Machine Learning with Pyspark

Student Name: Prem Kumar Pulluri **Student Id: 2021470494**

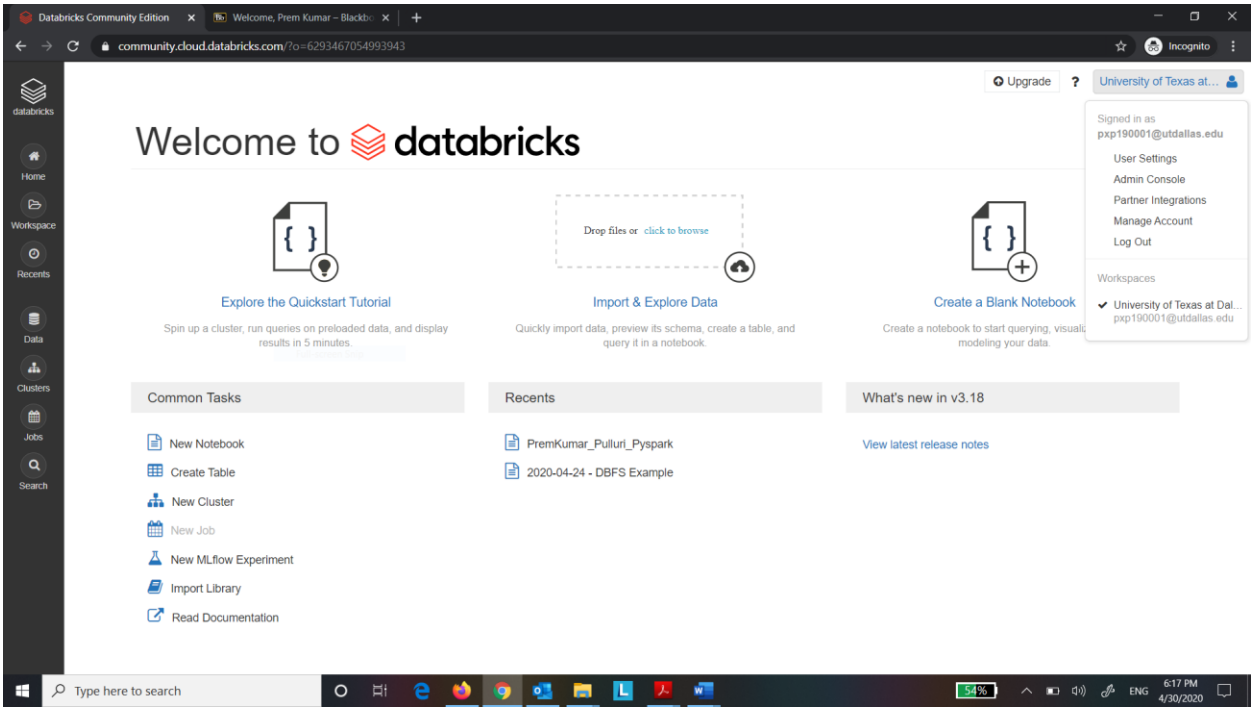
Date: 04/30/2020

Objective:

PySpark is a great language for performing exploratory data analysis and building machine learning pipelines. MLlib is a Spark implementation of useful machine learning (ML) functionality. The objective of this case study is to become familiar with Pyspark. We will load the data in Pyspark and build a machine learning model using the K-Means algorithm.

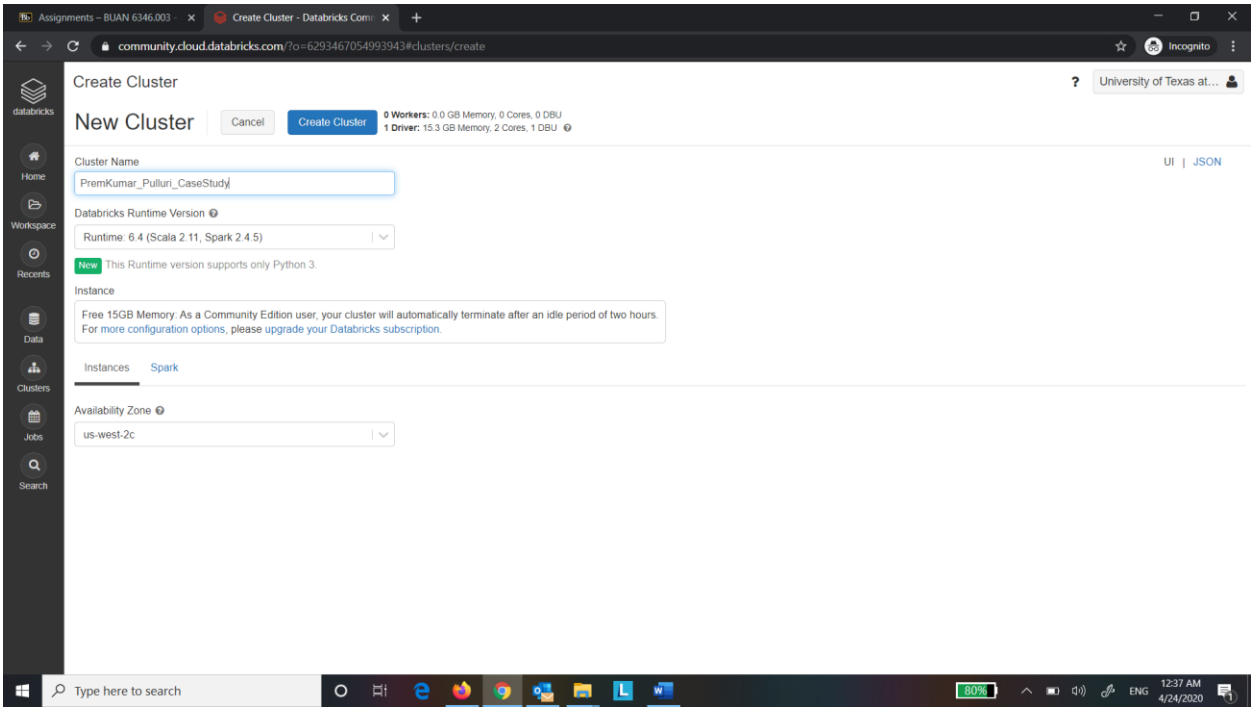
Part 1: Create a databricks account.

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



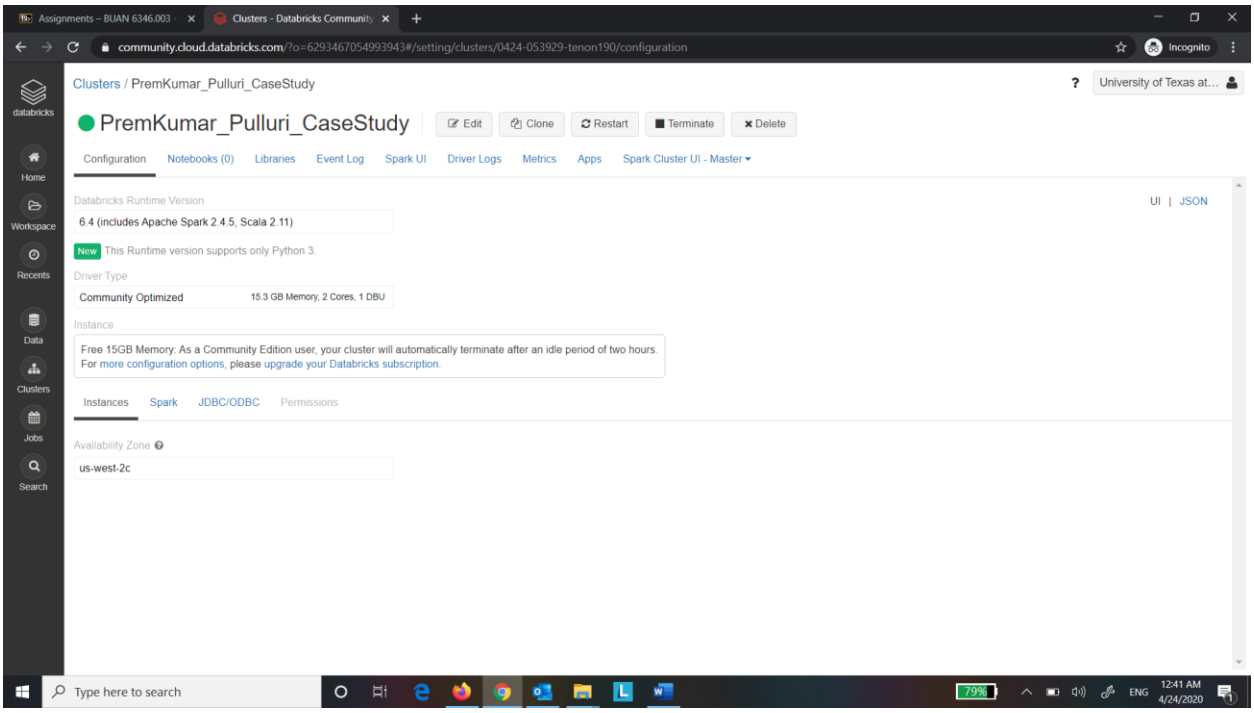
Part 2: Create a cluster node.

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.

