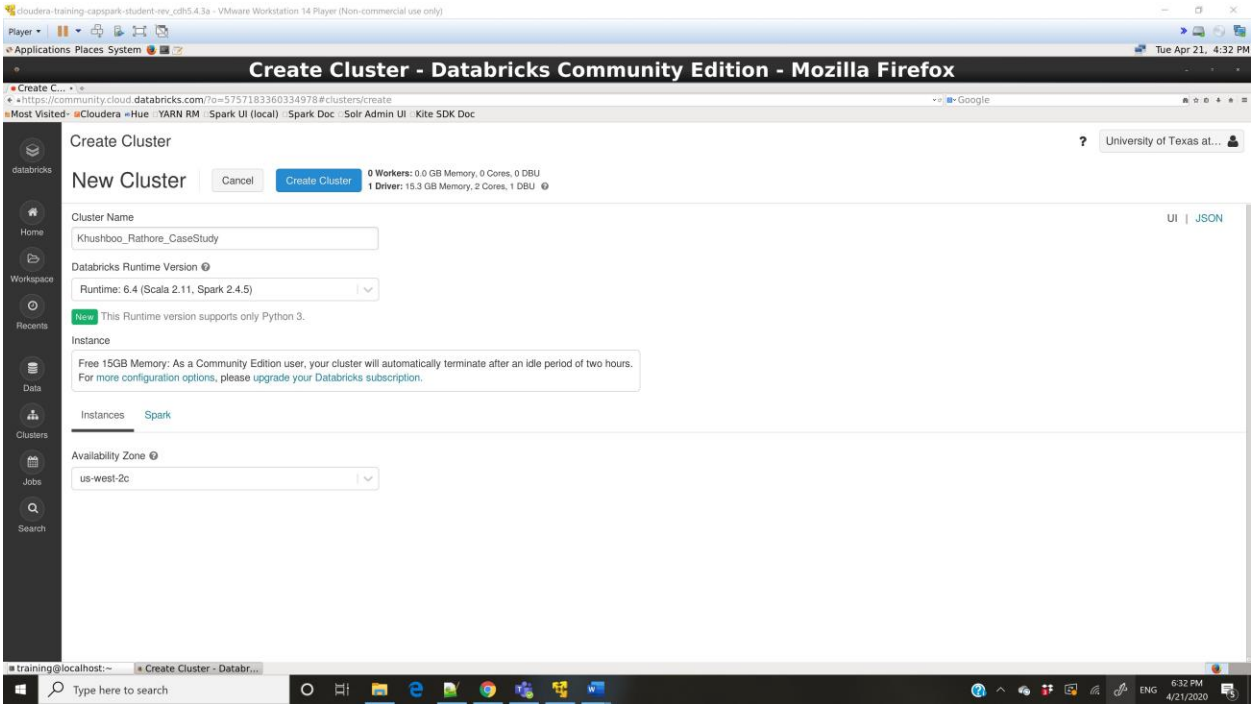


Part 2: Create a cluster node.

Step 3: Click on the “Clusters” icon as shown in the screenshot below:

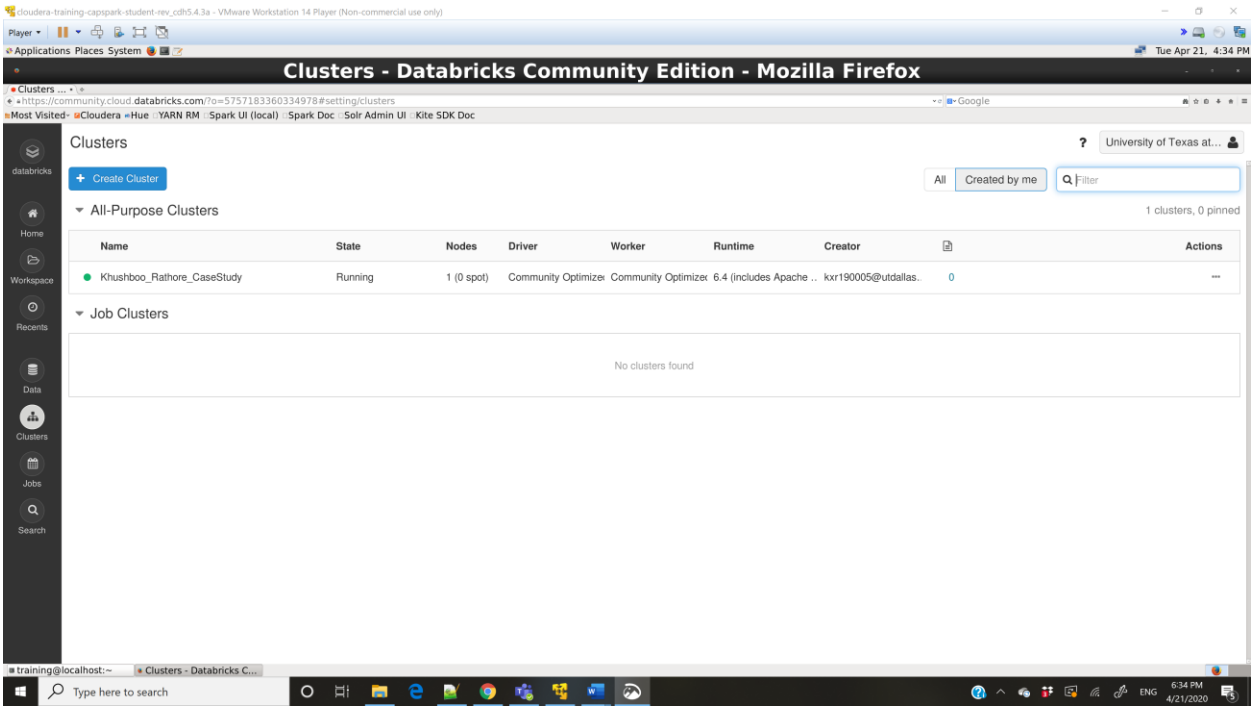
Step 4: Name the cluster as “FirstName_LastName_CaseStudy”.

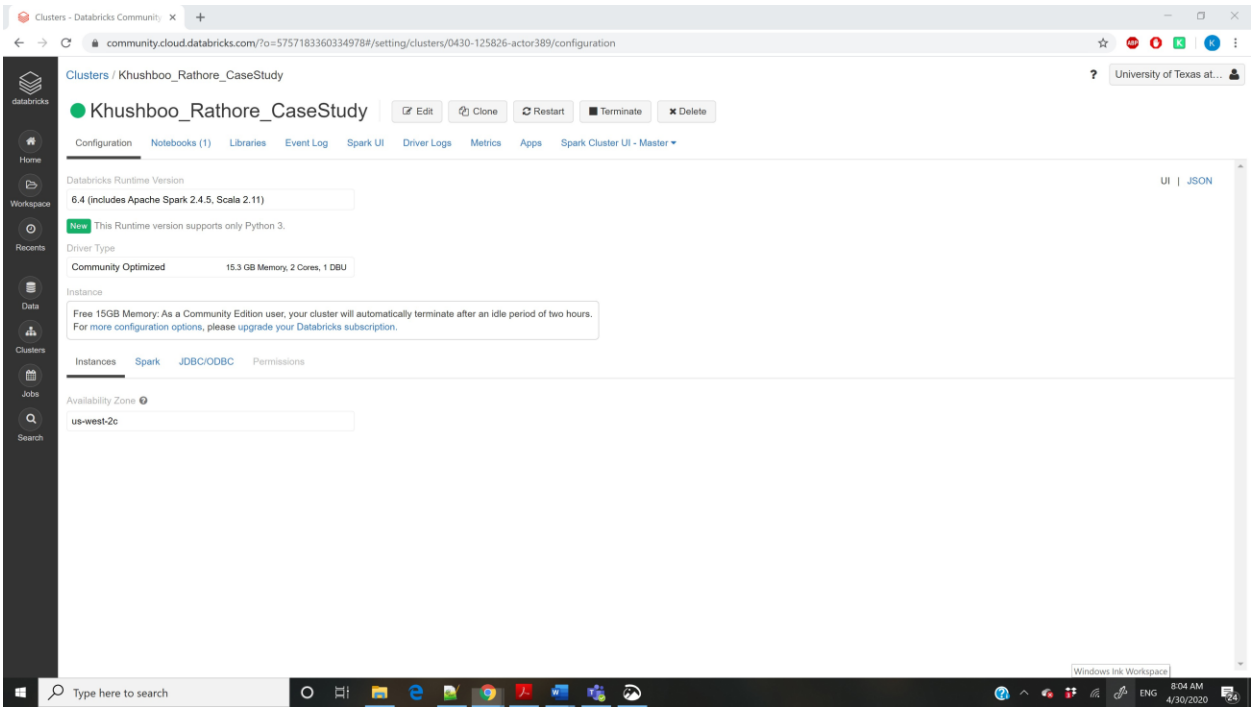
Q. Take a screenshot of the Create Cluster page and paste it below.



Step 5: Green sign shows that a cluster is created and running successfully.

Q. Take a screenshot of the running Cluster Page and paste it below.



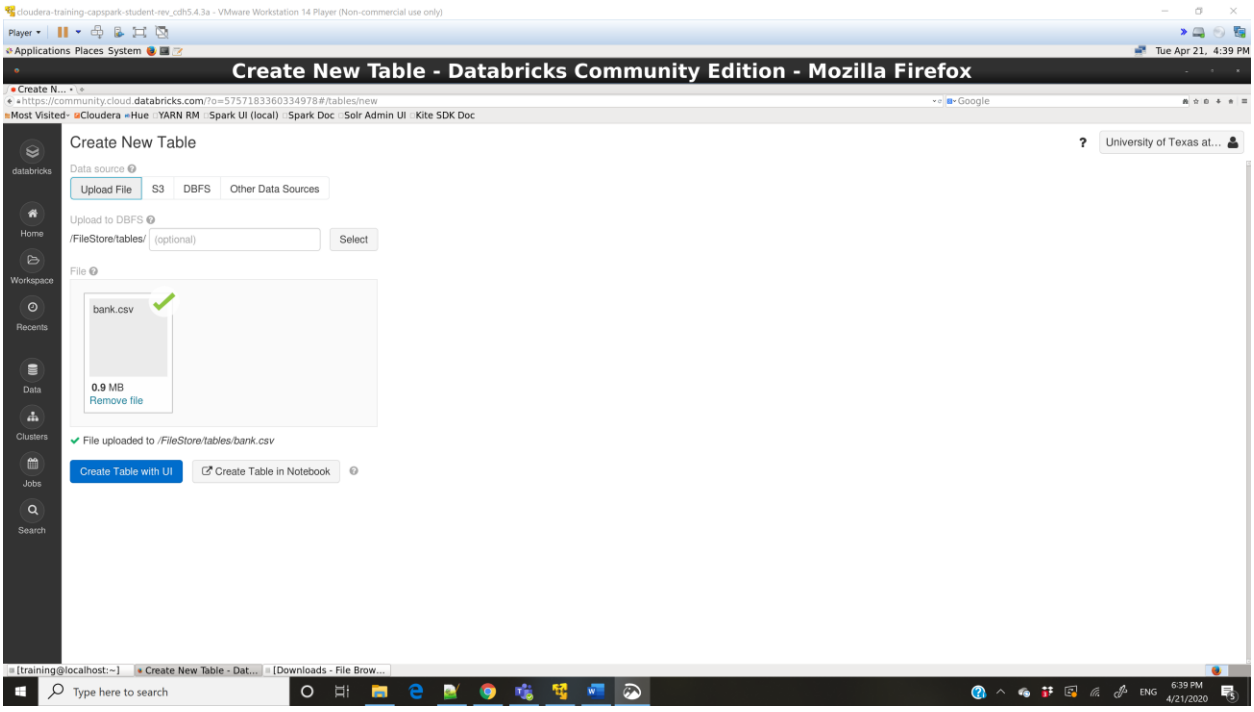


Part 3: Create Datatable.

Step 6: Download the “Bank.csv” file from eLearning.

Step 7: Click on browse and select the “bank.csv” file that you downloaded from eLearning.

Take a screenshot of the uploaded file and paste it below.



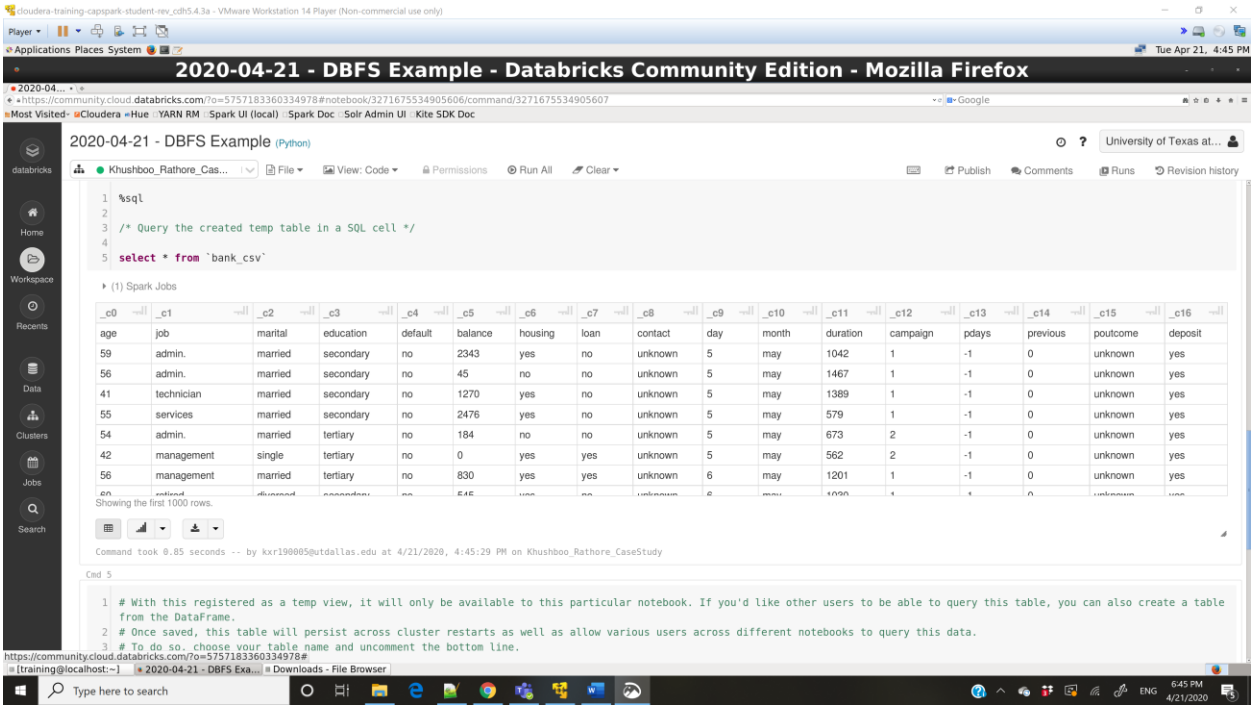
Step 8: Click on “Create Table in Notebook”.

