databricksK-means

```
from pyspark.sql import functions as fun
from matplotlib import pyplot as plt
```

```
path = "/FileStore/tables/"
fmall = path + "Mall_Customers.csv"
mall = spark.read.csv( fmall, inferSchema=True, header=True )
mall.describe().show()
```

```
+-------+------------------+------+-----------------+------------------+-----------
-----------+
|summary|        CustomerID| Genre|              Age|Annual Income (k$)|Spending Sc
ore (1-100)|
+-------+------------------+------+-----------------+------------------+-----------
-----------+
|  count|               200|   200|              200|               200|
200|
|   mean|             100.5|  null|            38.85|             60.56|
50.2|
| stddev|57.879184513951124|  null|13.96900733155888| 26.26472116527124|    25.8235
21668370173|
|    min|                 1|Female|               18|                15|
1|
|    max|               200|  Male|               70|               137|
99|
+-------+------------------+------+-----------------+------------------+-----------
-----------+
```

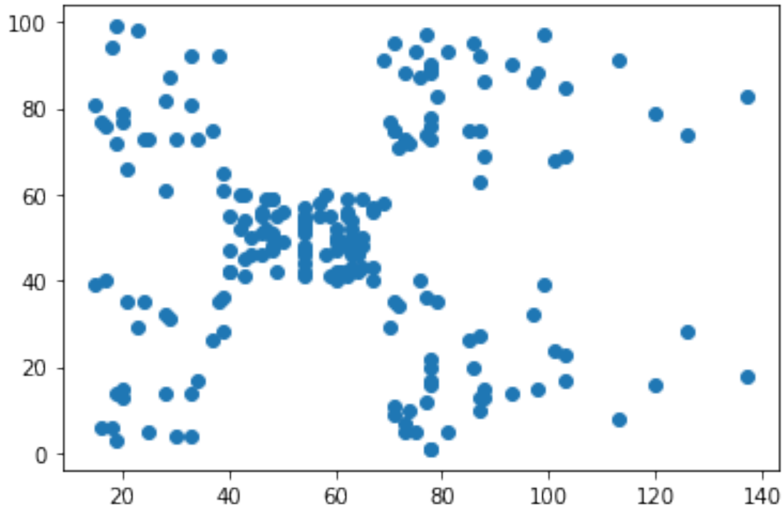
```
mall = mall.select( fun.col("Genre").alias("gender"),
                    fun.col("Age").alias("age"), fun.col("Annual Income
(k$)").alias("income"),
                    fun.col("Spending Score (1-100)").alias("score") )
mall.printSchema()
```

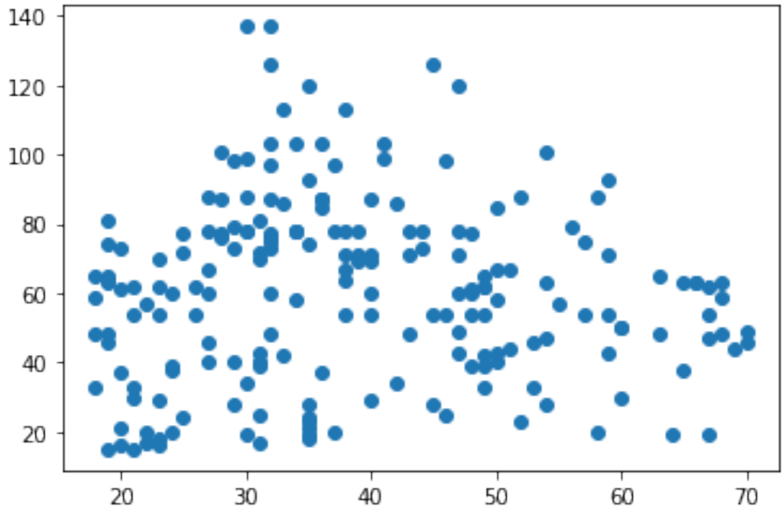
```
root
 |-- gender: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- income: integer (nullable = true)
 |-- score: integer (nullable = true)
```