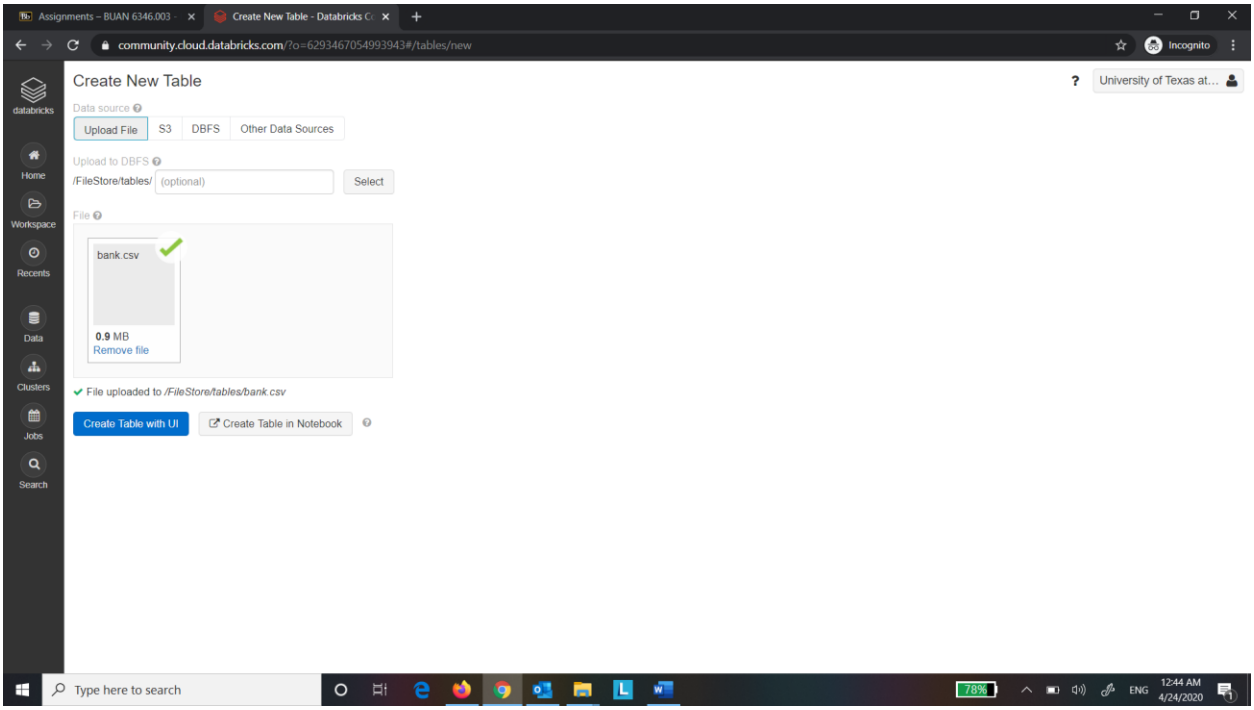
Q. Take a screenshot of the running Cluster Page and paste it below.



Part 3: Create Datatable.

Step 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.

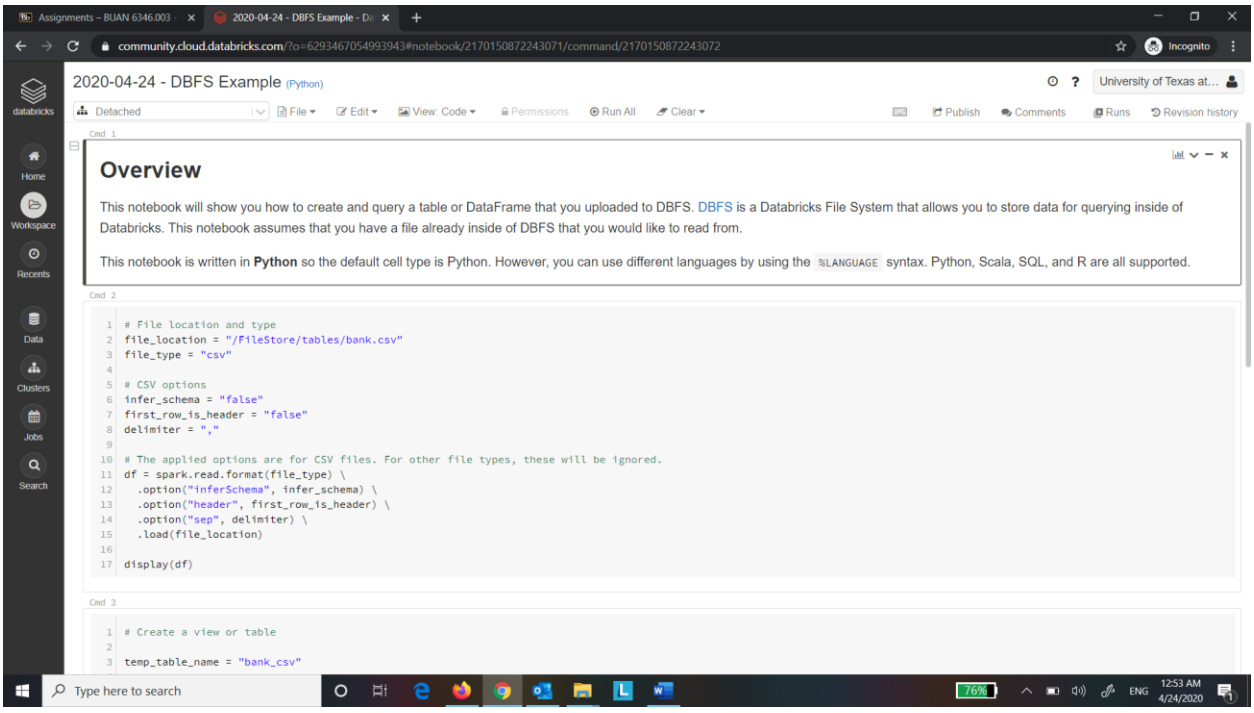


Step 9: The file location can now be seen.

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

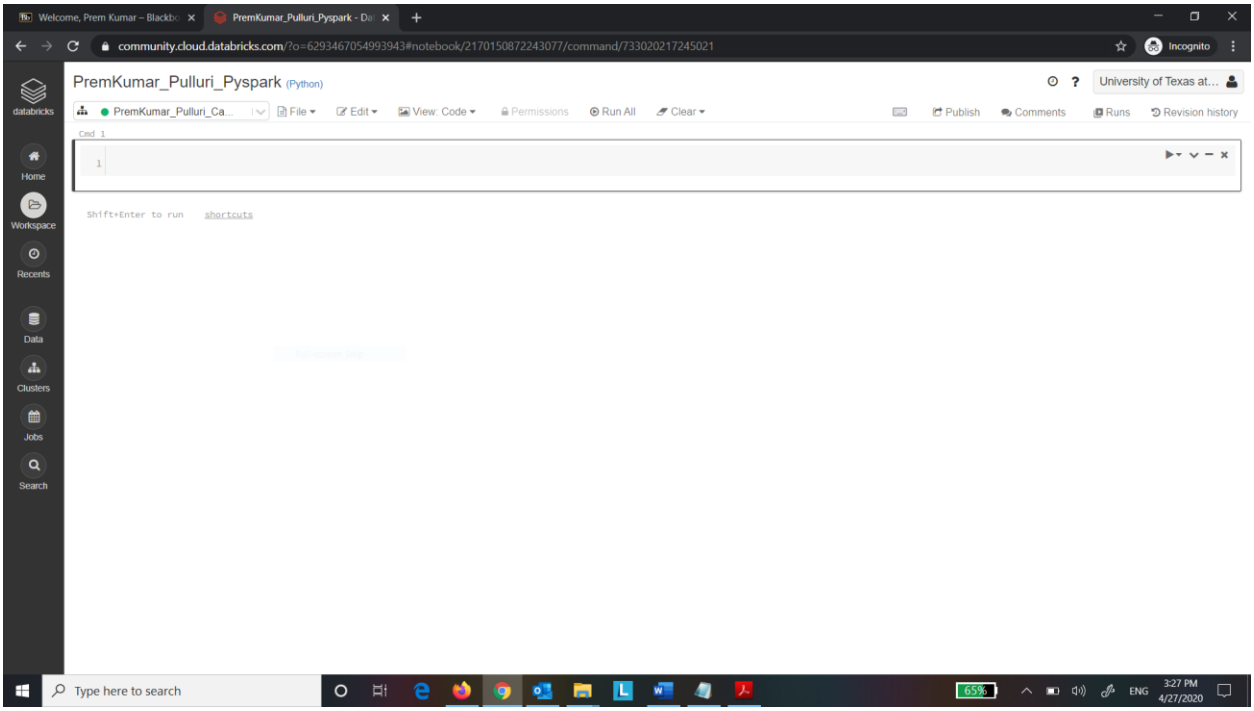
Copy the file location and paste it here.

```
# File location and type
file_location = "/FileStore/tables/bank.csv"
file_type = "csv"
```



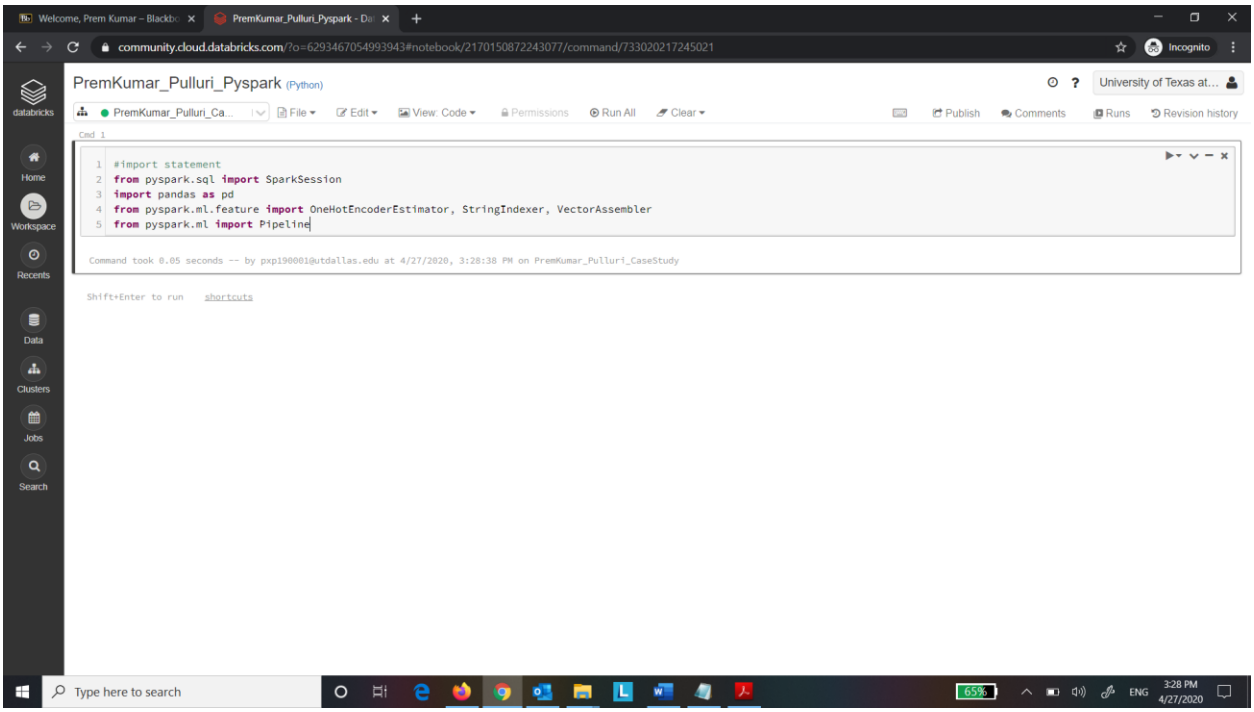
Part 4: DataFrame Operations.