Step 9: The file location can now be seen.

Q. Take a screenshot of the table created and paste it below.

Copy the file location and paste it here.

file_location = "/FileStore/tables/bank.csv"



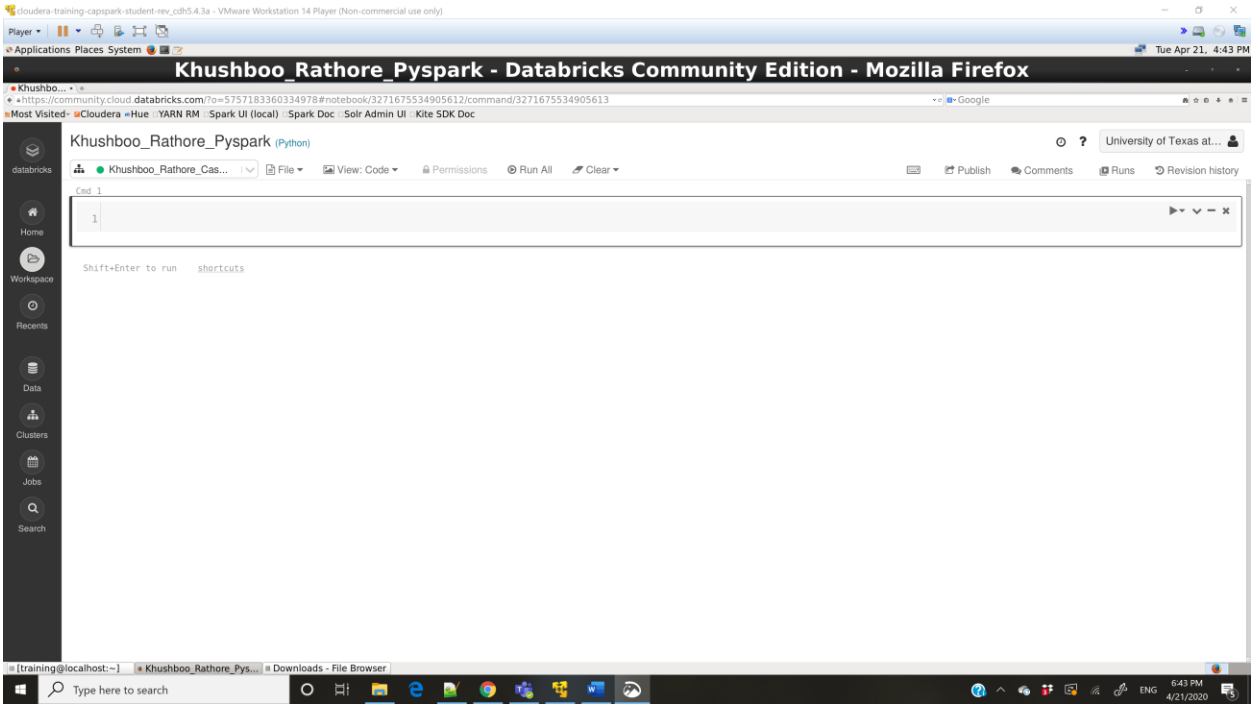
_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c8	_c9	_c10	_c11	_c12	_c13	_c14	_c15	_c16
age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
59	admin.	married	secondary	no	2343	yes	no	unknown	5	may	1042	1	-1	0	unknown	yes
56	admin.	married	secondary	no	45	no	no	unknown	5	may	1467	1	-1	0	unknown	yes
41	technician	married	secondary	no	1270	yes	no	unknown	5	may	1389	1	-1	0	unknown	yes
55	services	married	secondary	no	2476	yes	no	unknown	5	may	579	1	-1	0	unknown	yes
54	admin.	married	tertiary	no	184	no	no	unknown	5	may	673	2	-1	0	unknown	yes
42	management	single	tertiary	no	0	yes	yes	unknown	5	may	562	2	-1	0	unknown	yes
56	management	married	tertiary	no	830	yes	yes	unknown	6	may	1201	1	-1	0	unknown	yes

Part 4 : DataFrame Operations.

Step 10: Create new Notebook.

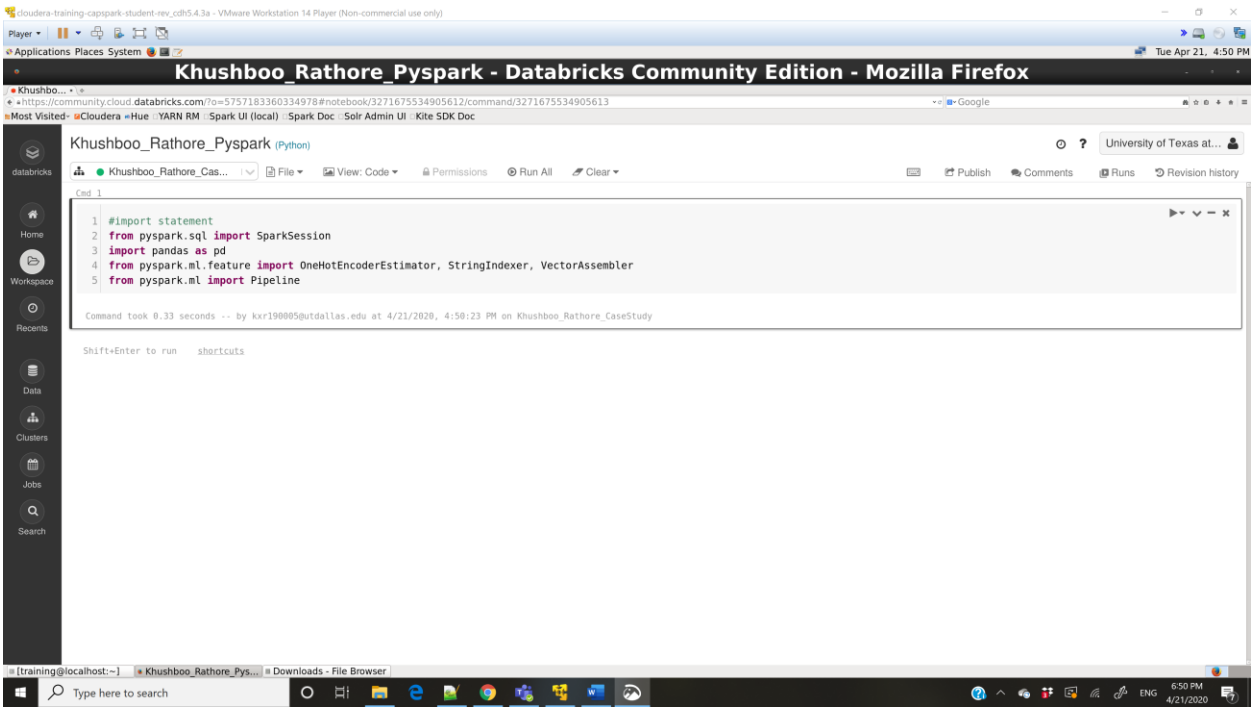
Step 11: Create the notebook with “FirstName_LastName_Pyspark”.

Q. Take a screenshot of the blank notebook and paste it below.



Step 11: Import the required libraries from Spark.

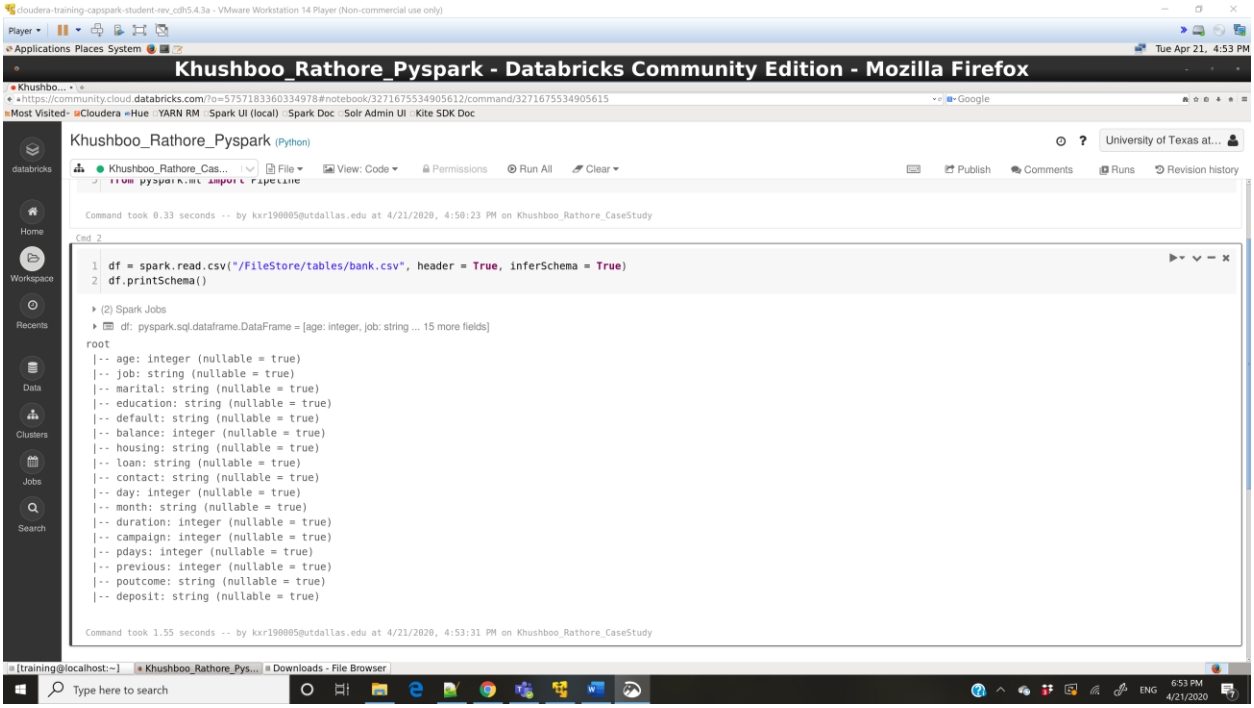
Q. Take a screenshot of the output of the code and paste it below.



Step 12:

Q. Take a screenshot of the output of the code and paste it below.

The input variables are age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome



Step 13: Type the code below in the next cell and Run the cell.

Q. Take a screenshot of the output of the code and paste it below.

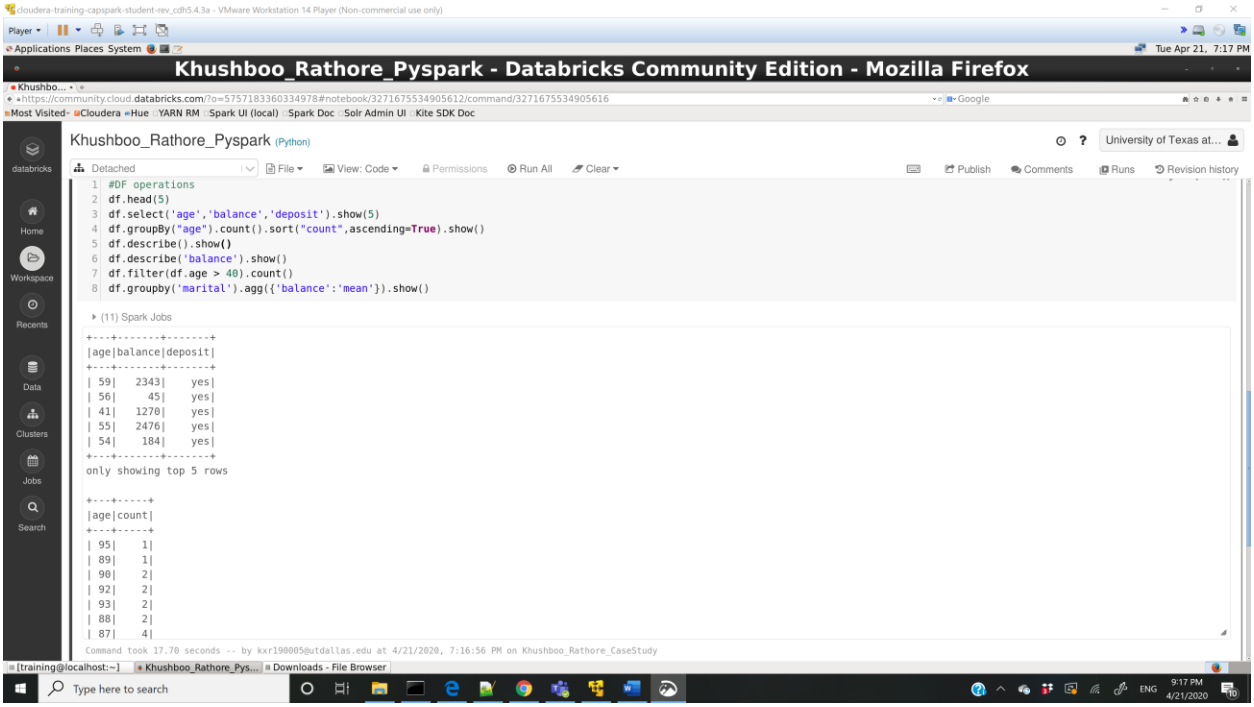
i. Describe “Select”, “Group by” operations in one line. –

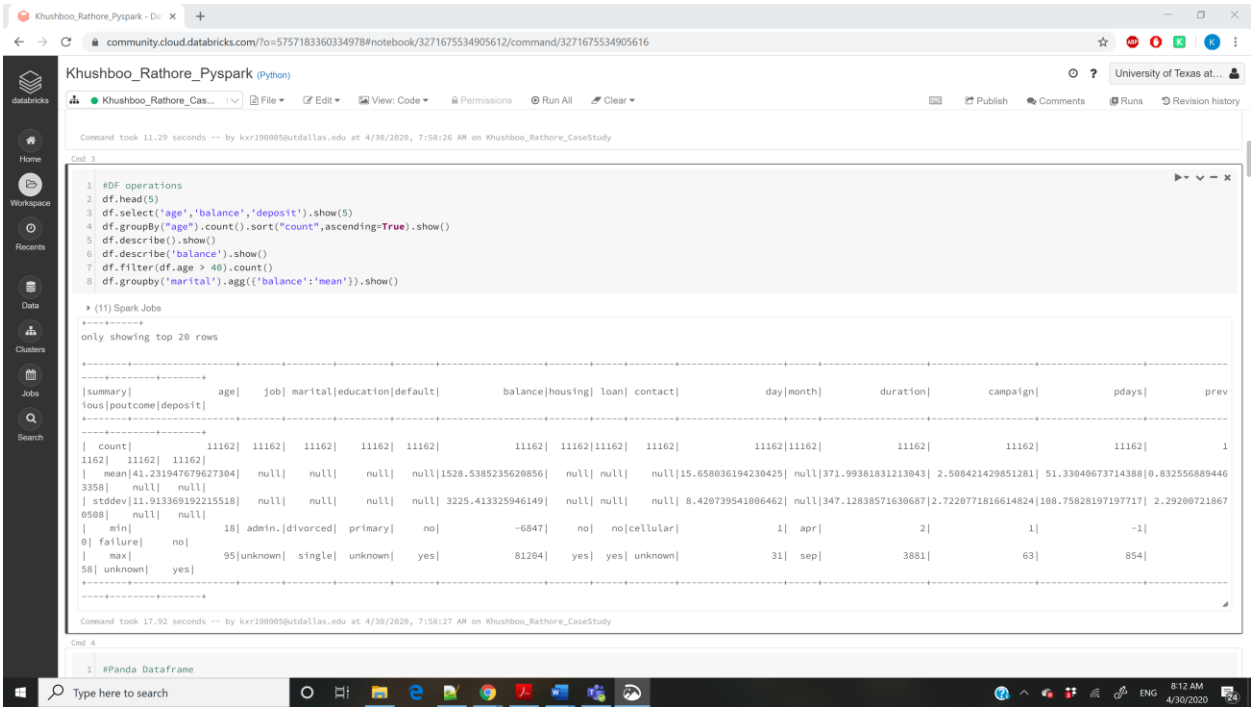
Select statement is to show number of records from defined columns (age, balance, and deposit) in dataframe df.

