Untitled117

July 5, 2023

1 Module 9: K-Means Case Study

Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: Consider yourself to be Sam who is a data scientist. He has been approached by a retail car showroom to help them segregate the cars into different clusters. Tasks To Be Performed: 1. Building the K-Means clustering algorithm: a. Start off by extracting the 'mpg', 'disp' & 'hp' columns from the 'mtcars' data.frame. Store the result in 'car_features' b. Build the K-Means algorithm on top of 'car_features'. Here, the number of clusters should be 3 c. Bind the clustering vector to 'car_features' d. Extract observations belonging to individual clusters 2. On the same 'car_features'dataset build a K-Means algorithm, where the number of clusters is 5 a. Bind the clustering vector to 'car_features' b. Extract observations belonging to individual clusters

[1]:
```
 ## import the requirred libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
data=pd.read_csv('C:/Users/Vikas/Downloads/cars-1.csv')
```

[4]:
```
data.head()
```

[4]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

[5]:
```
data.columns
```

[5]:
```
Index(['model', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am',
       'gear', 'carb'],
      dtype='object')
```

[6]:
```
data.shape
```

[6]:
```
(32, 12)
```

[7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   model   32 non-null     object 
 1   mpg     32 non-null     float64
 2   cyl     32 non-null     int64  
 3   disp    32 non-null     float64
 4   hp      32 non-null     int64  
 5   drat    32 non-null     float64
 6   wt      32 non-null     float64
 7   qsec    32 non-null     float64
 8   vs      32 non-null     int64  
 9   am      32 non-null     int64  
 10  gear    32 non-null     int64  
 11  carb    32 non-null     int64  
dtypes: float64(5), int64(6), object(1)
memory usage: 3.1+ KB
```

[8]:
```
data.isnull().sum()
```

```
model    0
mpg      0
cyl      0
disp     0
hp       0
drat     0
wt       0
qsec     0
vs       0
am       0
gear     0
carb     0
dtype: int64
```

[10]:
```
data.columns
```

[10]:
```
Index(['model', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am',
       'gear', 'carb'],
      dtype='object')
```

[36]:
```
# a. Extract 'mpg', 'disp', and 'hp' columns from 'mtcars'
mtcars_fetures=data[['mpg','disp','hp']]
mtcars_fetures.head()
```

[36]:

	mpg	disp	hp
0	21.0	160.0	110
1	21.0	160.0	110
2	22.8	108.0	93
3	21.4	258.0	110
4	18.7	360.0	175

[23]:
```
from sklearn.cluster import KMeans

kmeans_model_3 = KMeans(n_clusters=3)
kmeans_model_3.fit(mtcars_fetures)
```

```
C:\Users\Vikas\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Vikas\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[23]:
```
KMeans(n_clusters=3)
```

[31]:
```
labels = kmeans.predict(mtcars_fetures)
labels
```

[31]:
```
array([1, 1, 1, 0, 2, 0, 2, 1, 1, 1, 1, 0, 0, 0, 2, 2, 2, 1, 1, 1, 1, 0,
       0, 2, 2, 1, 1, 1, 2, 1, 2, 1])
```

[42]:
```
centroids=kmeans.cluster_centers_
centroids
```

[42]:
```
array([[ 17.01428571, 276.05714286, 150.71428571],
       [ 24.5       , 122.29375   ,  96.875     ],
       [ 14.64444444, 388.22222222, 232.11111111]])
```

[46]:
```
kmeans_model_5 = kmeans(mtcars_features, centers = 5)
kmeans_model_5
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 kmeans_model_5 = kmeans(mtcars_features, centers = 5)
      2 kmeans_model_5

NameError: name 'mtcars_features' is not defined
```

[47]:
```
# c. Bind the clustering vector to 'car_features'
car_features_with_clusters_3 = cbind(mtcars_features, Cluster =␣
 ↪kmeans_model_3$cluster)
car_features_with_clusters_3
```

```
  Cell In[47], line 2
    car_features_with_clusters_3 = cbind(mtcars_features, Cluster =␣
 ↪kmeans_model_3$cluster)

 ↪ ^
SyntaxError: invalid syntax
```

[ ]:
```

```