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.

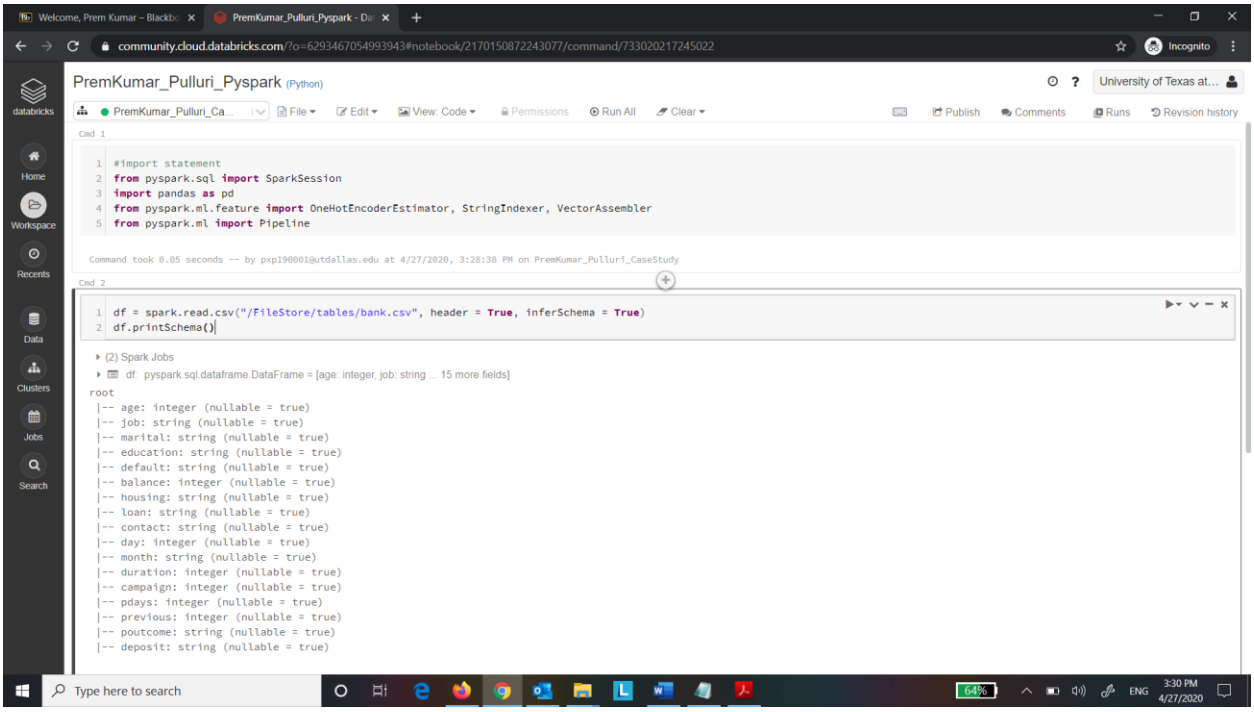


Step 12:

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

Here our output variable id is “deposit”. List all the input variables.

```
age: integer (nullable = true) job: string (nullable = true) marital: string
(nullable = true) education: string (nullable = true) default: string
(nullable = true) balance: integer (nullable = true) housing: string
(nullable = true) loan: string (nullable = true) contact: string (nullable =
true) day: integer (nullable = true) month: string (nullable = true)
duration: integer (nullable = true) campaign: integer (nullable = true)
pdays: integer (nullable = true) previous: integer (nullable = true)
poutcome: string (nullable = true)
```



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.

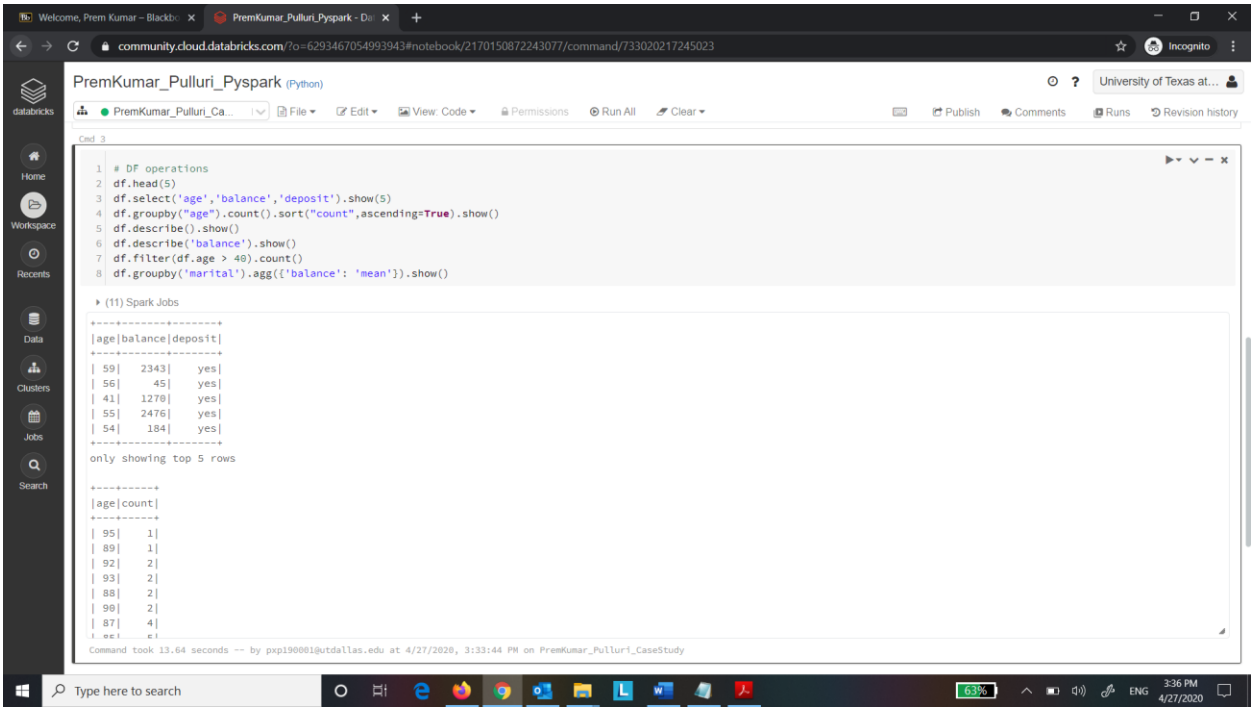
“Select” will select the mentioned columns and “Group by” will aggregate all the rows by the mentioned columns

ii. How many customers are there with an age greater than 40?

There are 4967 customers whose age is greater than 40

iii. Why are there only 5 records viewable?

Only 5 records are viewable because we have passed 5 as parameter in show function

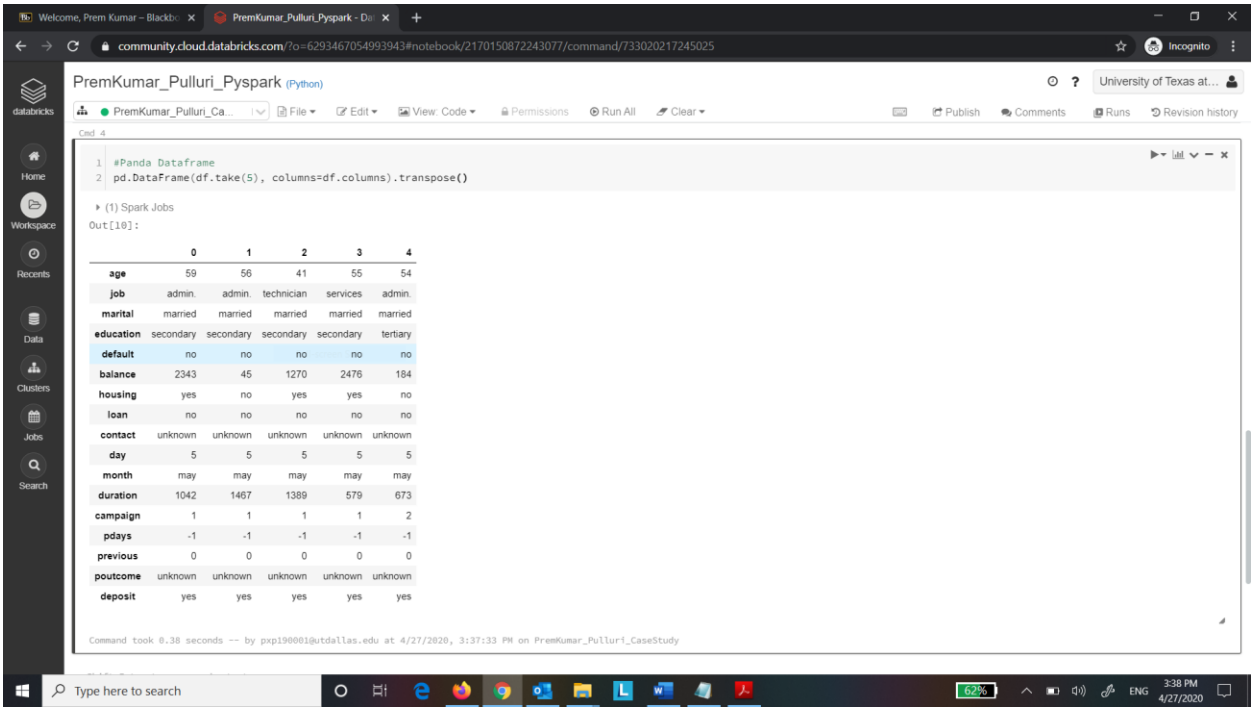


Step 14: Execute the following code:

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

What is seen in the output of the transpose?