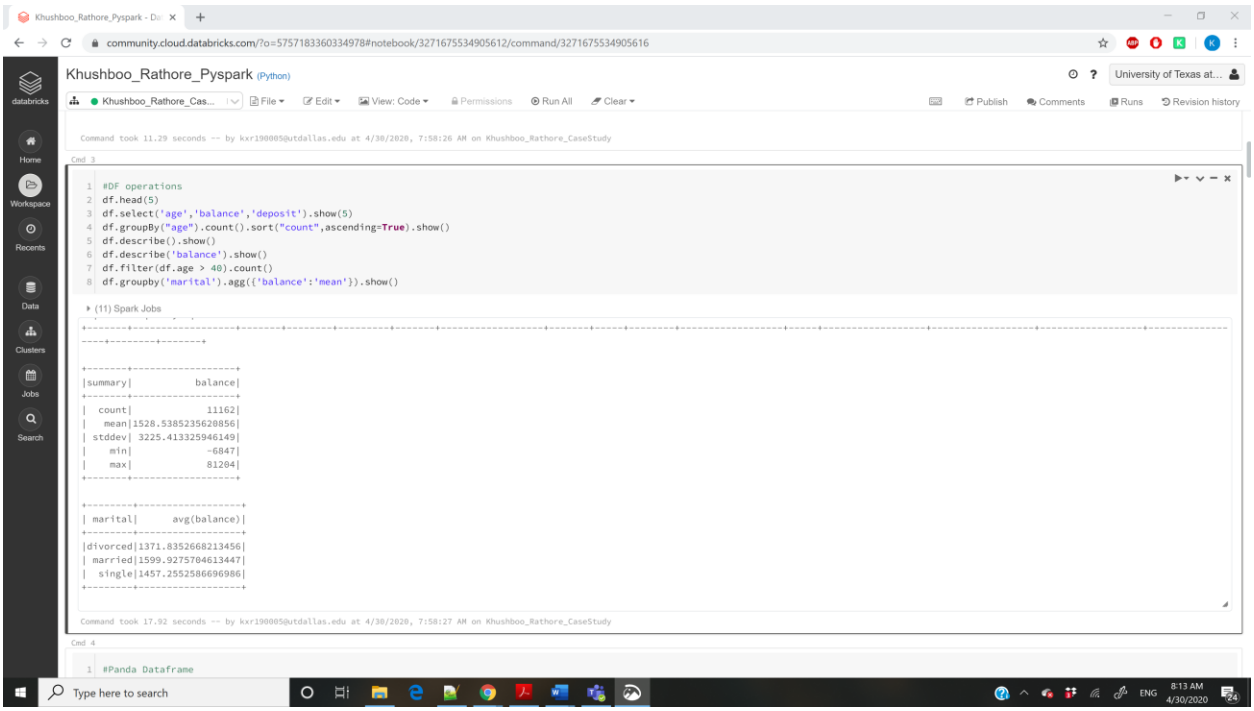
Group by is used to form groups based on values of “age” column for aggregate function “count”. First groupby operation finds count of customers for every age group and sorts them is ascending order and second groupby finds mean of balance for every marital status.

ii. How many customers are there with an age greater than 40? - There are 4967 customers with age > 40, but it is not printed in the output because show() command is not used.

iii. Why are there only 5 records viewable? Due to command show(5)


Khushboo_Rathore_Pyspark - Databricks Community Edition - Mozilla Firefox
Khushboo_Rathore_Pyspark (Python)
#DF operations
df.head(5)
df.select('age','balance','deposit').show(5)
df.groupBy("age").count().sort("count",ascending=True).show()
df.describe().show()
df.describe('balance').show()
df.filter(df.age > 40).count()
df.groupby('marital').agg({'balance':'mean'}).show()
(11) Spark Jobs
+---+-------+-------+
|age|balance|deposit|
+---+-------+-------+
59	2343	yes
56	45	yes
41	1270	yes
55	2476	yes
54	184	yes
+---+-------+-------+
only showing top 5 rows
+---+-----+
|age|count|
+---+-----+
95	1
89	1
90	2
92	2
93	2
88	2
87	4
Command took 17.70 seconds -- by kxr190005@utdallas.edu at 4/21/2020, 7:16:56 PM on Khushboo_Rathore_CaseStudy


Khushboo_Rathore_Pyspark (Python)
Command took 11.29 seconds -- by kxr190005@utdallas.edu at 4/30/2020, 7:58:26 AM on Khushboo_Rathore_CaseStudy
#DF operations
df.head(5)
df.select('age','balance','deposit').show(5)
df.groupBy("age").count().sort("count",ascending=True).show()
df.describe().show()
df.describe('balance').show()
df.filter(df.age > 40).count()
df.groupby('marital').agg({'balance':'mean'}).show()
(11) Spark Jobs
only showing top 20 rows
|summary| age| job| marital|education|default| balance|housing| loan| contact| day|month| duration| campaign| pdays| previous|poutcome|deposit|
|count| 11162| 11162| 11162| 11162| 11162| 11162| 11162|11162| 11162| 11162|11162| 11162| 11162| 11162| 1162| 11162| 11162|
|mean|41.231947679627304| null| null| null| null|1528.5385235620856| null| null| null|15.658036194230425| null|371.99381831213043| 2.508421429851281| 51.33040673714388|0.8325568894463358| null| null|
|stddev|11.913369192215518| null| null| null| null| 3225.413325946149| null| null| null| 8.420739541006462| null|347.12838571630687|2.7228771816614824|108.75828197197717| 2.2920072186708508| null| null|
|min| 18| admin.|divorced| primary| no| -6847| no| no|cellular| 1| apr| 2| 1| -1| 0| failure| no|
|max| 95|unknown| single| unknown| yes| 81204| yes| yes| unknown| 31| sep| 3881| 63| 854| 58| unknown| yes|
Command took 17.92 seconds -- by kxr190005@utdallas.edu at 4/30/2020, 7:58:27 AM on Khushboo_Rathore_CaseStudy
#Panda Dataframe

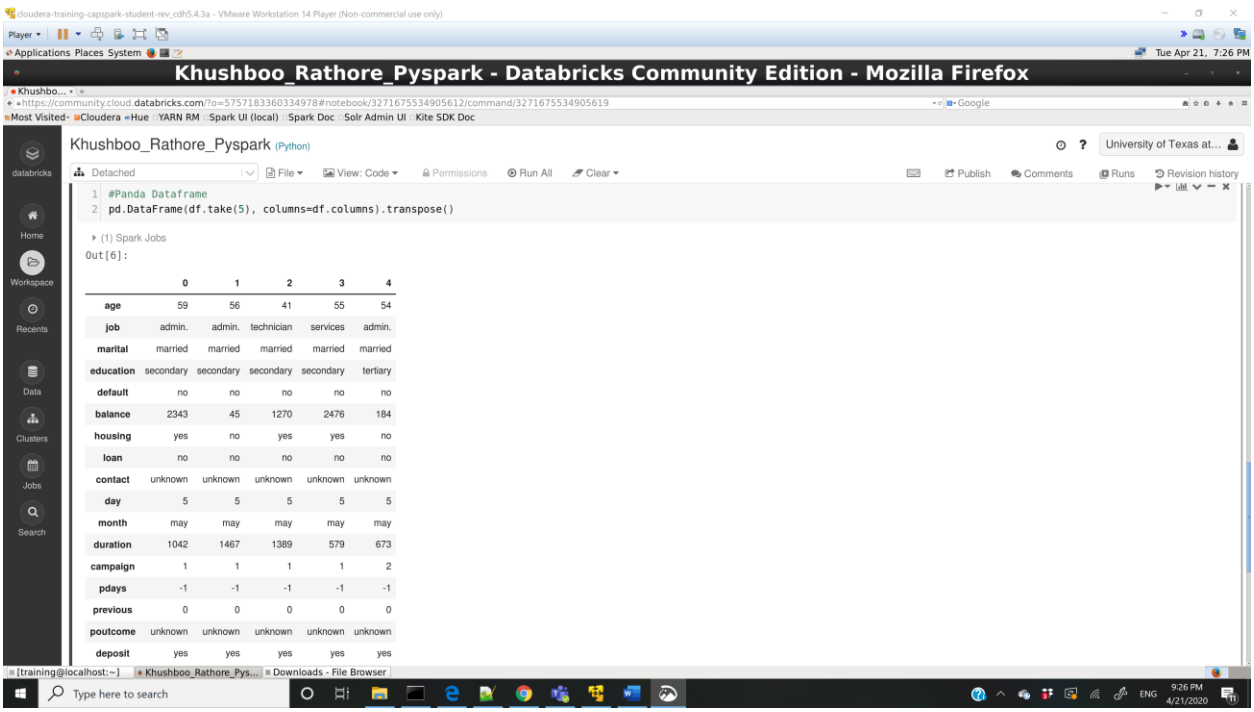


Step 14: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

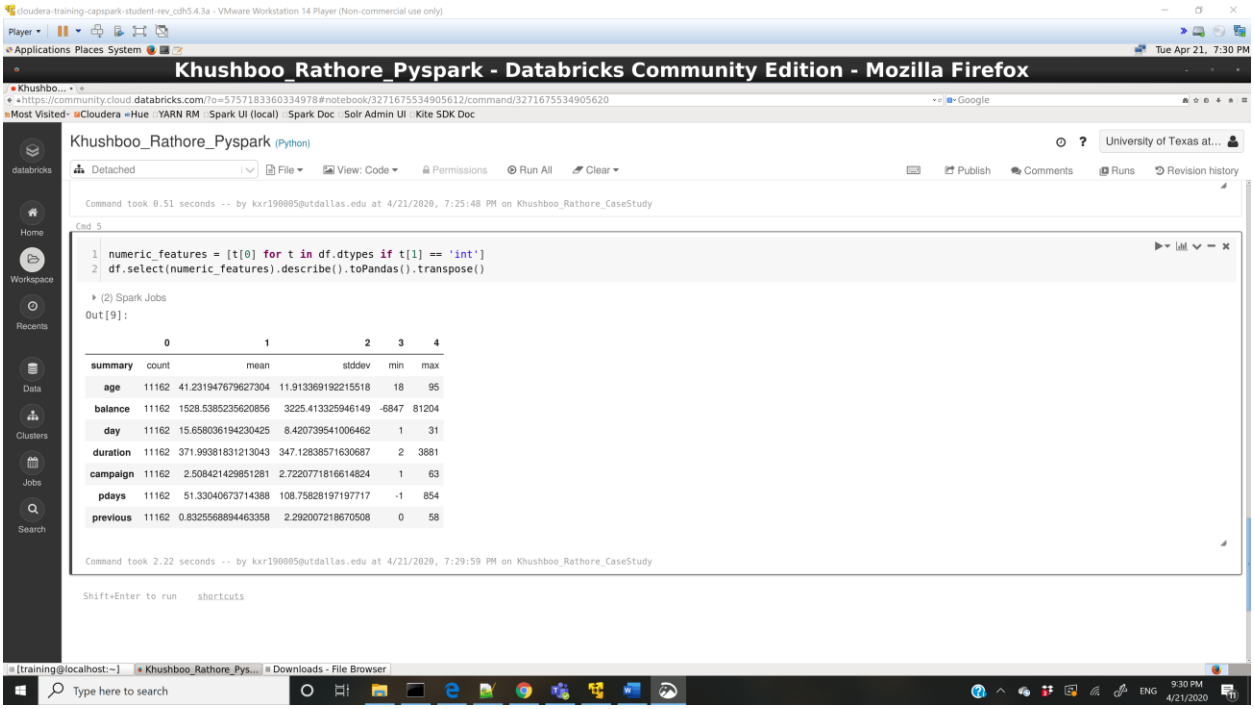
What is seen in the output of the transpose?