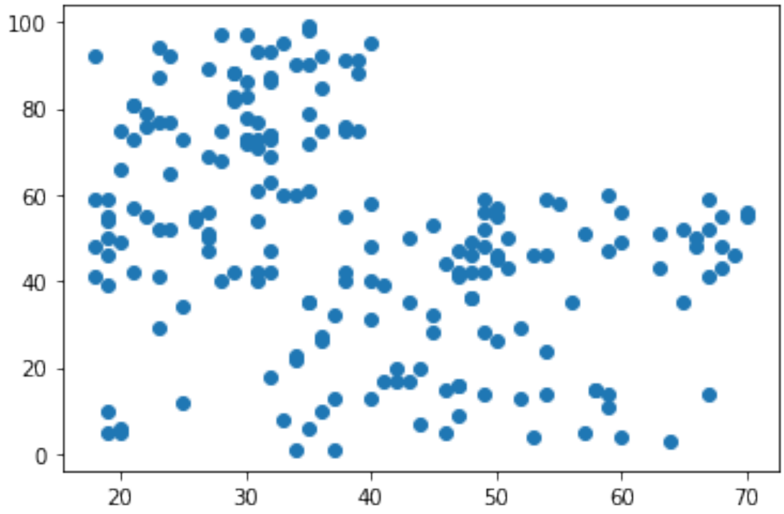
```
cols = mall.drop( "gender" ).columns
lisMean = [ fun.mean( c ) for c in cols ]
lisStd = [ fun.stddev( c ) for c in cols ]

mall.groupby("gender").agg( *lisMean, *lisStd ).show()
```

```
+------+------------------+------------------+------------------+------------------+
-------------------+------------------+
|gender|          avg(age)|       avg(income)|        avg(score)|  stddev_samp(age)|
stddev_samp(income)|stddev_samp(score)|
+------+------------------+------------------+------------------+------------------+
-------------------+------------------+
|Female|38.098214285714285|             59.25|51.526785714285715|12.644095457392353|
26.011951515055948|24.114949877478647|
|  Male| 39.80681818181818| 62.22727272727273| 48.51136363636363|15.514811576858186|
26.638373182494135|27.896769605833605|
+------+------------------+------------------+------------------+------------------+
-------------------+------------------+
```

```
# Let's explore the feature space
malldas = mall.toPandas()
plt.scatter( malldas["income"], malldas["score"] )
plt.show()
plt.scatter( malldas["age"], malldas["income"] )
plt.show()
plt.scatter( malldas["age"], malldas["score"] )
plt.show()
```







```
mall2 = mall.drop( "gender" )
```

```
from pyspark.ml.feature import VectorAssembler
vecassem = VectorAssembler( inputCols=[ "age", "income", "score" ],
outputCol="features" )
mall3 = vecassem.transform( mall2 )
mall3.show(4)
```

```
+---+------+-----+----------------+
|age|income|score|        features|
+---+------+-----+----------------+
| 19|    15|   39|[19.0,15.0,39.0]|
| 21|    15|   81|[21.0,15.0,81.0]|
| 20|    16|    6| [20.0,16.0,6.0]|
| 23|    16|   77|[23.0,16.0,77.0]|
+---+------+-----+----------------+
only showing top 4 rows
```

```
from pyspark.ml.clustering import KMeans
k1 = KMeans( featuresCol="features", predictionCol="cluster", k=5 )
k1Fit = k1.fit( mall3 )
print ( "Centroids coordinates : ",  k1Fit.clusterCenters() )
print( " Number of data points in each cluster:", k1Fit.summary.clusterSizes )
```

```
Centroids coordinates :  [array([45.2173913 , 26.30434783, 20.91304348]), array([43
.08860759, 55.29113924, 49.56962025]), array([32.69230769, 86.53846154, 82.12820513
]), array([40.66666667, 87.75      , 17.58333333]), array([25.52173913, 26.30434783
, 78.56521739])]
 Number of data points in each cluster: [23, 79, 39, 36, 23]
```

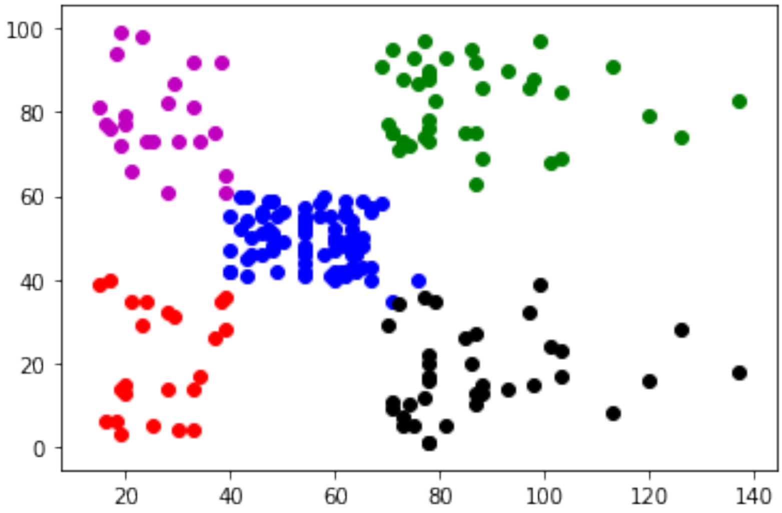
```
mallK = k1Fit.transform( mall3 )
mallK.show()
```