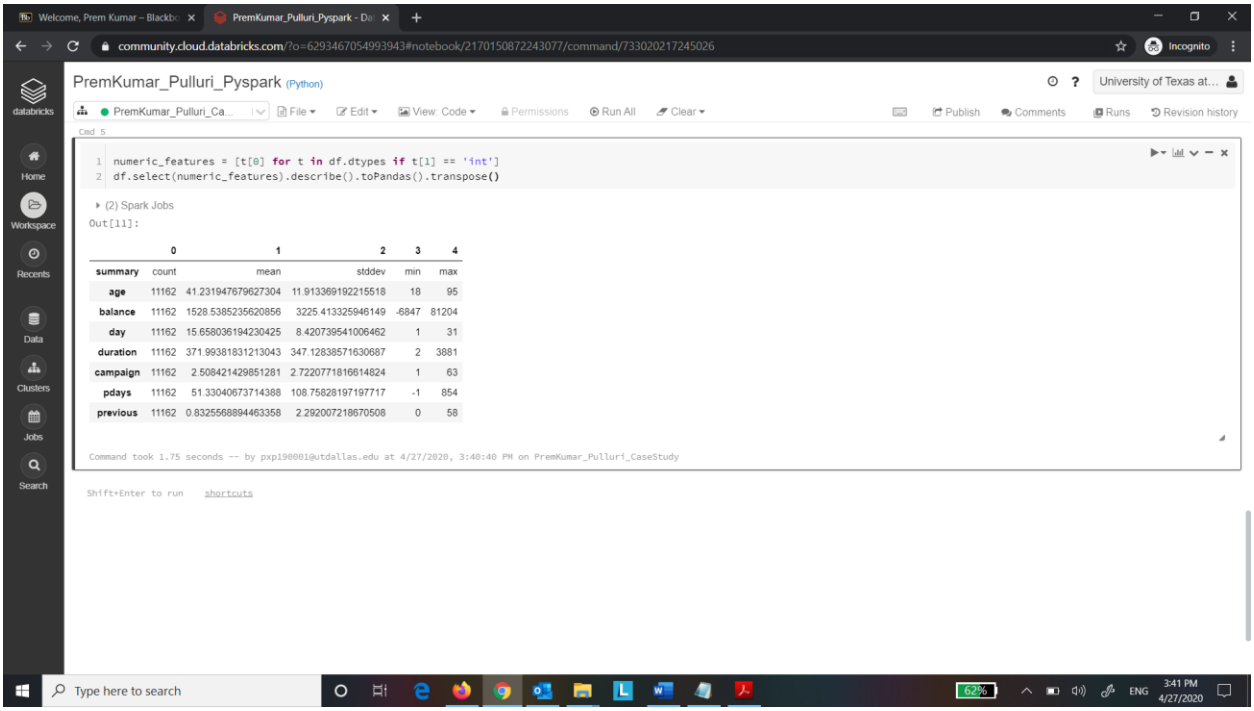
The rows and columns are interchanged.



	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

Step 15: Execute the following code:

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

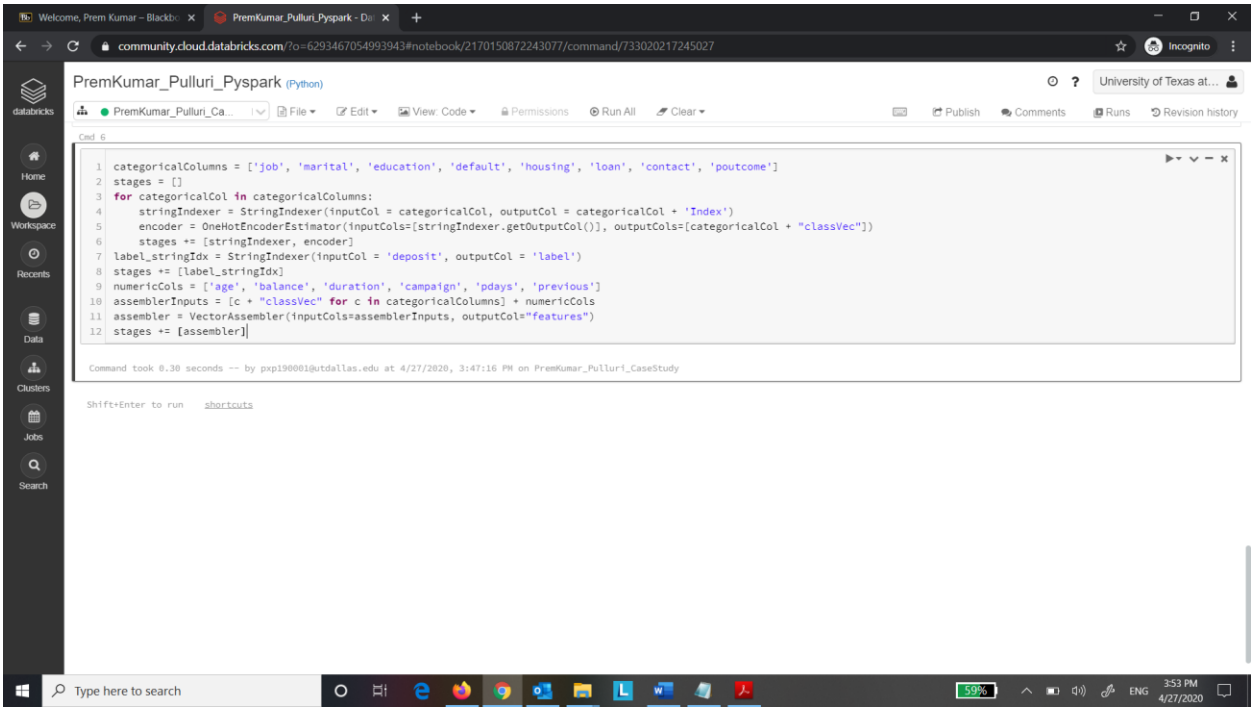


	0	1	2	3	4
summary	count	mean	stddev	min	max
age	11162	41.23194767962730	11.913369192215518	18	95
balance	11162	1528.5385235620856	3225.413325946149	-6847	81204
day	11162	15.658036194230425	8.420739541006462	1	31
duration	11162	371.99381831213043	347.12838571630687	2	3881
campaign	11162	2.508421429851281	2.7220771816614824	1	63
pdays	11162	51.33040673714388	108.75828197197717	-1	854
previous	11162	0.8325568894463358	2.292007218670508	0	58

Part 5: Data Preprocessing:

Step 16: Execute the following code:

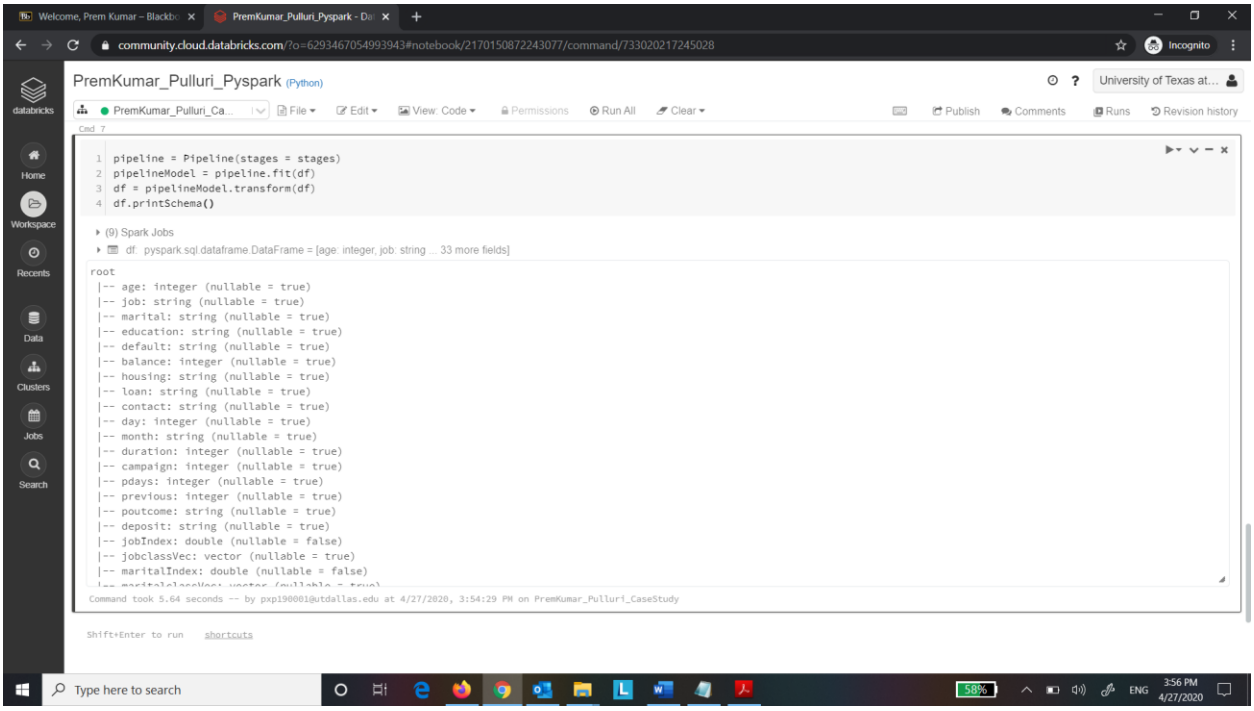
Q. Take a screenshot of the output of the code and paste it below.