The 5 rows are now converted into columns and the columns are now shown as rows



Step 15: Execute the following code:

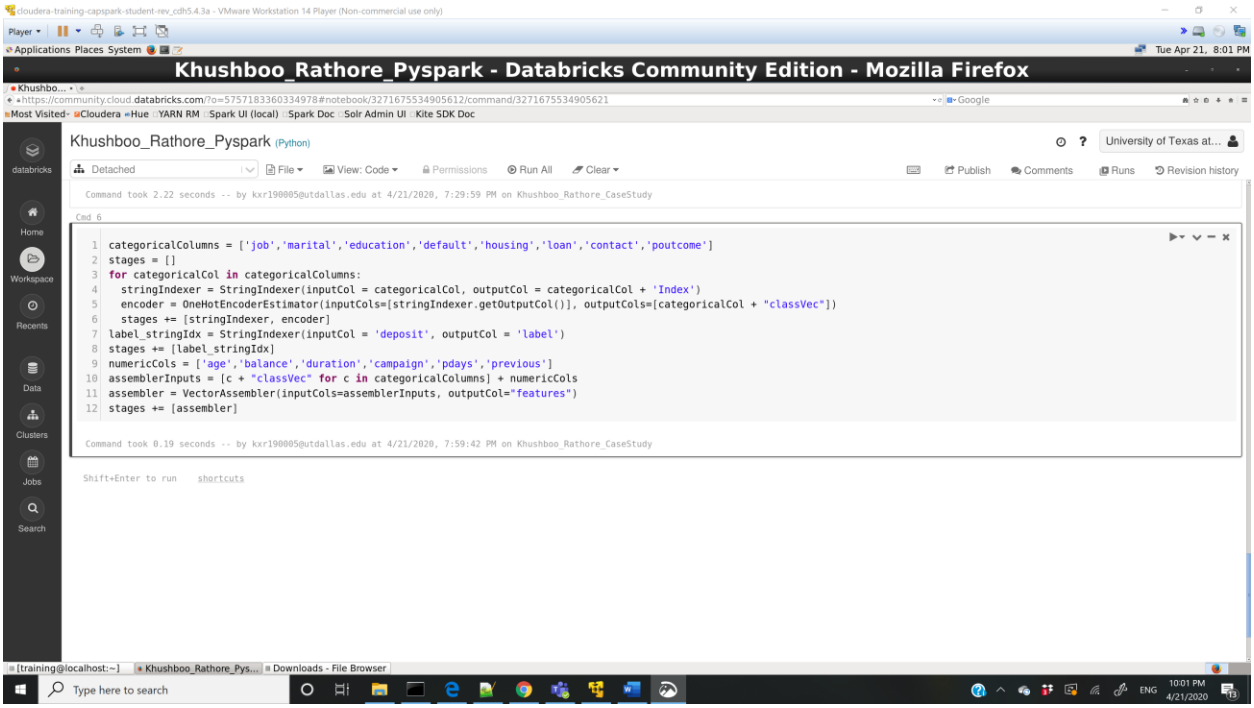
Q. Take a screenshot of the output of the code and paste it below.



Part 5: Data Preprocessing:

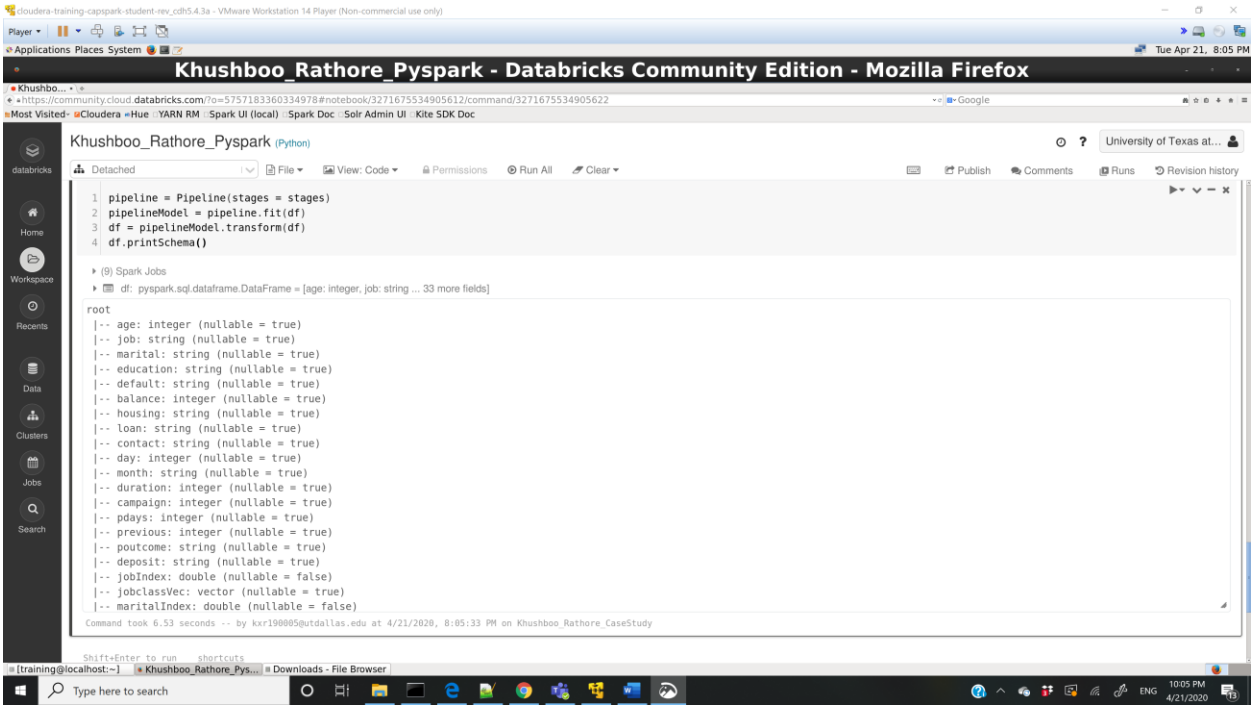
Step 16: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.



Step 17: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

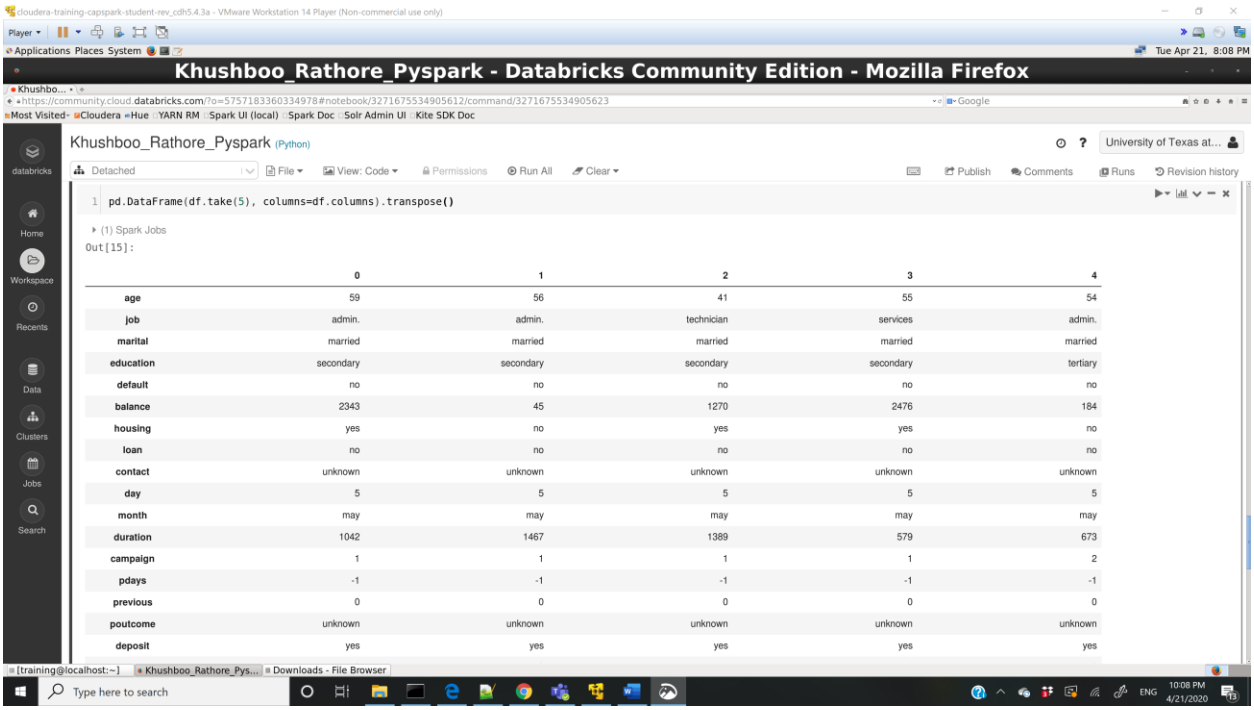


Step 18: Execute the following command:

Q. Take a screenshot of the output of the code and paste it below.

Define the purpose of the transpose function?

To represent the label names of the columns as rows and their corresponding values can be seen as columns for 5 records.

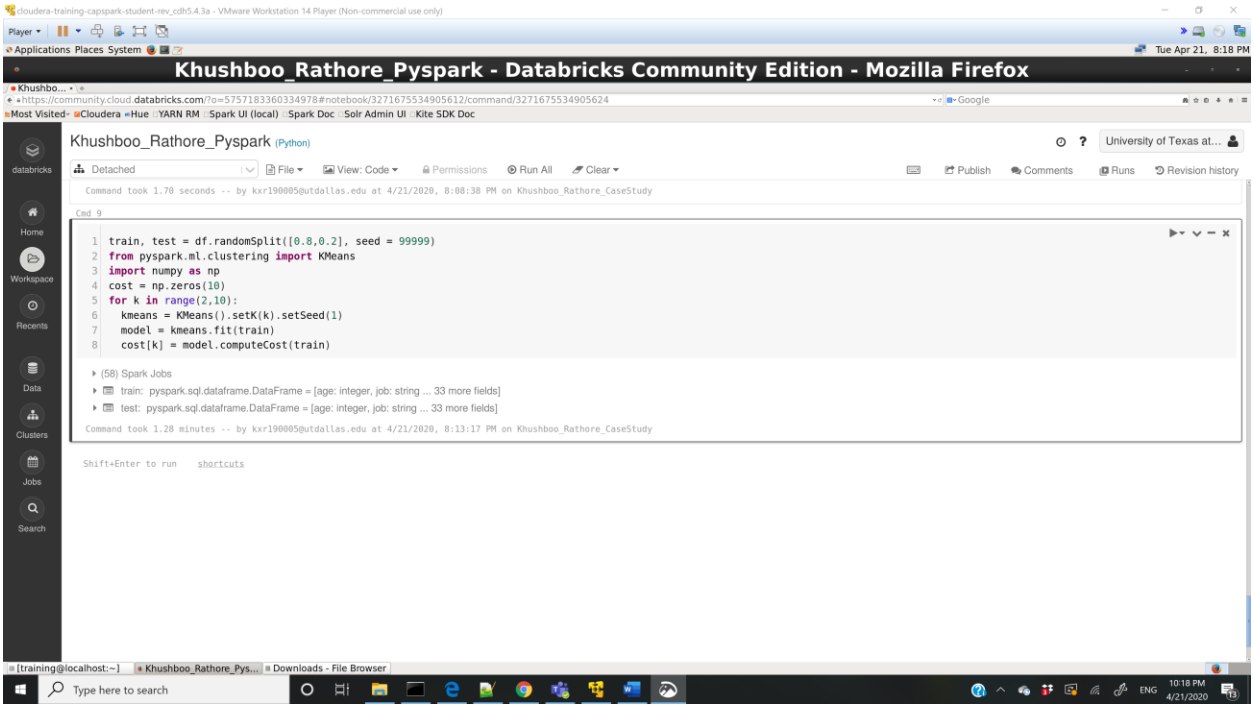


	0	1	2	3	4
age	59	56	41	55	54
job	admin.	admin.	technician	services	admin.
marital	married	married	married	married	married
education	secondary	secondary	secondary	secondary	tertiary
default	no	no	no	no	no
balance	2343	45	1270	2476	184
housing	yes	no	yes	yes	no
loan	no	no	no	no	no
contact	unknown	unknown	unknown	unknown	unknown
day	5	5	5	5	5
month	may	may	may	may	may
duration	1042	1467	1389	579	673
campaign	1	1	1	1	2
pdays	-1	-1	-1	-1	-1
previous	0	0	0	0	0
poutcome	unknown	unknown	unknown	unknown	unknown
deposit	yes	yes	yes	yes	yes

Part 6: Implementation of k-means algorithm

Step 19: Execute the following code:

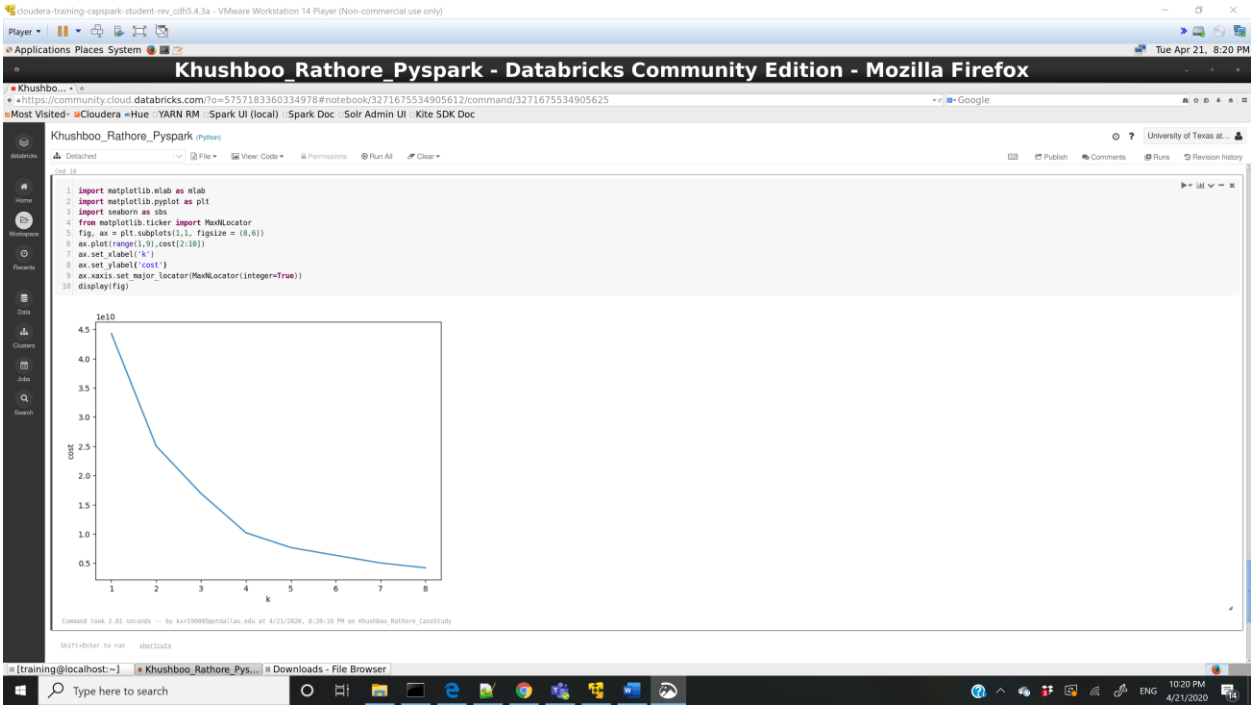
Q. Take a screenshot of the output of the code and paste it below.