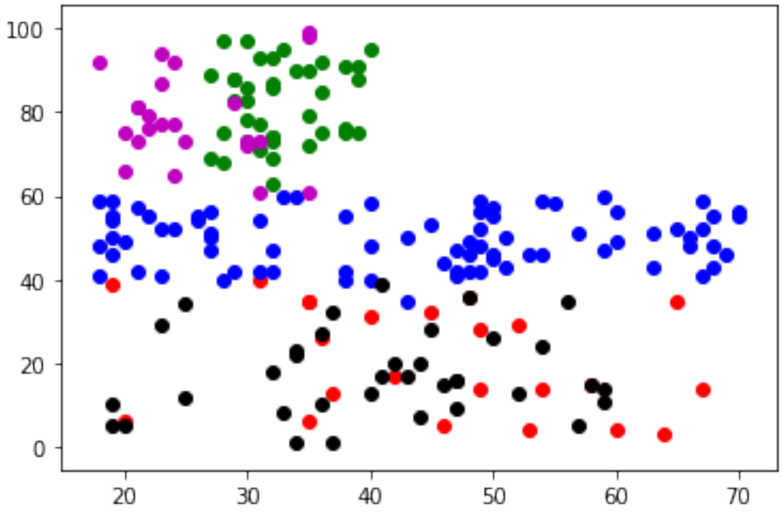
```
+---+------+-----+----------------+-------+
|age|income|score|        features|cluster|
+---+------+-----+----------------+-------+
| 19|    15|   39|[19.0,15.0,39.0]|      0|
| 21|    15|   81|[21.0,15.0,81.0]|      4|
| 20|    16|    6| [20.0,16.0,6.0]|      0|
| 23|    16|   77|[23.0,16.0,77.0]|      4|
| 31|    17|   40|[31.0,17.0,40.0]|      0|
| 22|    17|   76|[22.0,17.0,76.0]|      4|
| 35|    18|    6| [35.0,18.0,6.0]|      0|
| 23|    18|   94|[23.0,18.0,94.0]|      4|
| 64|    19|    3| [64.0,19.0,3.0]|      0|
| 30|    19|   72|[30.0,19.0,72.0]|      4|
| 67|    19|   14|[67.0,19.0,14.0]|      0|
| 35|    19|   99|[35.0,19.0,99.0]|      4|
| 58|    20|   15|[58.0,20.0,15.0]|      0|
| 24|    20|   77|[24.0,20.0,77.0]|      4|
| 37|    20|   13|[37.0,20.0,13.0]|      0|
| 22|    20|   79|[22.0,20.0,79.0]|      4|
| 35|    21|   35|[35.0,21.0,35.0]|      0|
| 20|    21|   66|[20.0,21.0,66.0]|      4|
| 52|    23|   29|[52.0,23.0,29.0]|      0|
| 35|    23|   98|[35.0,23.0,98.0]|      4|
+---+------+-----+----------------+-------+
only showing top 20 rows
```

```
from pyspark.ml.evaluation import ClusteringEvaluator
eval1 = ClusteringEvaluator( predictionCol='cluster', featuresCol='features' )
eval1.evaluate( mallK )
```

```
Out[46]: 0.6316639508003641
```

```
mallKandas = mallK.toPandas()
colours = ["r", "b", "g", "k", "m"]
for i in range( len(k1Fit.clusterCenters()) ) :
  plt.scatter( mallKandas[mallKandas["cluster"]==i]["income"],
mallKandas[mallKandas["cluster"]==i]["score"], c=colours[i] )
```


0
20
40
60
80
100
20
40
60
80
100
120
140


0
20
40
60
80
100
20
30
40
50
60
70