Step 17: Execute the following code:

```
pipeline = Pipeline(stages = stages)
pipelineModel = pipeline.fit(df)
df = pipelineModel.transform(df)
df.printSchema()
```

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



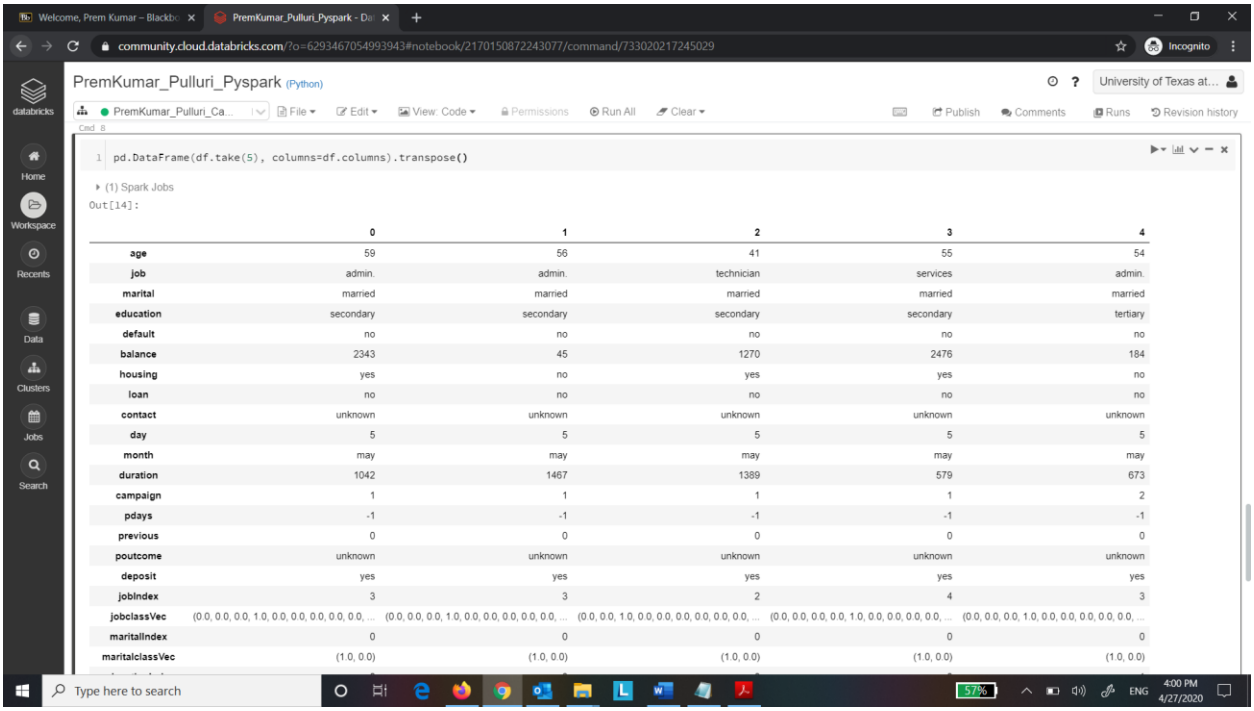
Step 18: Execute the following command:

```
pd.DataFrame(df.take(5), columns=df.columns).transpose()
```

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

Define the purpose of the transpose function?

The purpose of the transpose function is to swap the rows and columns

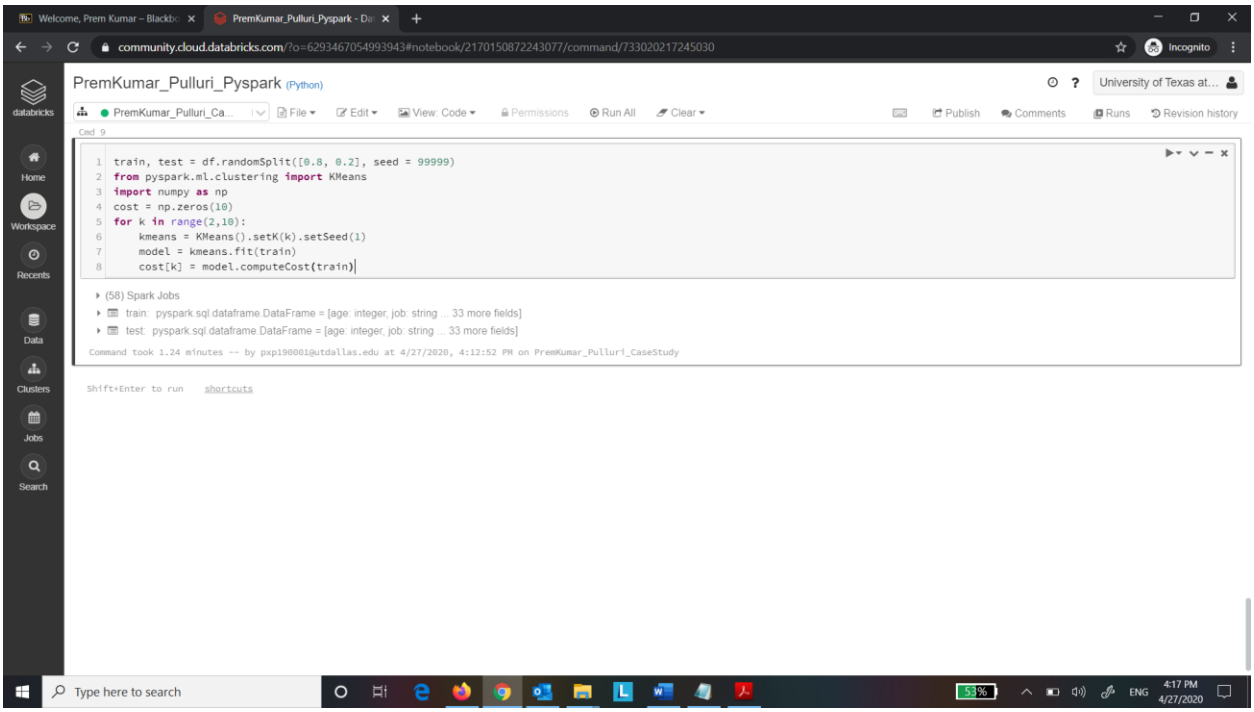


	0	1	2	3	4
age	59	56	41	55	54
job	admin.	admin.	technician	services	admin.
marital	married	married	married	married	married
education	secondary	secondary	secondary	secondary	tertiary
default	no	no	no	no	no
balance	2343	45	1270	2476	184
housing	yes	no	yes	yes	no
loan	no	no	no	no	no
contact	unknown	unknown	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
jobIndex	3	3	2	4	3
jobclassVec	(0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
maritalIndex	0	0	0	0	0
maritalclassVec	(1.0, 0.0)	(1.0, 0.0)	(1.0, 0.0)	(1.0, 0.0)	(1.0, 0.0)

Step 19: Execute the following code:

```
train, test = df.randomSplit([0.8, 0.2], seed = 99999)
from pyspark.ml.clustering import KMeans
import numpy as np
cost = np.zeros(10)
for k in range(2,10):
    kmeans = KMeans().setK(k).setSeed(1)
    model = kmeans.fit(train)
    cost[k] = model.computeCost(train)
```

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

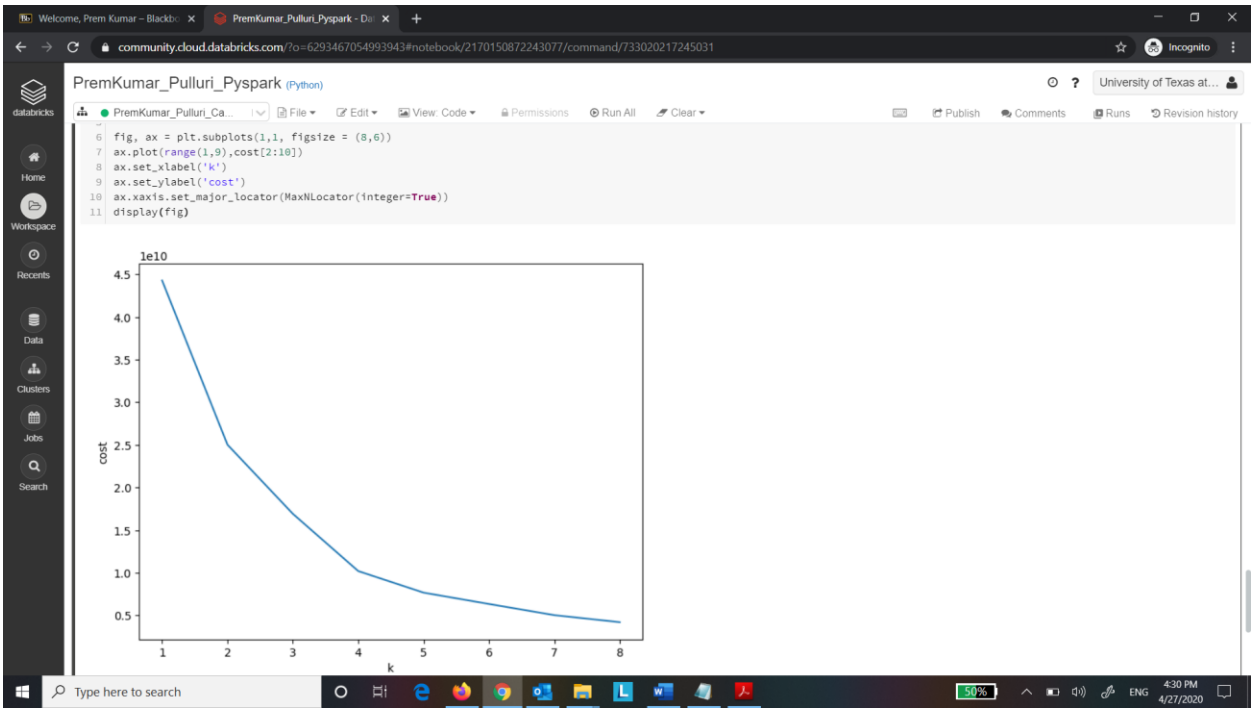


Step 20: Execute the following code:

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

Comment about the graph that is plotted here.

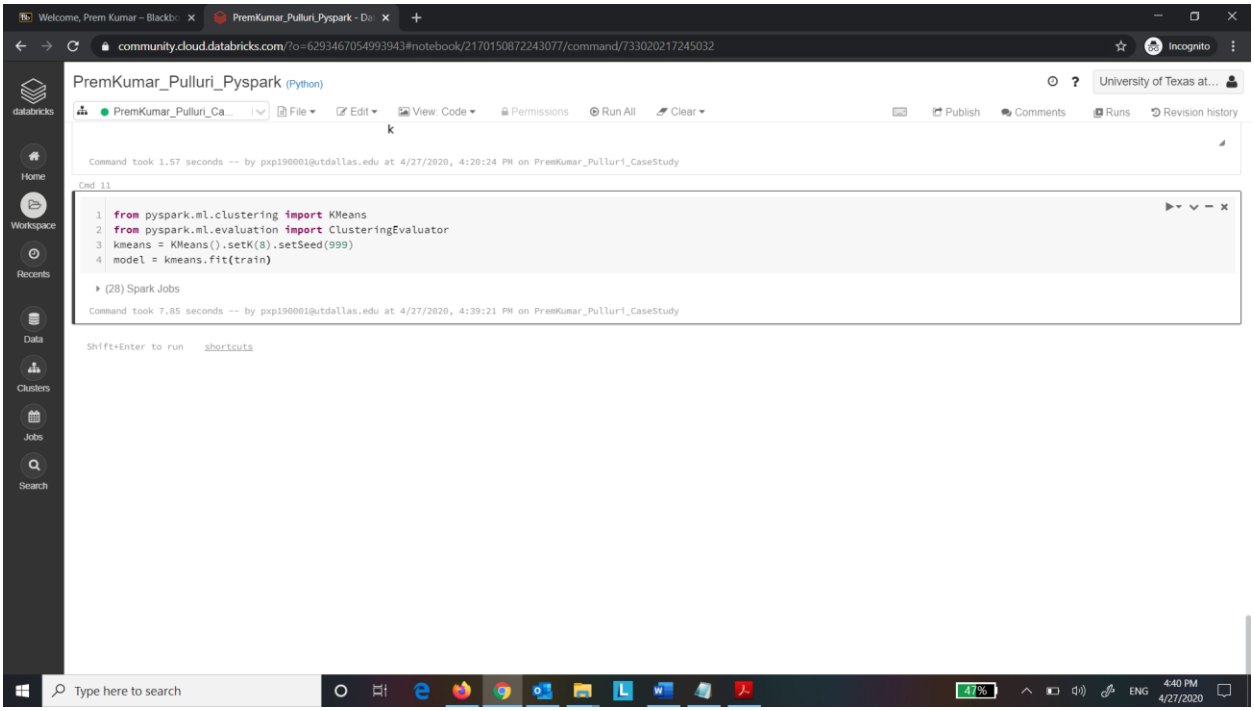
From the elbow plot, we can say that the cost decreases as number of clusters increases