Step 20: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

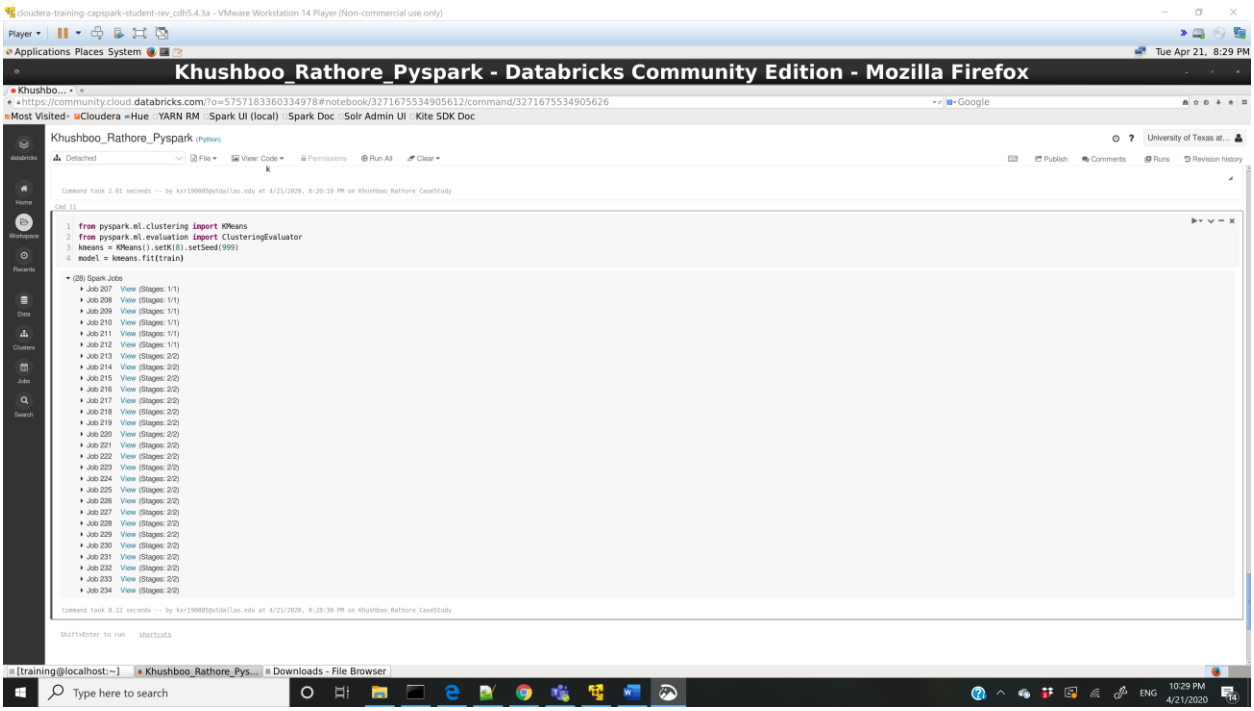
The graph plotted is an **Elbow Plot** of number of clusters vs cost which can be used to determine the optimum number of clusters to be used on the dataset to get best result. Here the elbow is at k = 2, so we should get the best model at 2 clusters.



Part 7: Evaluate Model

Step 21: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

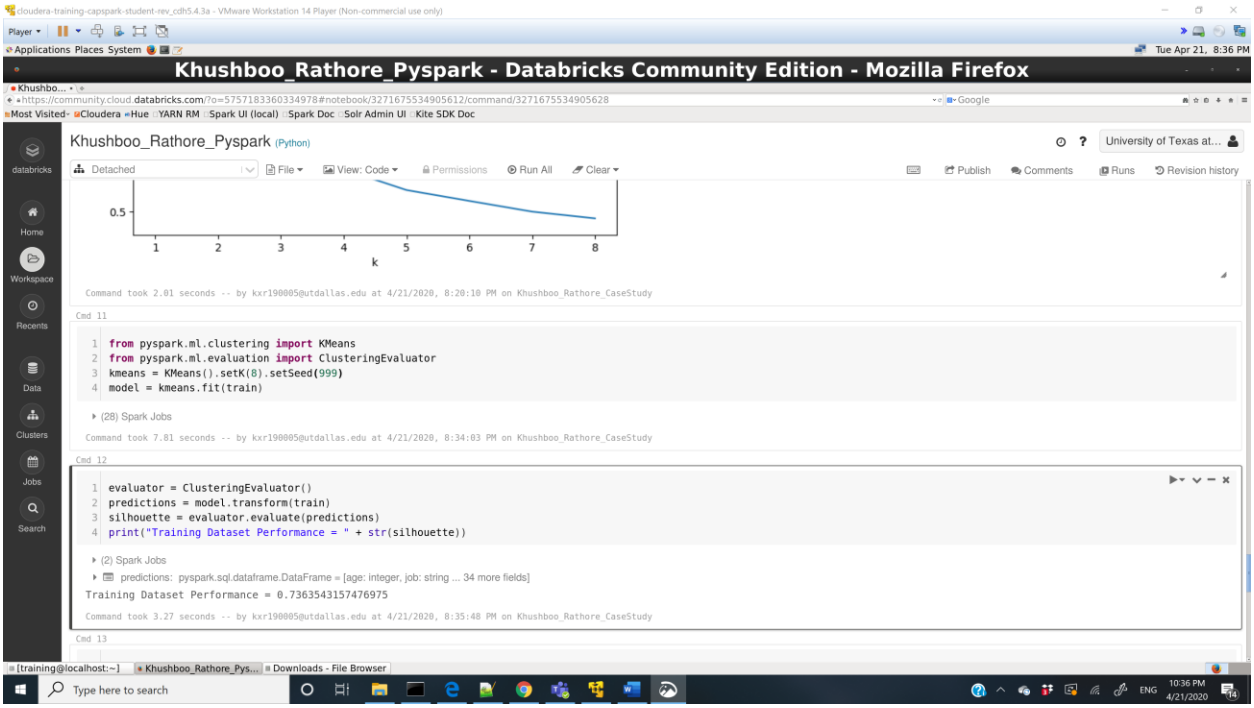


Step 22: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

COMBINING STEP 21 AND STEP 22 FOR TRAIN DATASET

KMeans().setK(8).setSeed(999) : Performance = 0.74 (approx)

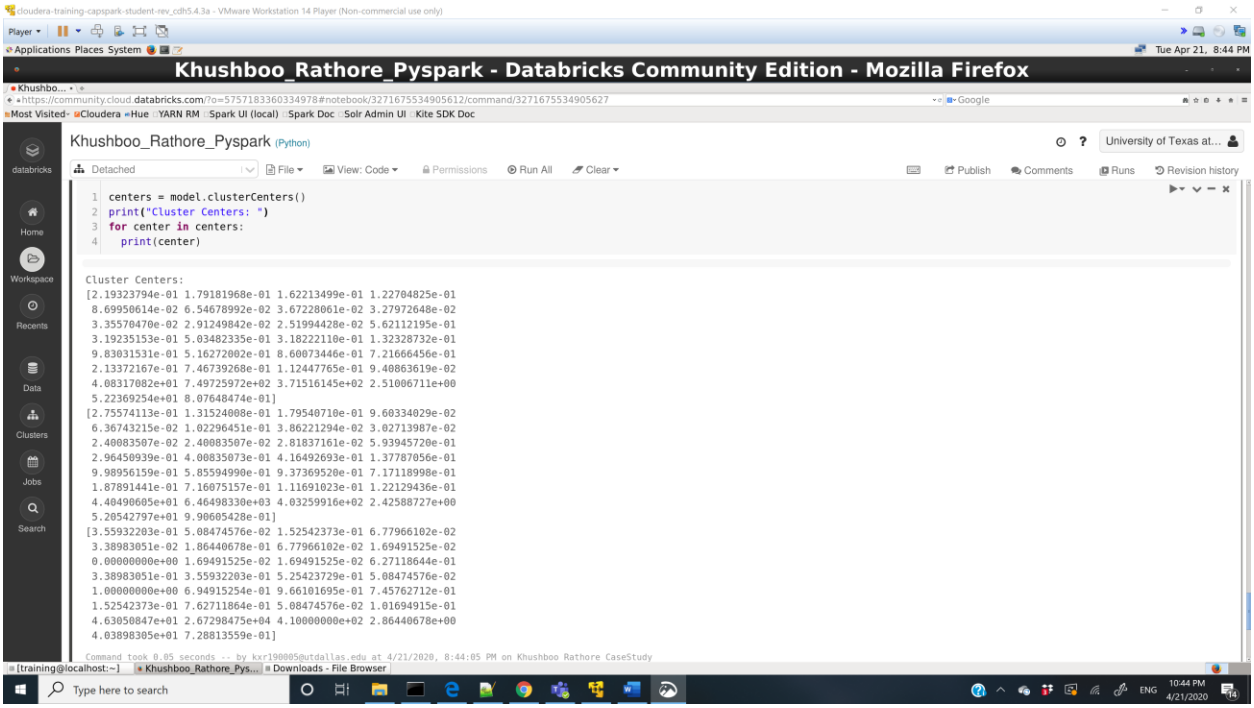


Step 23: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.

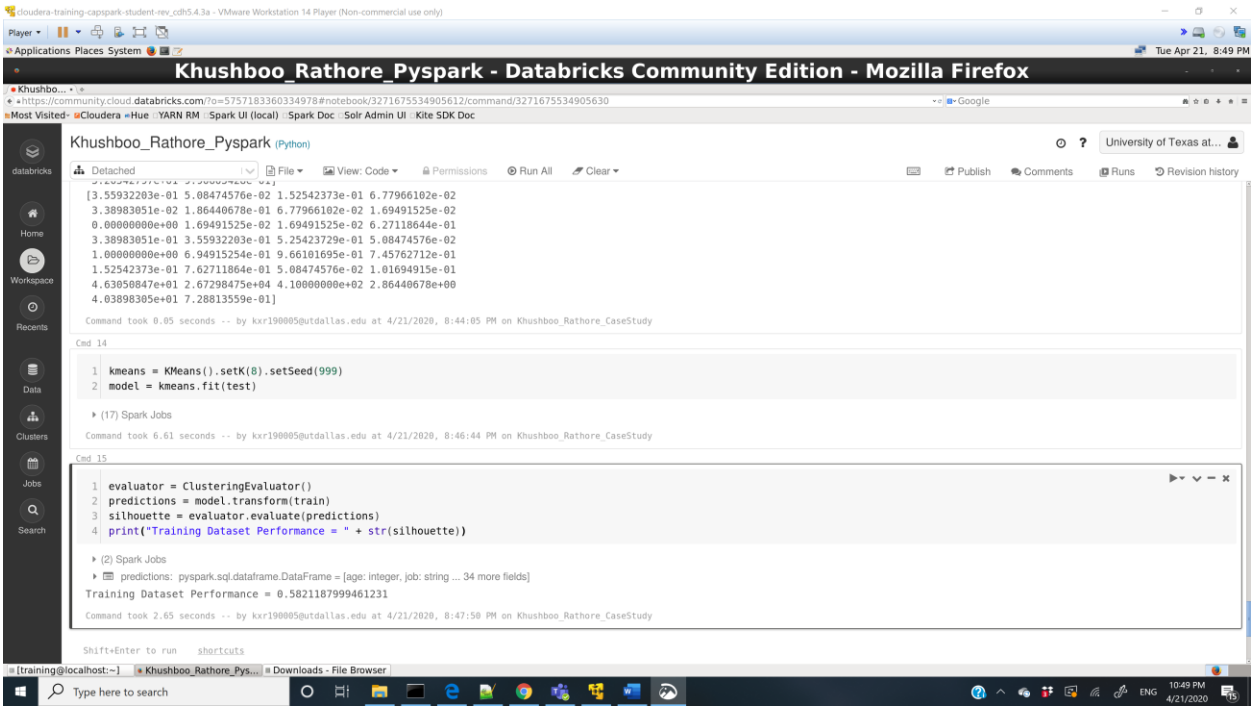
What is Centroid and how is it calculated in K-means?

Centroid is the center of the cluster.
In K Means, as per the value of K (number of clusters) specified by the user, number of clusters are randomly assigned to different points in the dataset and the unassigned points are then "assigned" to a cluster based on their proximity to the previously assigned point using Euclidean distance.
Once assignment completes, the new centroid values of the cluster is computed by calculating the mean values and again the values of Euclidean distance is calculated from the new centriods. This process repeats until we find a constant value for centroids and the latest cluster will be considered as the final cluster solution.



