Name: Akshat Tiwari	Class/Roll No: D16AD / 62	Grade:

Program:

```
[cloudera@quickstart ~]$ pyspark
Python 2.6.6 (r266:84292, Jul 23 2015, 15:22:56)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/parquet/lib/slf4j-log4j12-1.7.5.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/avro/avro-tools-1.7.6-cdh5.12.0.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/10/09 13:42:21 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
23/10/09 13:42:22 WARN util.Utils: Your hostname, quickstart.cloudera resolves t
o a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth0)
23/10/09 13:42:22 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to ano
ther address
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Python version 2.6.6 (r266:84292, Jul 23 2015 15:22:56)
SparkContext available as sc, HiveContext available as sqlContext.
>>> df = sqlContext.createDataFrame([[0, 33.3, -17.5], [1, 40.4, -20.5], [2, 28.
6, -23.9], [3, 29.5, -19.0], [4, 32.8, -18.84]], ["other", "lat", "long"])
23/10/09 13:45:29 WARN shortcircuit.DomainSocketFactory: The short-circuit local
 reads feature cannot be used because libhadoop cannot be loaded.
>>> df.show()
+-----+----+------+
|other| lat|  long|
+-----+----+------+
|    0|33.3| -17.5|
|    1|40.4| -20.5|
|    2|28.6| -23.9|
|    3|29.5| -19.0|
|    4|32.8|-18.84|
+-----+----+------+
```

```
>>> from pyspark.ml.feature import VectorAssembler
>>> vecAssembler = VectorAssembler(inputCols = ["lat", "long"], outputCol = "features")
>>> new_df = vecAssembler.transform(df)
>>> new_df.show()
+-----+----+------+-------------+
|other| lat|  long|     features|
+-----+----+------+-------------+
|    0|33.3| -17.5| [33.3,-17.5]|
|    1|40.4| -20.5| [40.4,-20.5]|
|    2|28.6| -23.9| [28.6,-23.9]|
|    3|29.5| -19.0| [29.5,-19.0]|
|    4|32.8|-18.84|[32.8,-18.84]|
+-----+----+------+-------------+

>>> from pyspark.ml.clustering import KMeans
>>> kmeans = KMeans(k=2, seed=1)
>>> model = kmeans.fit(new_df.select('features'))
```


```
>>> from pyspark.ml.clustering import KMeans
>>> kmeans = KMeans(k=2, seed=1)
>>> model = kmeans.fit(new_df.select('features'))
```


```
>>> transformed = model.transform(new_df)
>>> transformed.show()
+-----+----+------+-------------+----------+
|other| lat|  long|     features|prediction|
+-----+----+------+-------------+----------+
|    0|33.3| -17.5| [33.3,-17.5]|         0|
|    1|40.4| -20.5| [40.4,-20.5]|         1|
|    2|28.6| -23.9| [28.6,-23.9]|         0|
|    3|29.5| -19.0| [29.5,-19.0]|         0|
|    4|32.8|-18.84|[32.8,-18.84]|         0|
+-----+----+------+-------------+----------+
```