Step 21: Execute the following code:

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
kmeans = KMeans().setK(8).setSeed(999)
model = kmeans.fit(train)
```

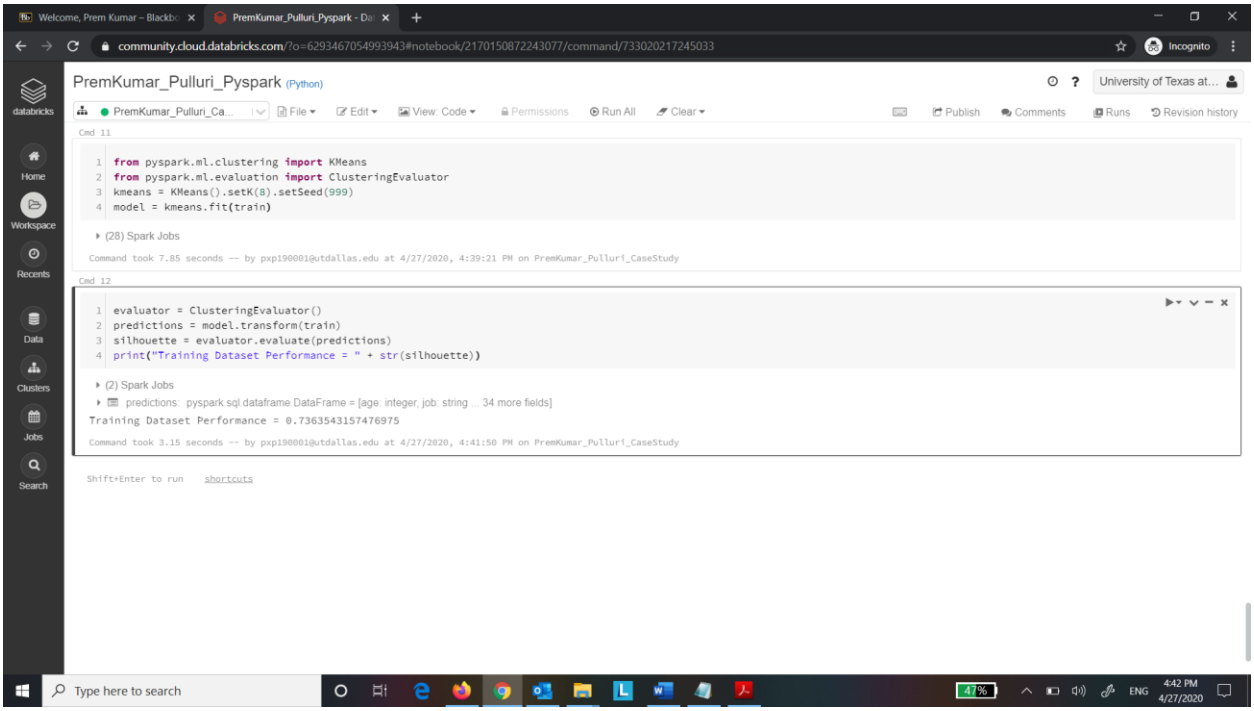
Q. Take a screenshot of the output of the code and paste it below.



Step 22: Execute the following code:

```
evaluator = ClusteringEvaluator()
predictions = model.transform(train)
silhouette = evaluator.evaluate(predictions)
print("Training Dataset Performance = " + str(silhouette))
```

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



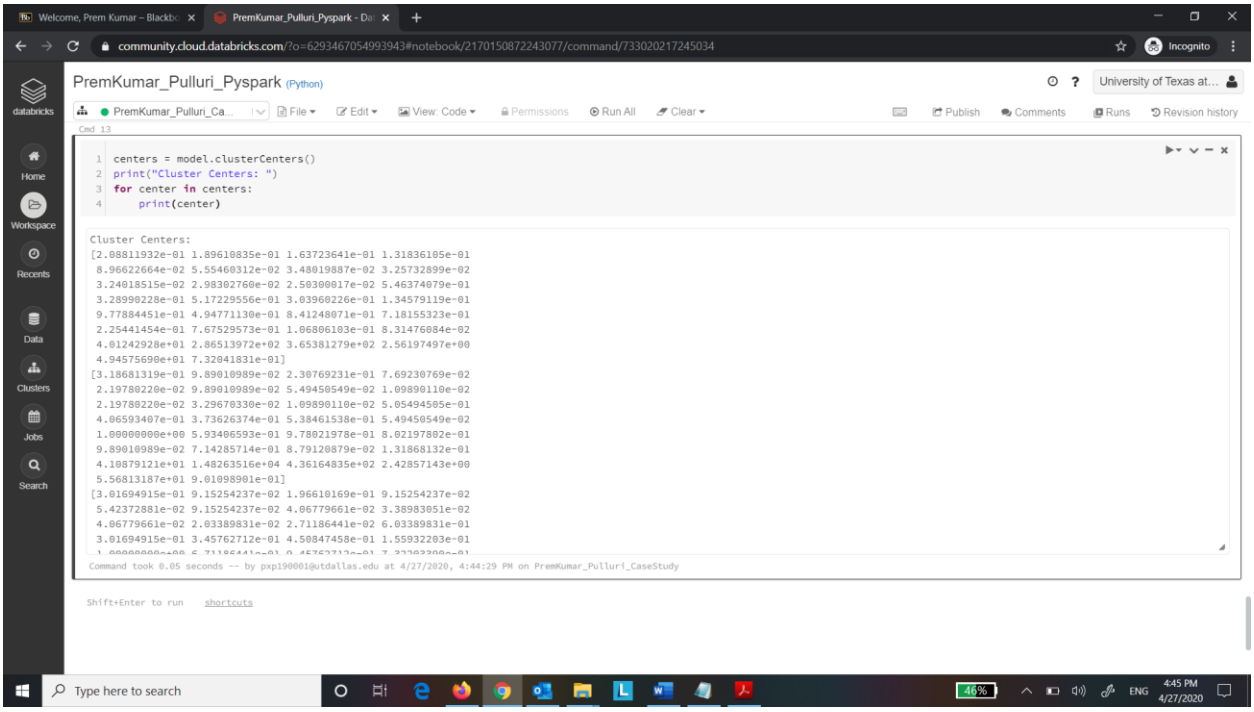
Step 23: Execute the following code:

```
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

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?

A centroid is a data point at the center of a cluster. Centroid is calculated by taking the mean of all the data points in a cluster

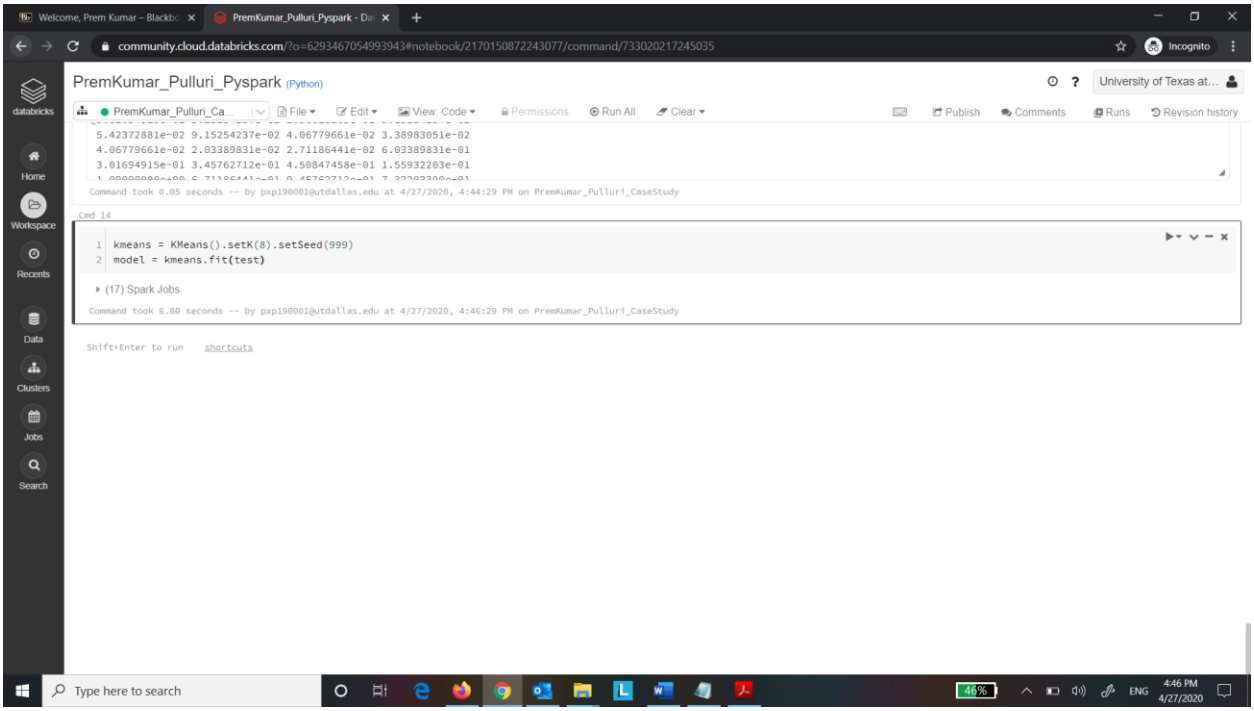


Step 24:

Execute the following code:

```
kmeans = KMeans().setK(8).setSeed(999)
model = kmeans.fit(test)
```

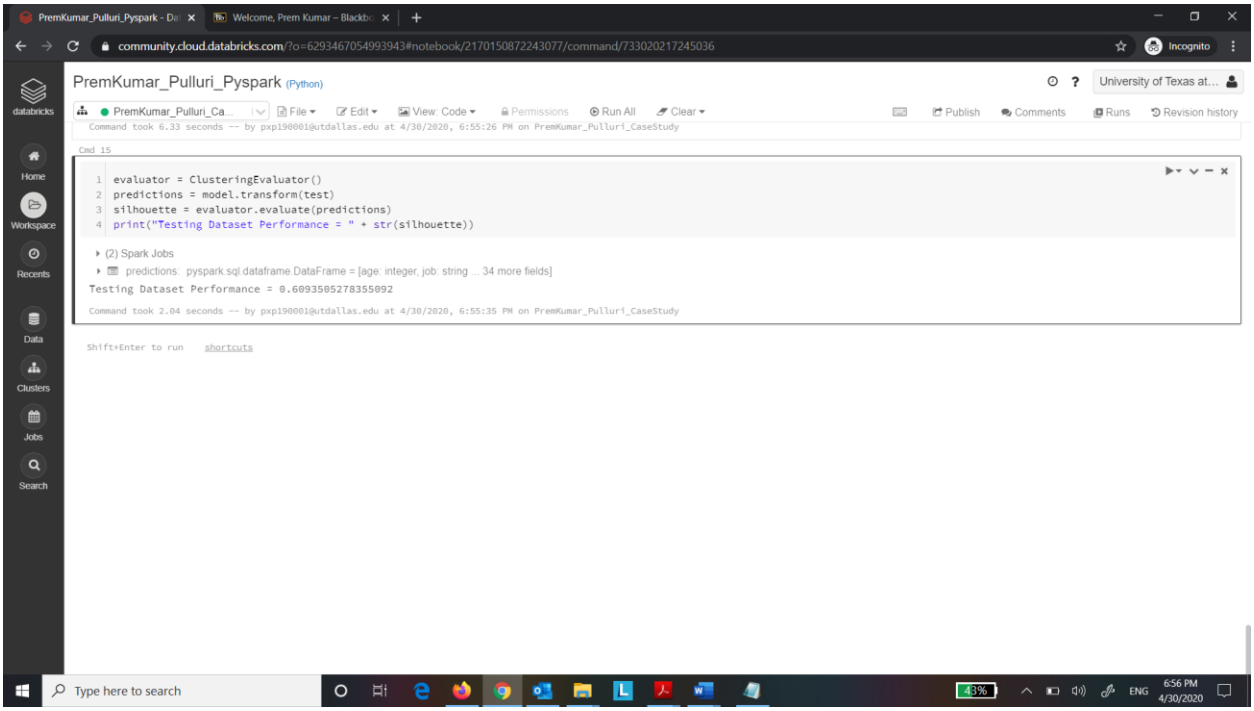
Q. Take a screenshot of the output of the code and paste it below.



Step 25: Execute the following code:

```
evaluator = ClusteringEvaluator()
predictions = model.transform(test)
silhouette = evaluator.evaluate(predictions)
print("Testing Dataset Performance = " + str(silhouette))
```

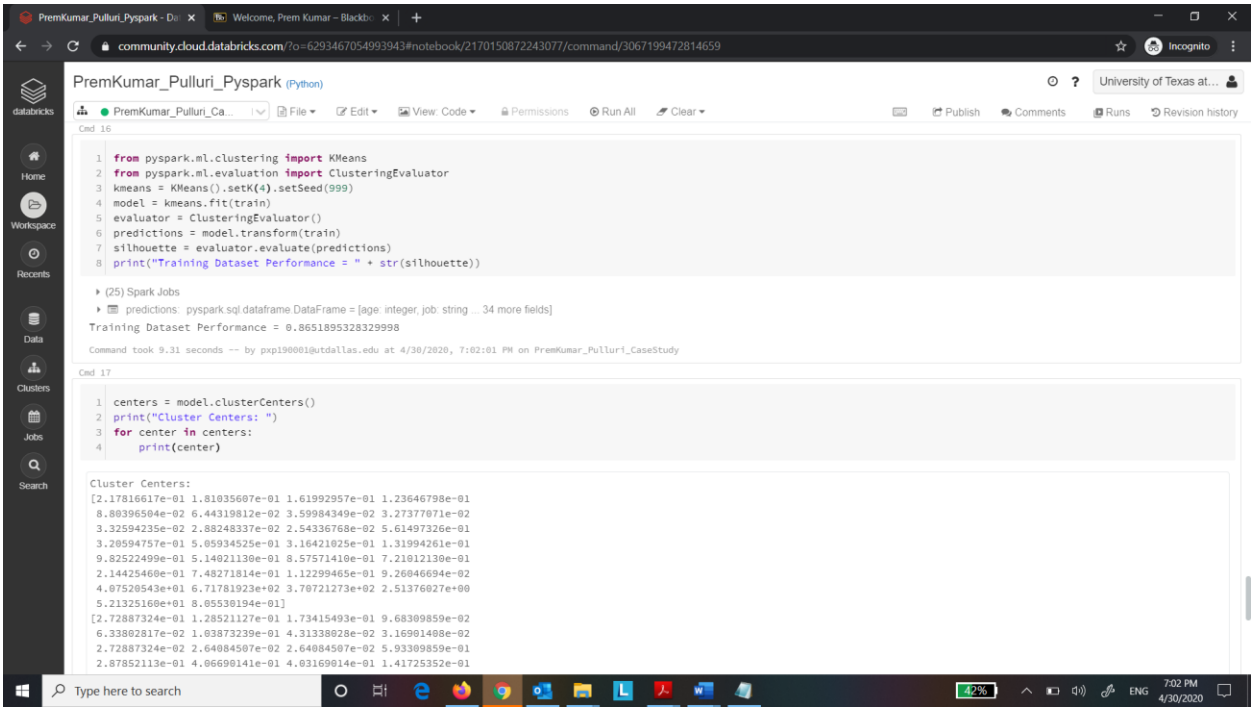
Q. Take a screenshot of the output of the code and paste it below.

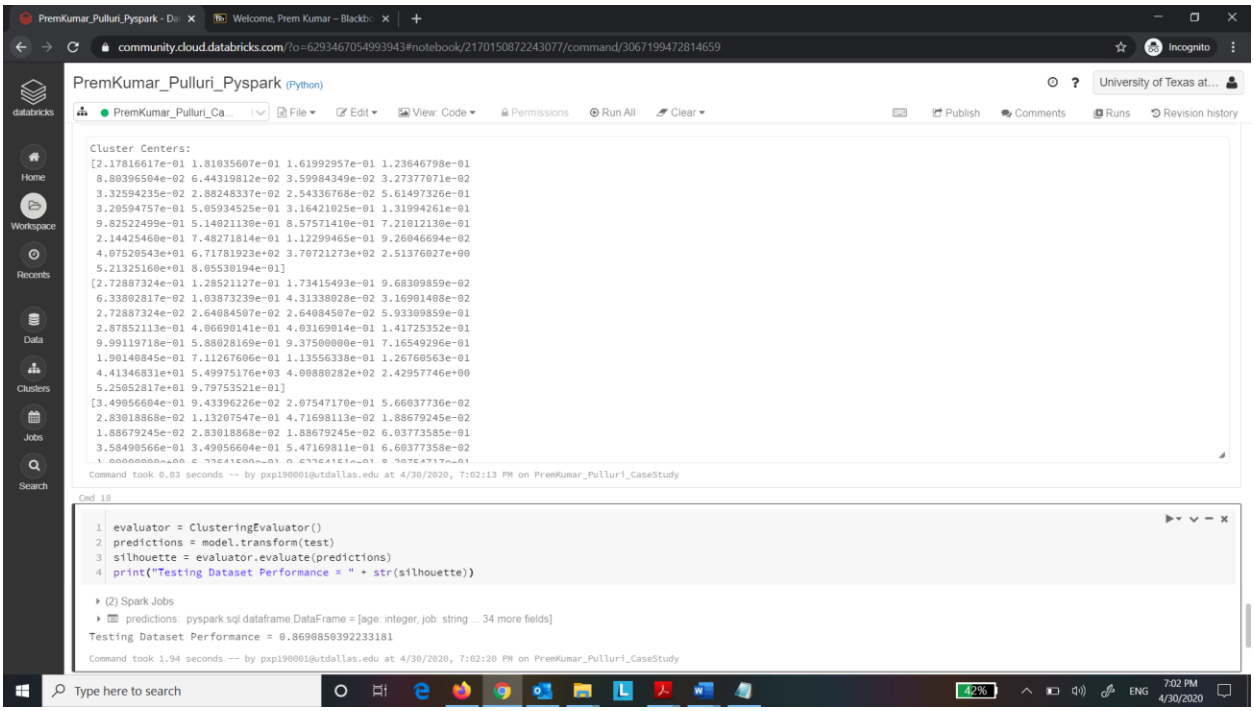


Copy the Train and Test performance results of all the clusters here. Compare the results and comment about the performance of each model. Select your best model and explain why you chose that number of clusters as the optimal level.