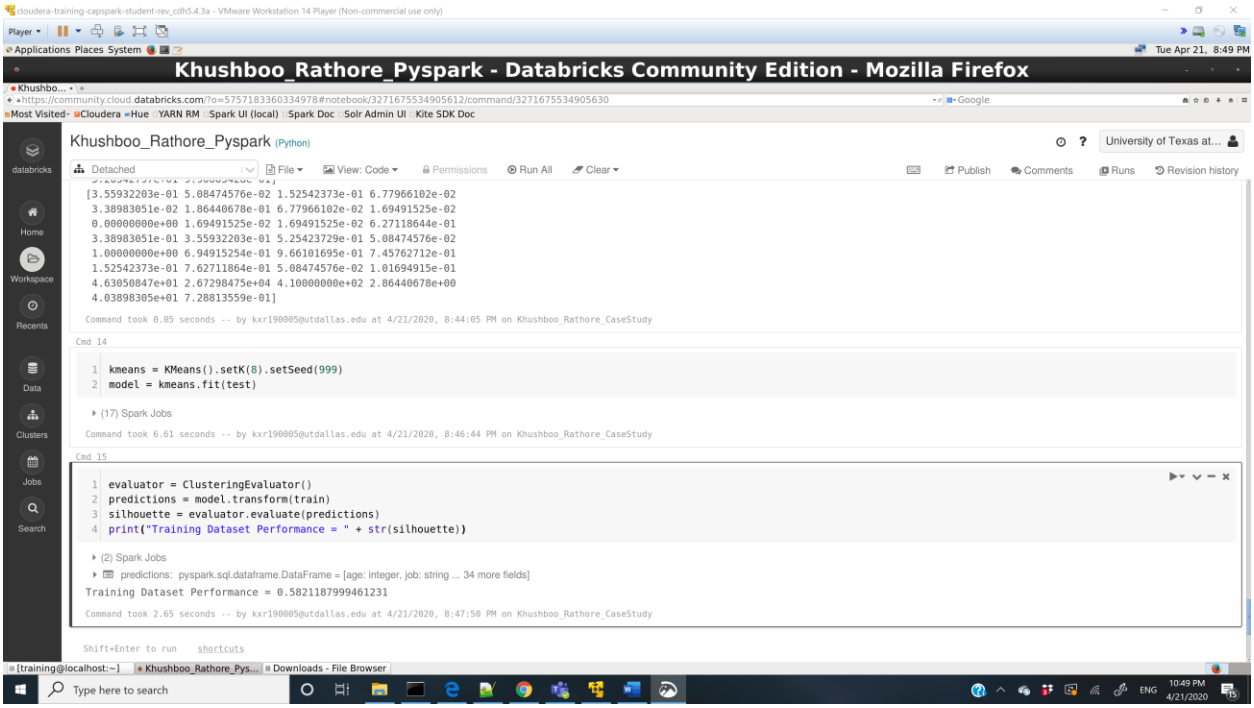
Step 24: Execute the following code:

Q. Take a screenshot of the output of the code and paste it below.



Step 25: Execute the following code:

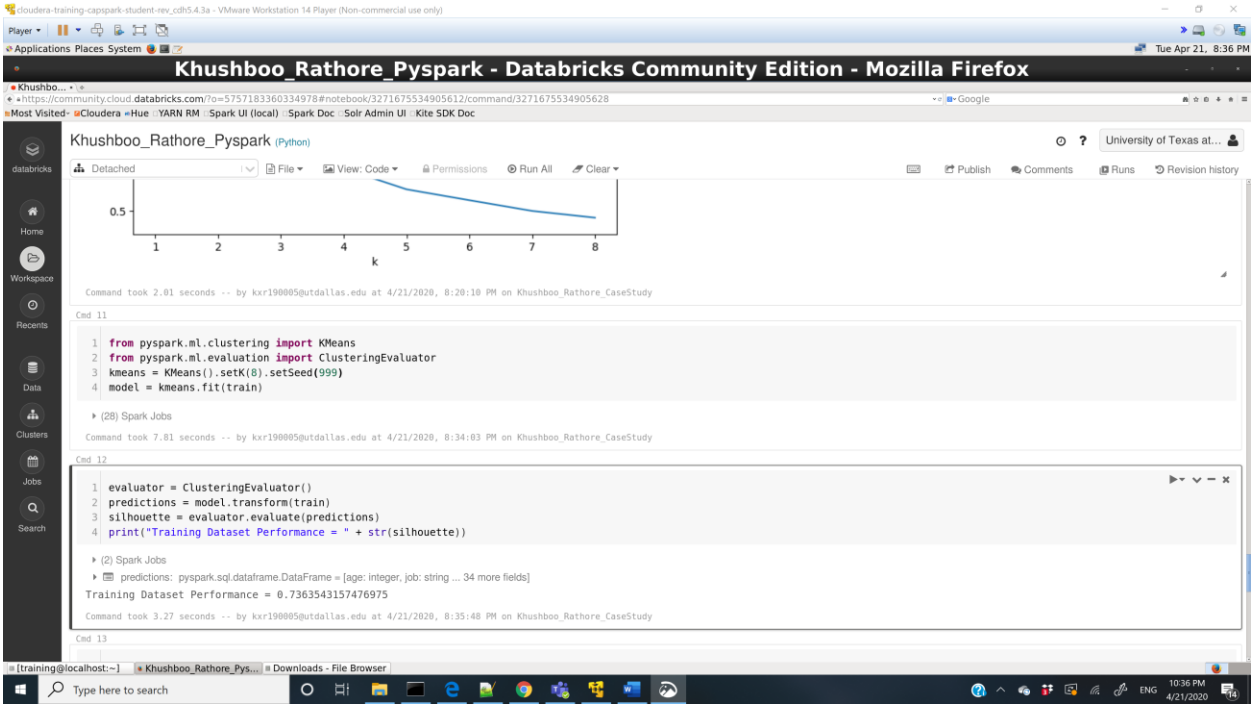
Q. Take a screenshot of the output of the code and paste it below.



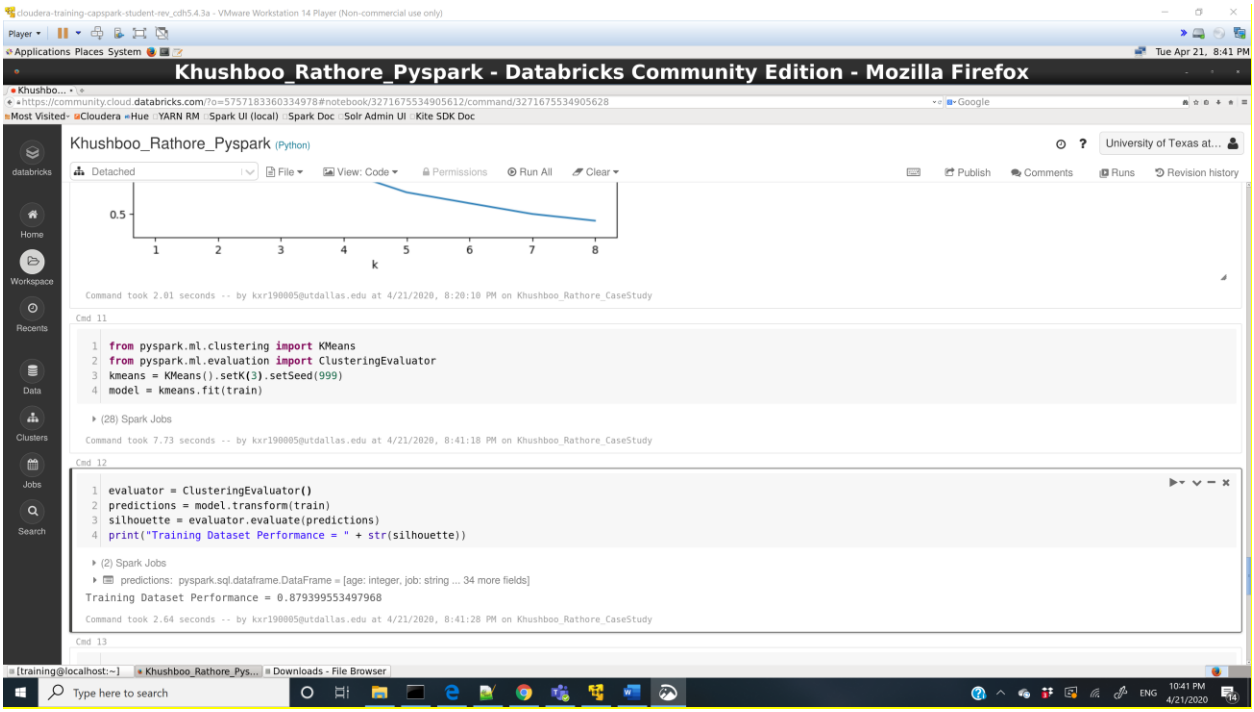
Copy the Train and Test performance results of all the clusters here. Compare the results and comment about the performance of each model. Select your best model and explain why you chose that number of clusters as the optimal level.

TRAIN - COMBINING STEP 21 AND STEP 22 FOR TRAIN DATASET

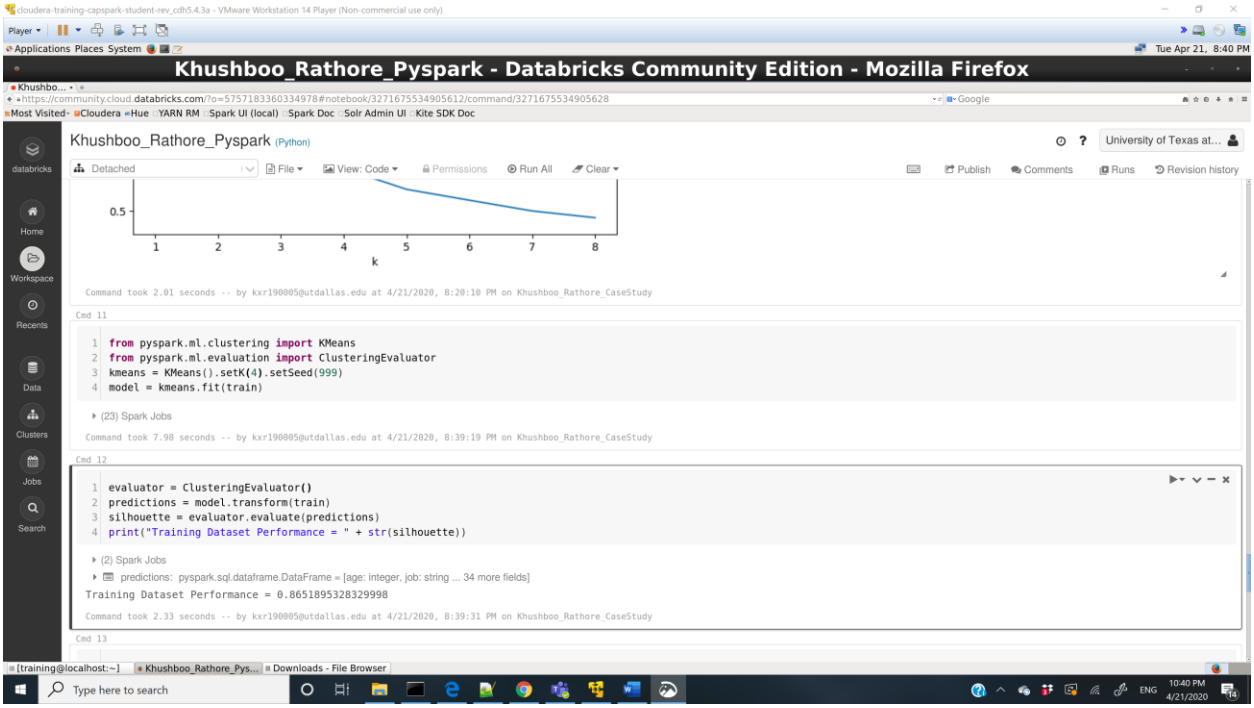
KMeans().setK(8).setSeed(999) : Performance = 0.74 (approx)



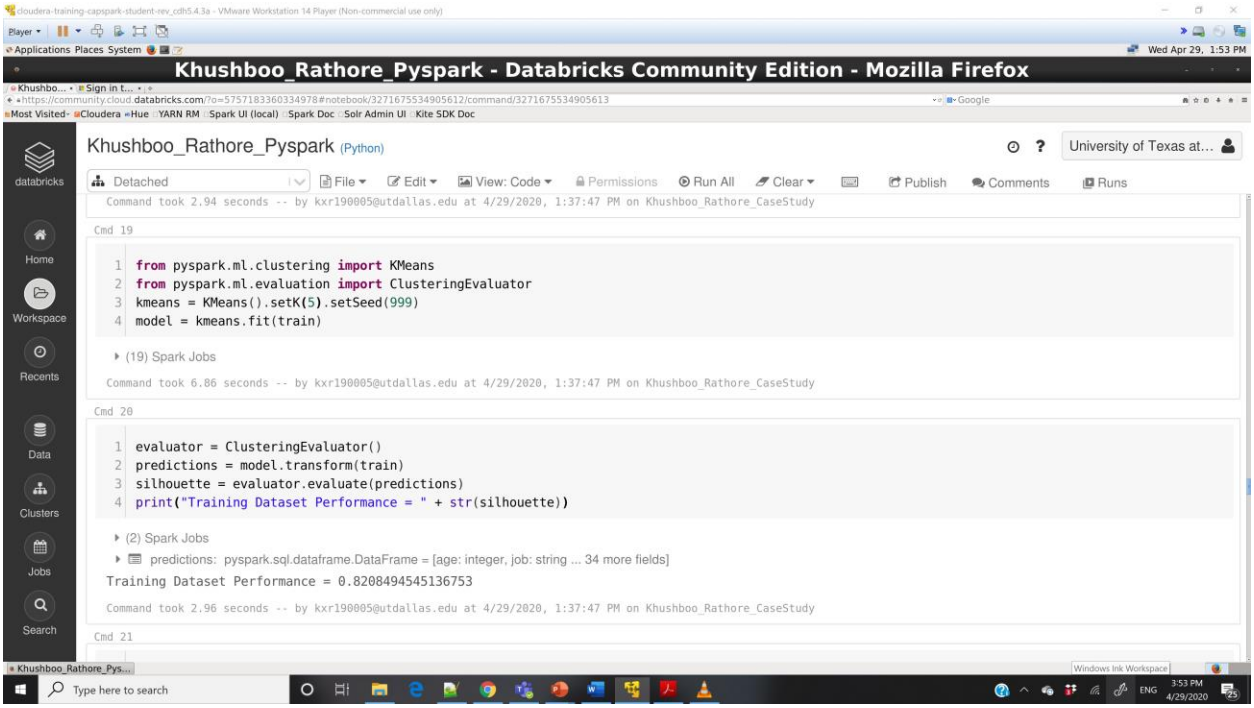
KMeans().setK(3).setSeed(999) : Performance = 0.88 (approx)



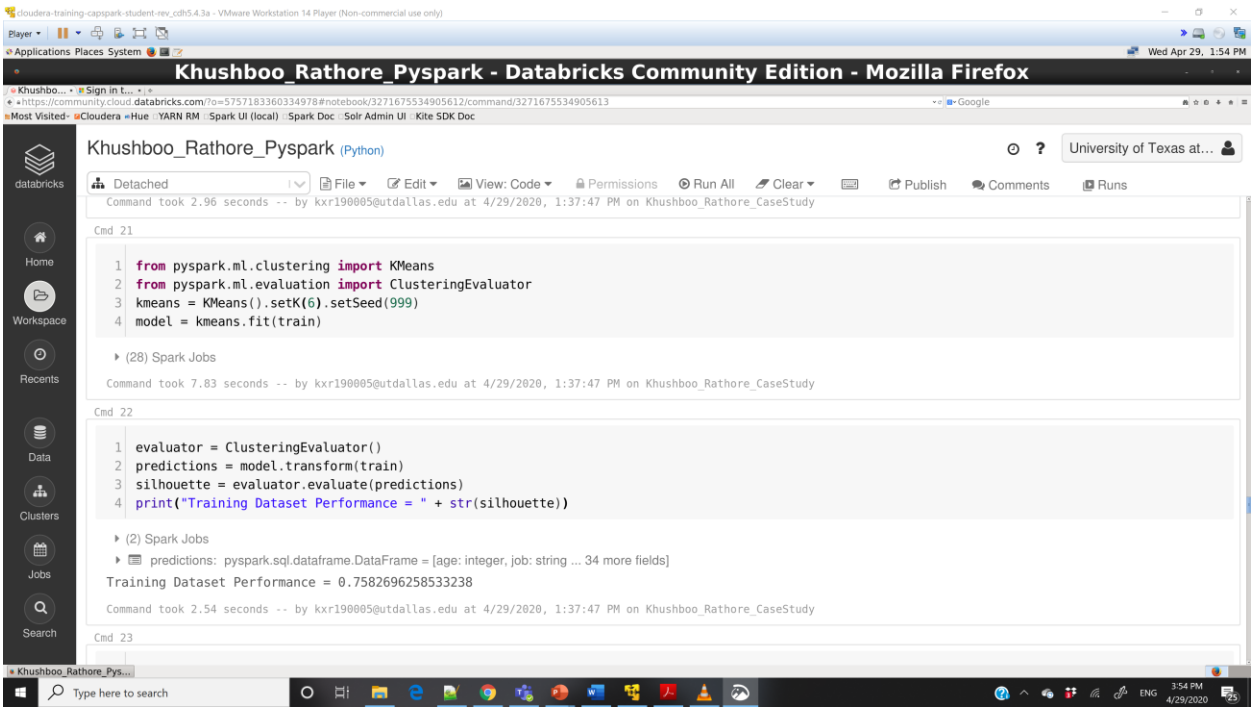
KMeans().setK(4).setSeed(999) : Performance = 0.86 (approx)



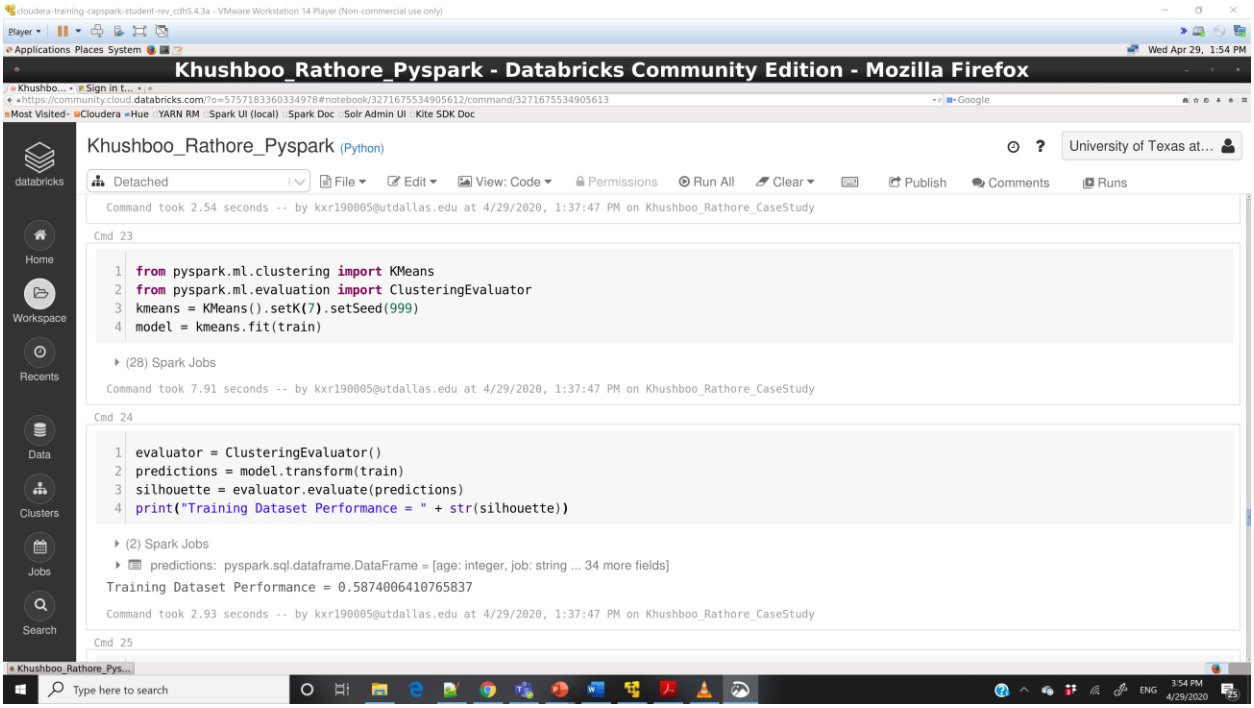
KMeans().setK(5).setSeed(999) : Performance = 0.82 (approx)



KMeans().setK(6).setSeed(999) : Performance = 0.76 (approx)

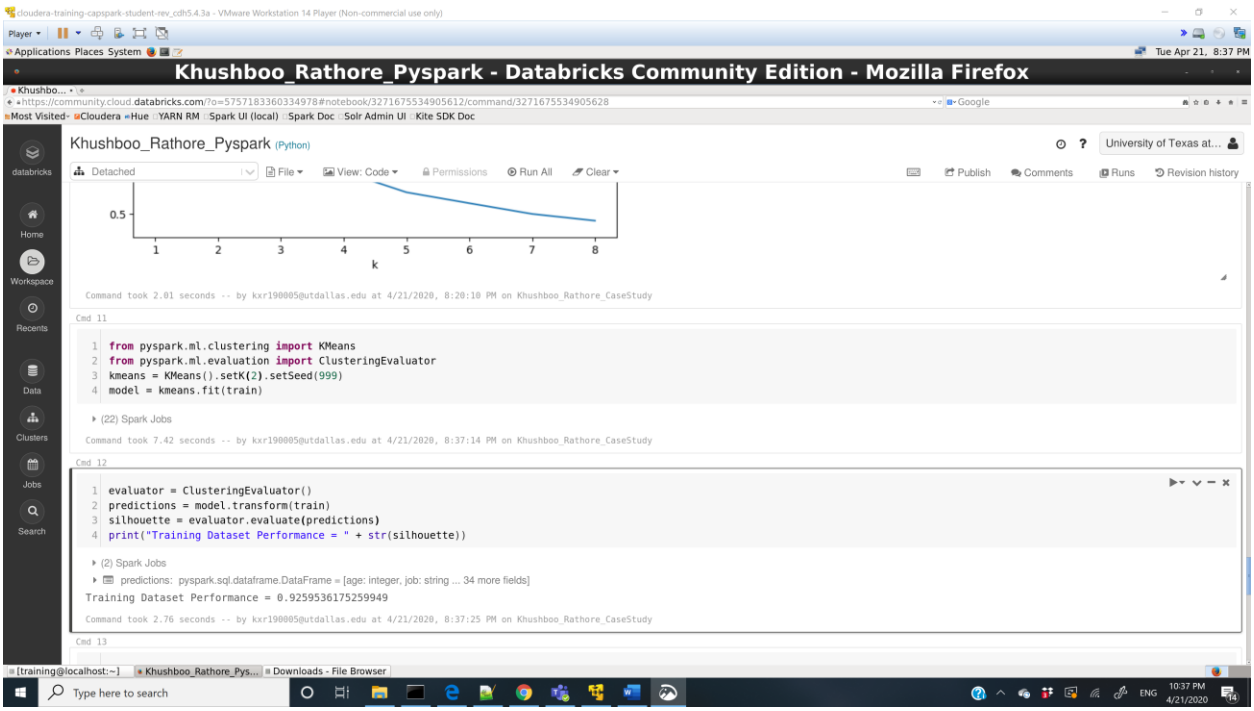


KMeans().setK(7).setSeed(999) : Performance = 0.59 (approx) which is the worst score received in the performance results of all the clusters.



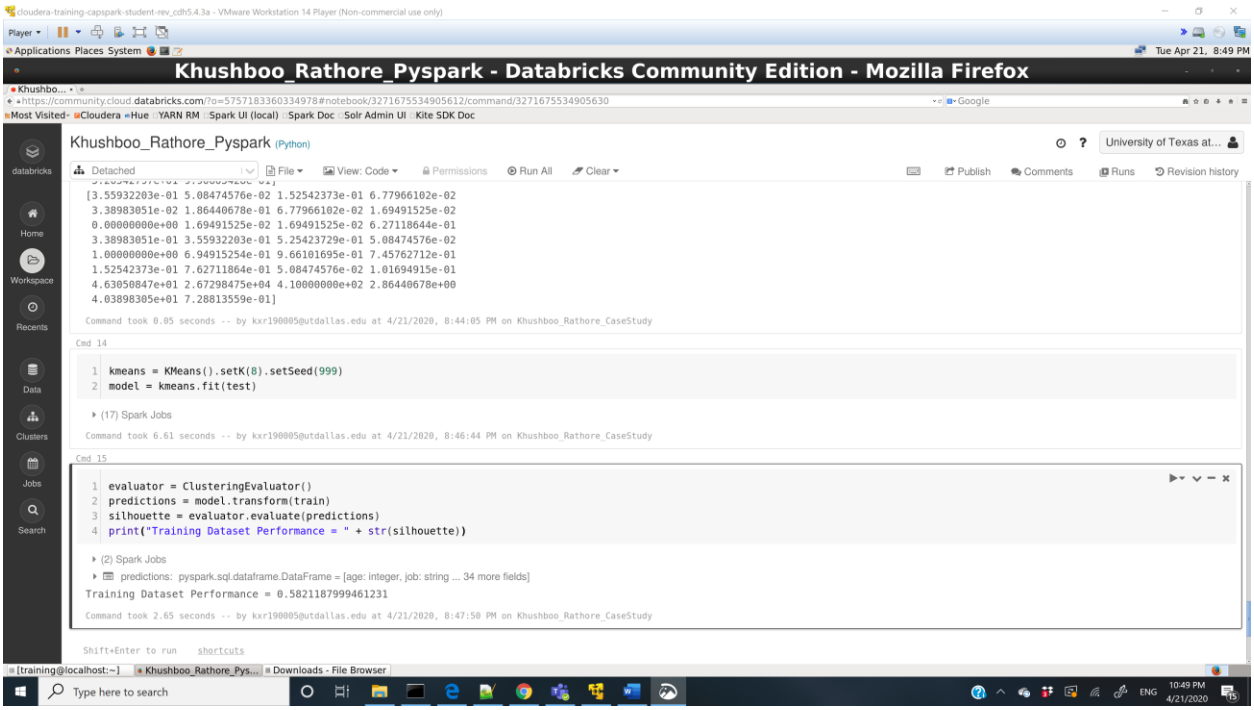
Best Model – From the Elbow plot we can see that the best model would be for 2 number of clusters as it best fits the data for Train dataset and has the highest score. By increasing the number of clusters, the training performance of the model is dropping continuously.

KMeans().setK(2).setSeed(999) : Performance = 0.93 (approx)

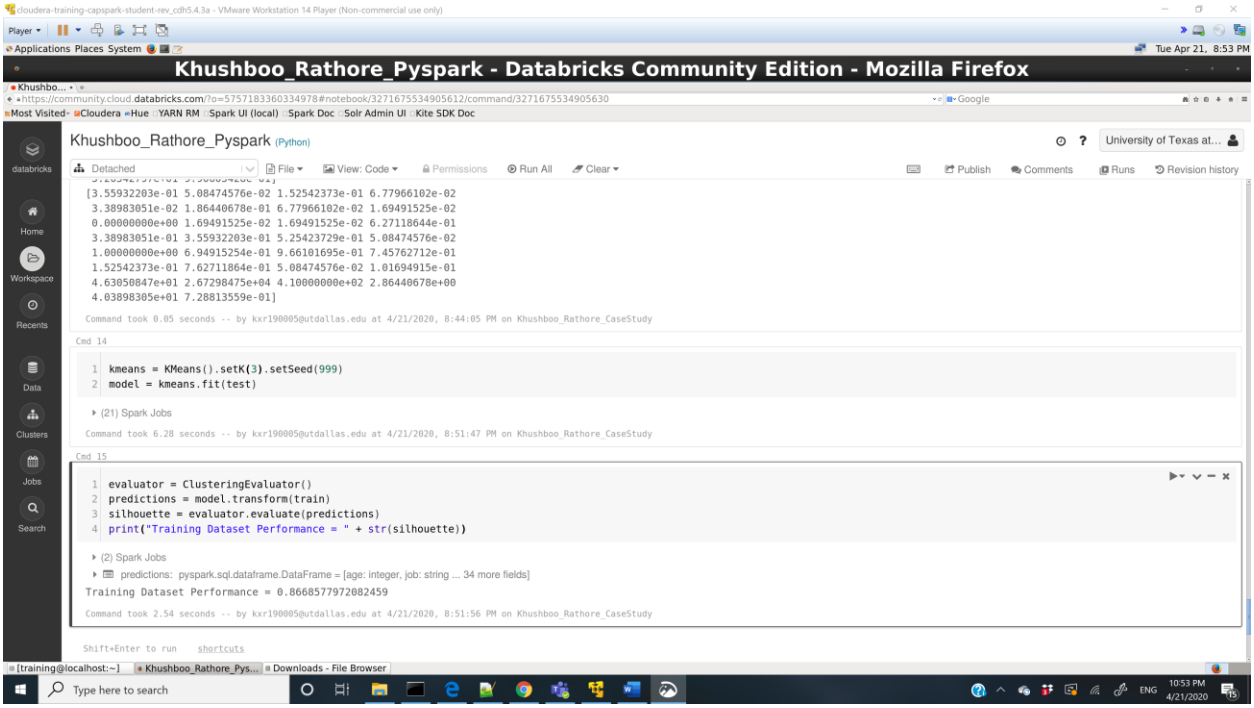


TEST - COMBINING STEP 24 AND STEP 25 FOR TRAIN DATASET

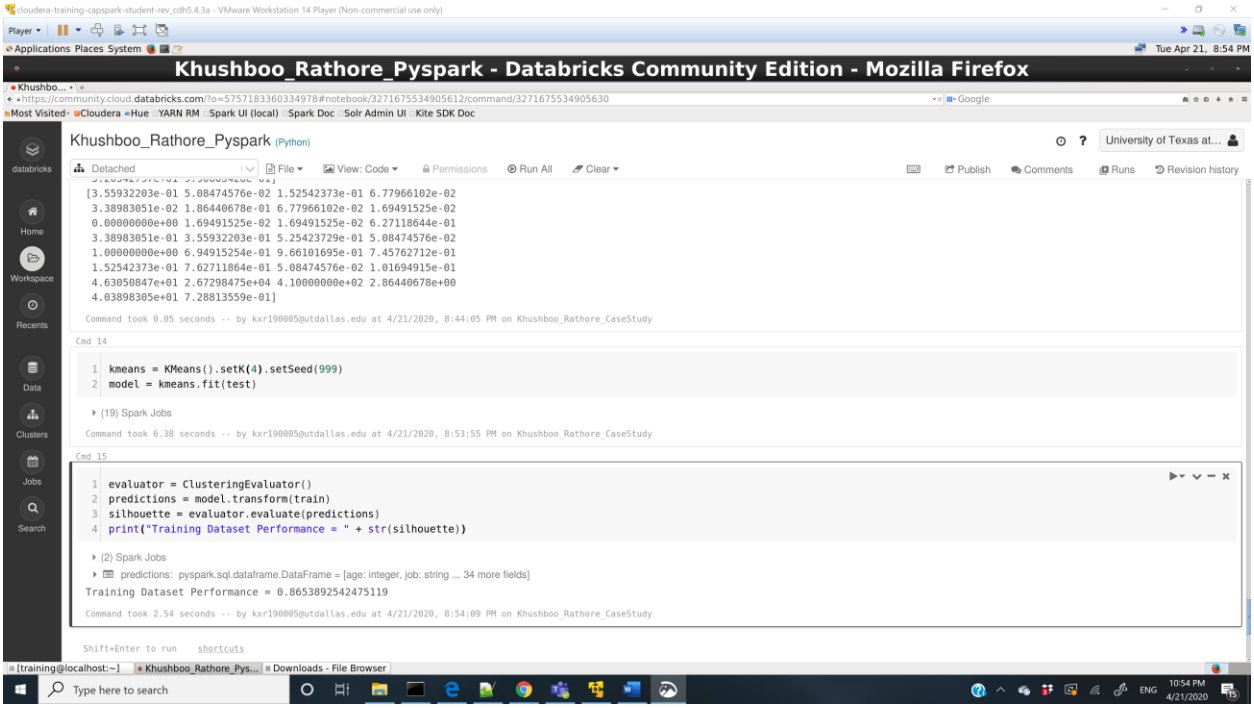
KMeans().setK(8).setSeed(999) : Performance = 0.58 (approx) which is the worst score received in the performance results of all the clusters.