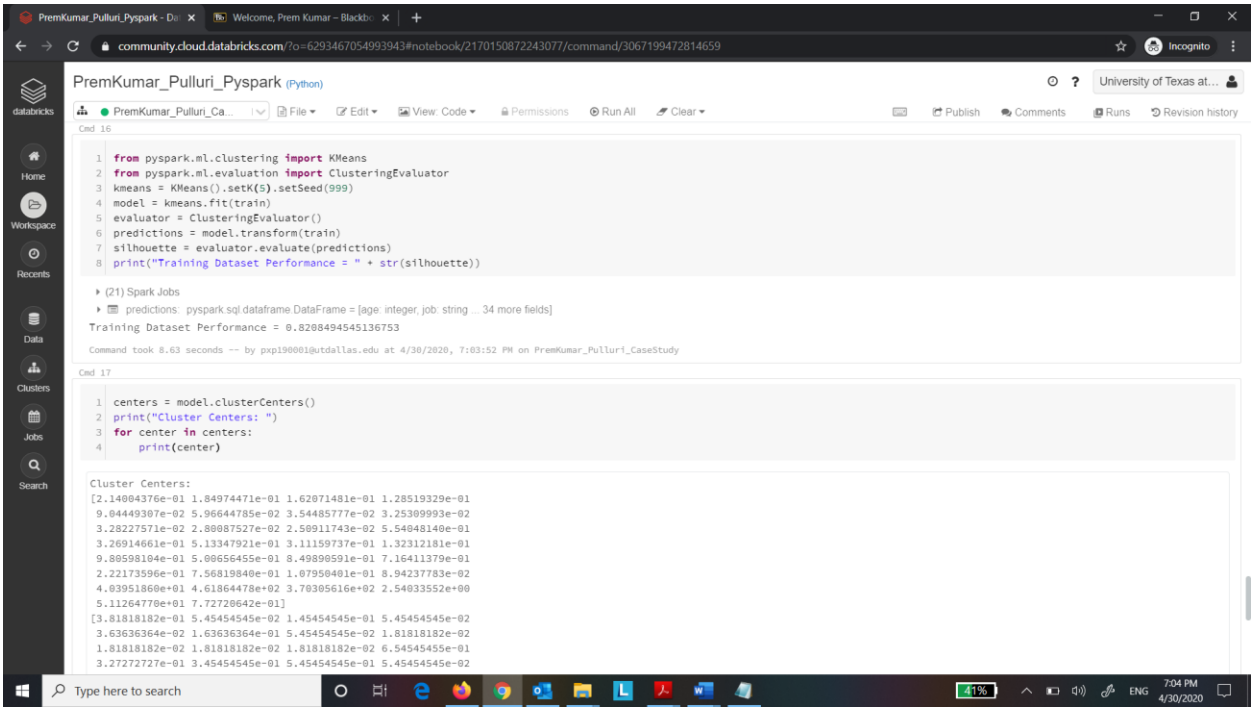
From the elbow plot, we can see that the cost function decreases significantly until K=4 but after that there isn't significant decrease. If K is less, it leads to overfitting and higher value of K leads to underfitting. So, 4 or 5 might be the optimal value for number of clusters K

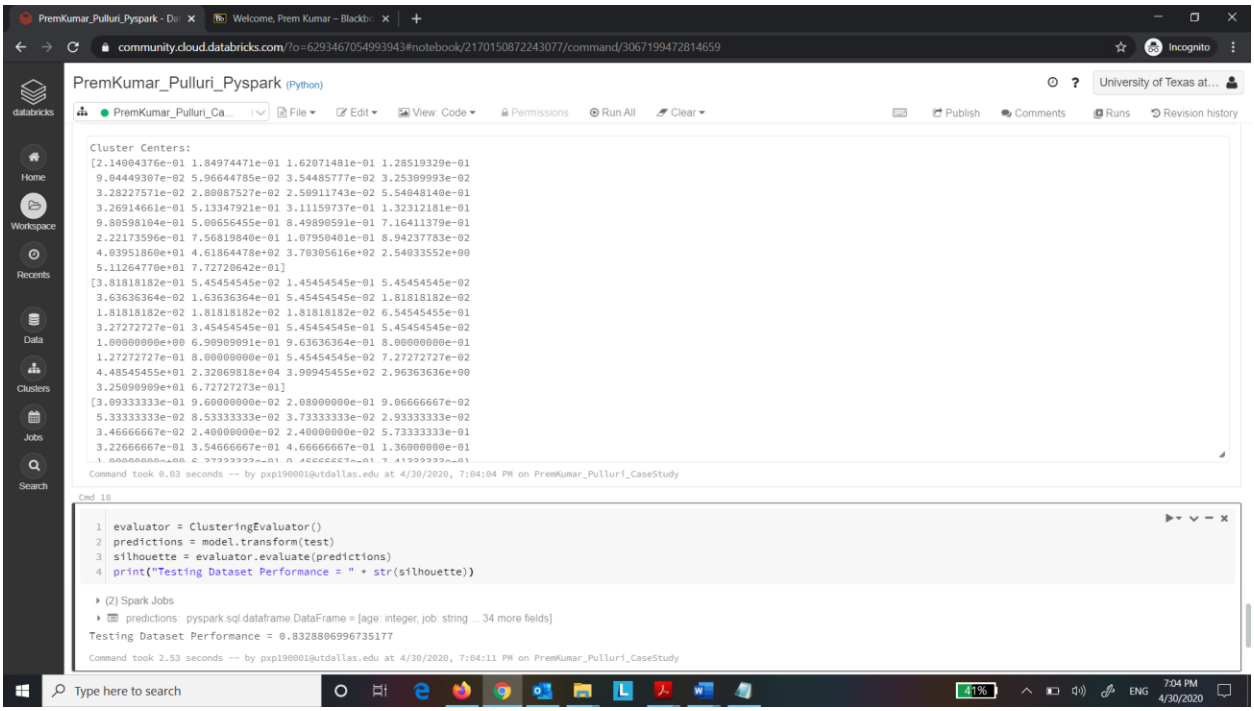
For K=4




PremKumar_Pulluri_Pyspark - Da
Welcome, Prem Kumar – Blackb
community.cloud.databricks.com/?o=6293467054993943#notebook/2170150872243077/command/3067199472814659
Incognito
PremKumar_Pulluri_Pyspark (Python)
University of Texas at...
databricks
Home
Workspace
Recents
Data
Clusters
Jobs
Search
PremKumar_Pulluri_Ca...
File
Edit
View: Code
Permissions
Run All
Clear
Publish
Comments
Runs
Revision history
Cluster Centers:
[2.17816617e-01 1.81035607e-01 1.61992957e-01 1.23646798e-01
8.80396504e-02 6.44319812e-02 3.59984349e-02 3.27377071e-02
3.32594235e-02 2.88248337e-02 2.54336768e-02 5.61497326e-01
3.20594757e-01 5.05934525e-01 3.16421025e-01 1.31994261e-01
9.82522499e-01 5.14021130e-01 8.57571410e-01 7.21012130e-01
2.14425460e-01 7.48271814e-01 1.12299465e-01 9.26046694e-02
4.07520543e+01 6.71781923e+02 3.70721273e+02 2.51376027e+00
5.21325160e+01 8.05530194e-01]
[2.72887324e-01 1.28521127e-01 1.73415493e-01 9.68309859e-02
6.33802817e-02 1.03873239e-01 4.31338028e-02 3.16901408e-02
2.72887324e-02 2.64084507e-02 2.64084507e-02 5.93309859e-01
2.87852113e-01 4.06690141e-01 4.03169014e-01 1.41725352e-01
9.99119718e-01 5.88028169e-01 9.37500000e-01 7.16549296e-01
1.90140845e-01 7.11267606e-01 1.13556338e-01 1.26760563e-01
4.41346831e+01 5.49975176e+03 4.00880282e+02 2.42957746e+00
5.25052817e+01 9.79753521e-01]
[3.49056604e-01 9.43396226e-02 2.07547170e-01 5.66037736e-02
2.83018868e-02 1.13207547e-01 4.71698113e-02 1.88679245e-02
1.88679245e-02 2.83018868e-02 1.88679245e-02 6.03773585e-01
3.58490566e-01 3.49056604e-01 5.47169811e-01 6.60377358e-02
Command took 0.03 seconds -- by pxp190001@utdallas.edu at 4/30/2020, 7:02:13 PM on PremKumar_Pulluri_CaseStudy
Cmd 18
evaluator = ClusteringEvaluator()
predictions = model.transform(test)
silhouette = evaluator.evaluate(predictions)
print("Testing Dataset Performance = " + str(silhouette))
(2) Spark Jobs
predictions: pyspark.sql.dataframe.DataFrame = [age: integer, job: string ... 34 more fields]
Testing Dataset Performance = 0.8690850392233181
Command took 1.94 seconds -- by pxp190001@utdallas.edu at 4/30/2020, 7:02:20 PM on PremKumar_Pulluri_CaseStudy
Type here to search
42%
ENG
7:02 PM
4/30/2020

For K=5


PremKumar_Pulluri_Pyspark - Da
Welcome, Prem Kumar – Blackb
community.cloud.databricks.com/?o=6293467054993943#notebook/2170150872243077/command/3067199472814659
Incognito
PremKumar_Pulluri_Pyspark (Python)
University of Texas at...
databricks
Home
Workspace
Recents
Data
Clusters
Jobs
Search
PremKumar_Pulluri_Ca...
File
Edit
View: Code
Permissions
Run All
Clear
Publish
Comments
Runs
Revision history
Cmd 16
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
kmeans = KMeans().setK(5).setSeed(999)
model = kmeans.fit(train)
evaluator = ClusteringEvaluator()
predictions = model.transform(train)
silhouette = evaluator.evaluate(predictions)
print("Training Dataset Performance = " + str(silhouette))
(21) Spark Jobs
predictions: pyspark.sql.dataframe.DataFrame = [age: integer, job: string ... 34 more fields]
Training Dataset Performance = 0.8208494545136753
Command took 8.63 seconds -- by pxp190001@utdallas.edu at 4/30/2020, 7:03:52 PM on PremKumar_Pulluri_CaseStudy
Cmd 17
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
print(center)
Cluster Centers:
[2.14004376e-01 1.84974471e-01 1.62071481e-01 1.28519329e-01
9.04449307e-02 5.96644785e-02 3.54485777e-02 3.25309993e-02
3.28227571e-02 2.80087527e-02 2.50911743e-02 5.54048140e-01
3.26914661e-01 5.13347921e-01 3.11159737e-01 1.32312181e-01
9.80598104e-01 5.00656455e-01 8.49890591e-01 7.16411379e-01
2.22173596e-01 7.56819840e-01 1.07950401e-01 8.94237783e-02
4.03951860e+01 4.61864478e+02 3.70305616e+02 2.54033552e+00
5.11264770e+01 7.72720642e-01]
[3.81818182e-01 5.45454545e-02 1.45454545e-01 5.45454545e-02
3.63636364e-02 1.63636364e-01 5.45454545e-02 1.81818182e-02
1.81818182e-02 1.81818182e-02 1.81818182e-02 6.54545455e-01
3.27272727e-01 3.45454545e-01 5.45454545e-01 5.45454545e-02
Type here to search
41%
ENG
7:04 PM
4/30/2020



The train and test dataset performance is better for K = 4 compared to K = 5. Hence K = 4 is the best choice for the value of K