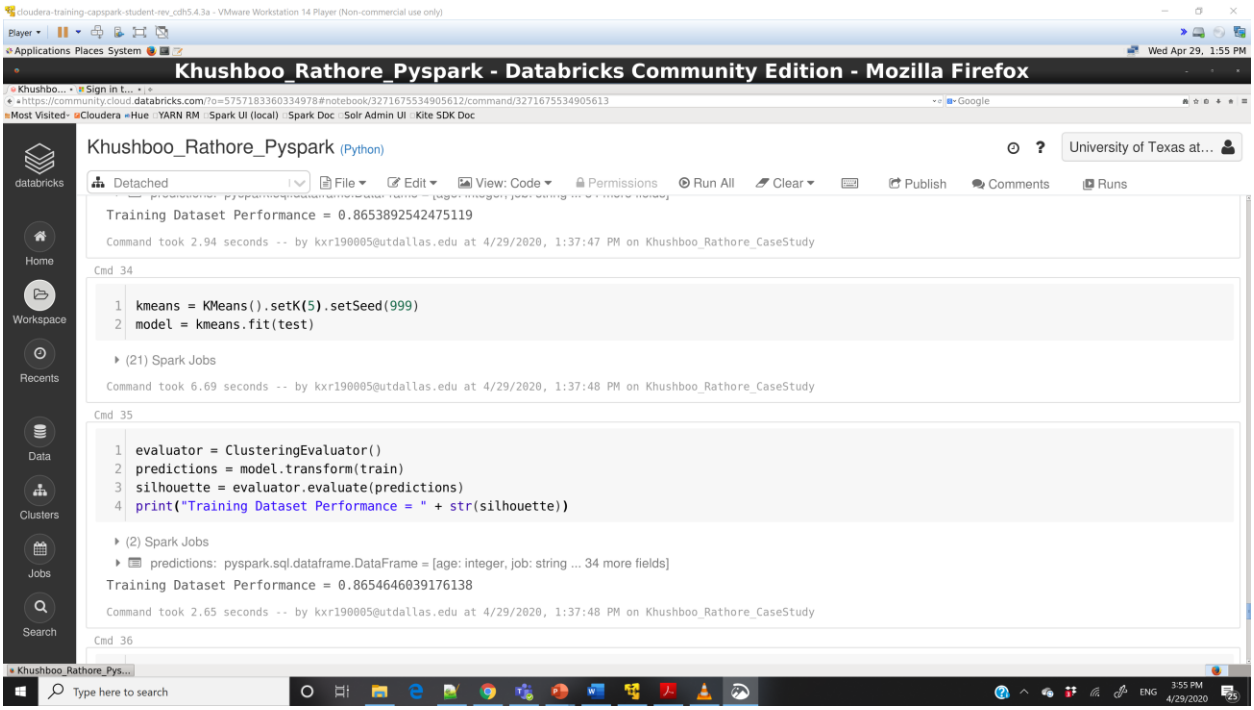
KMeans().setK(3).setSeed(999) : Performance = 0.87 (approx)



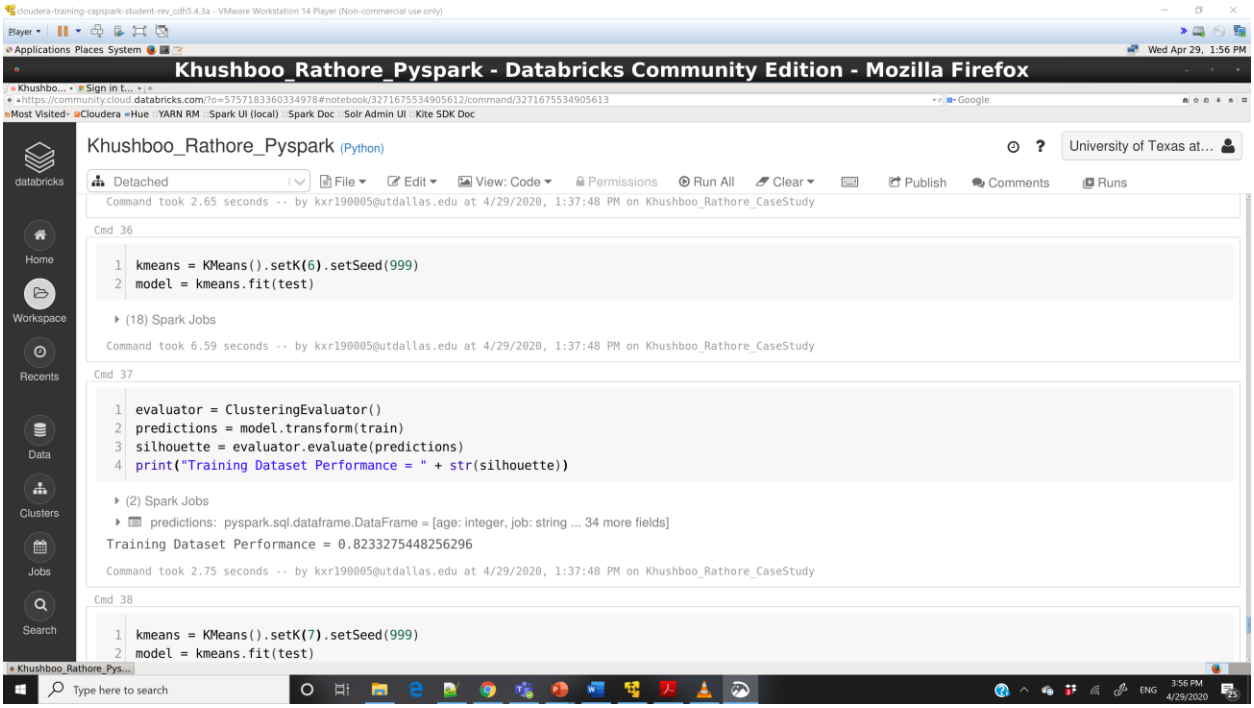
KMeans().setK(4).setSeed(999) : Performance = 0.87 (approx)



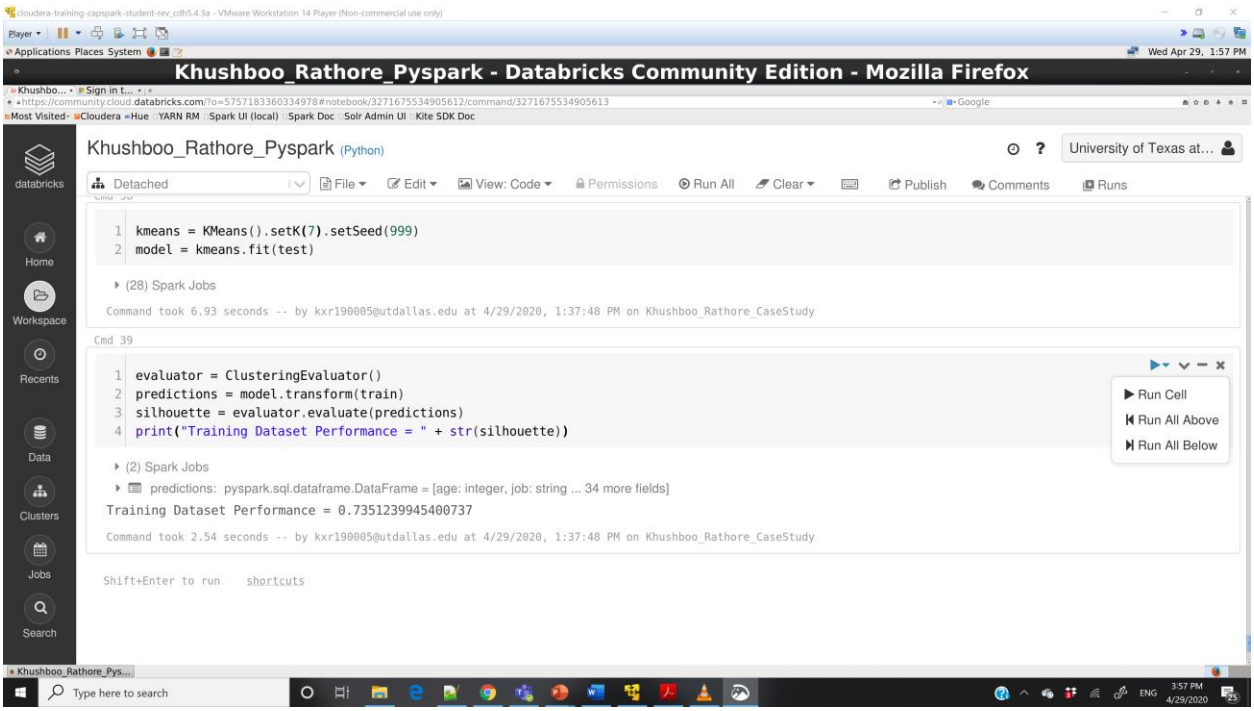
KMeans().setK(5).setSeed(999) : Performance = 0.87 (approx)



KMeans().setK(6).setSeed(999) : Performance = 0.82 (approx)

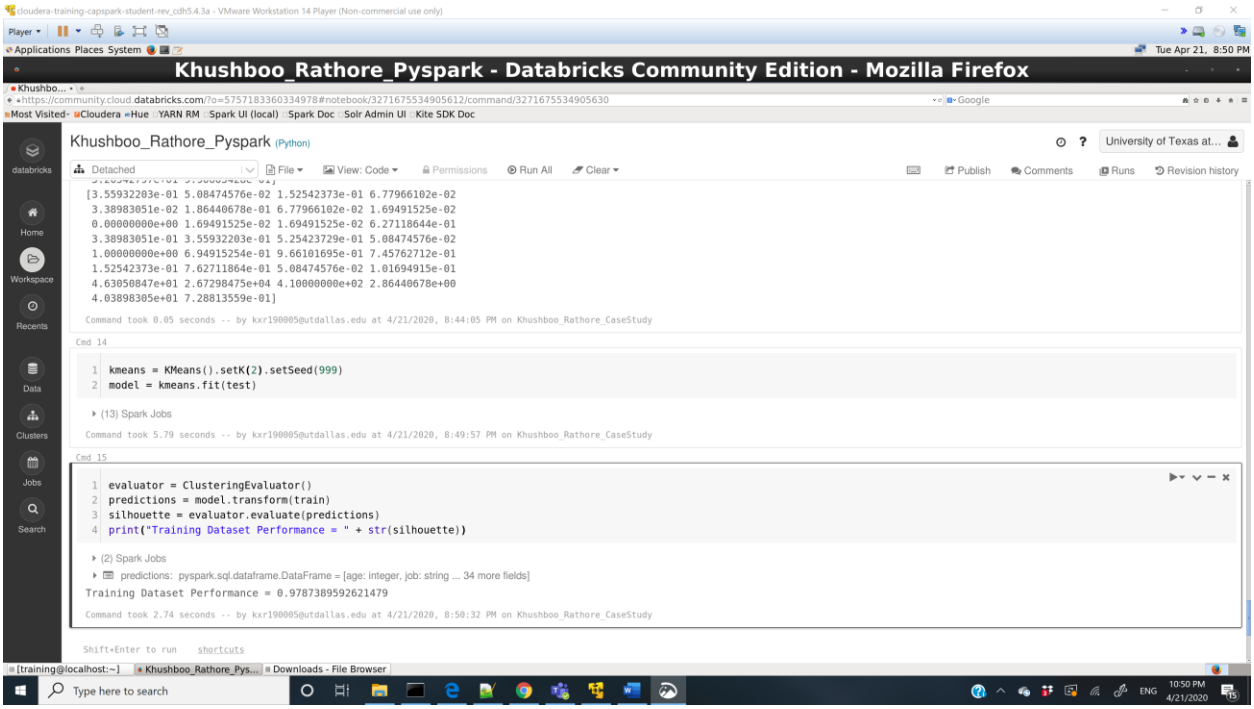


KMeans().setK(7).setSeed(999) : Performance = 0.73 (approx)



Best Model – From the Elbow plot we can see that the best model would be for 2 number of clusters as it best fits the data for Test dataset and has the highest score. By increasing the number of clusters, the testing performance of the model is dropping continuously.

KMeans().setK(2).setSeed(999) : Performance = 0.98 (approx)



Hence, we can conclude that the Best Model would be when number of cluster k = 2 as it gives the highest dataset performance for both Train and Test Dataset and the model generalizes well. By increasing the number of clusters the train and test set performance is decreasing.

Hence 2 is the optimal number of clusters for this model.

Step 26:

Export the file as “HTML” and save that with your “Firstname_Lastname” naming convention.

Convert this HTML file to PDF (This is your 2nd submission file)

Export as “IPNYB” file. (This is your 3rd submission file)