Week7_7_Kmeans_cluster_3d_git

May 21, 2021

K Means Clustering

Import Libraries

[1]:
```
from sklearn.datasets import make_blobs
from sklearn.datasets import make_gaussian_quantiles
from sklearn.datasets import make_classification, make_regression
import argparse
import json
import re
import os
import sys
import plotly
import plotly.graph_objs as go
plotly.offline.init_notebook_mode()
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/home/jayanthikishore/Desktop/Analysis/Work/ML_EIT/Data/
↪Mall_Customers.csv')
df.head()
```

[1]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Data Shape and display specified columns

[2]:
```
df.shape
```

[2]:
```
(200, 5)
```

[3]:
```
#specific columns only copy
df_sel = df.iloc[:,[2,3,4]].values
df_sel.shape
```

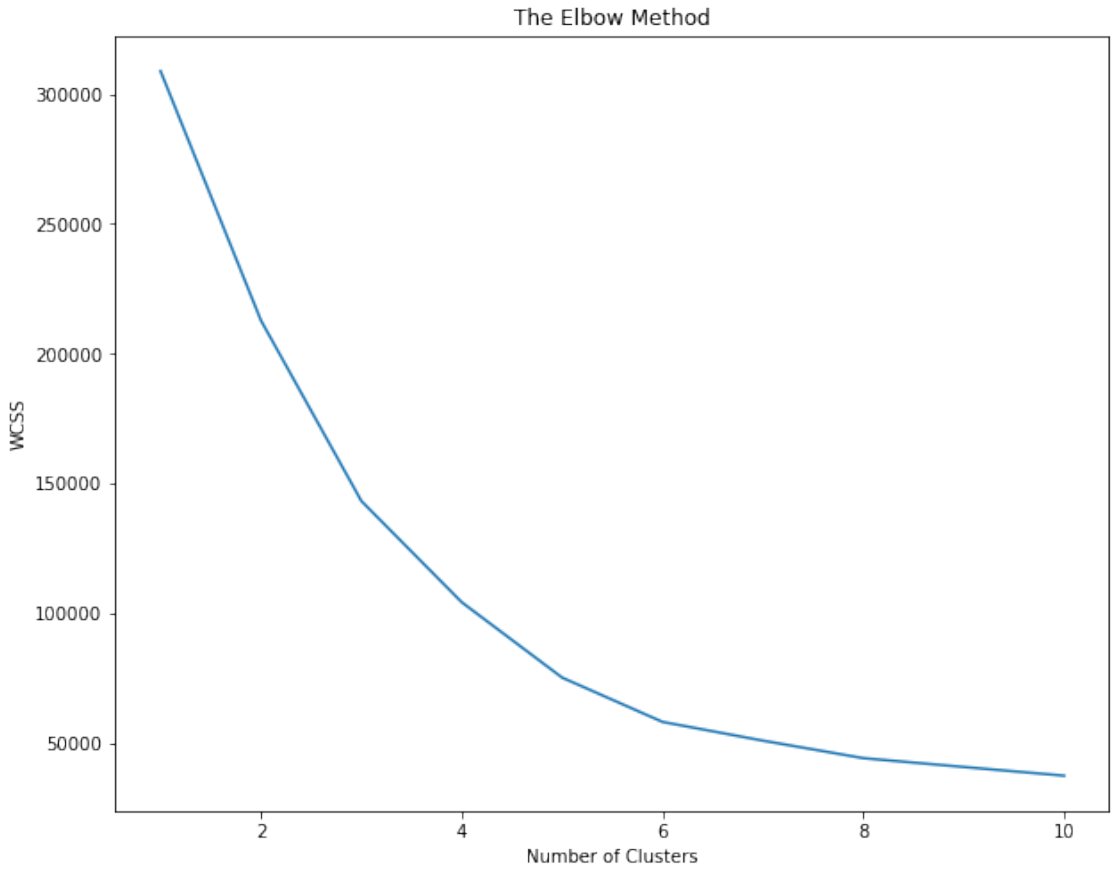
[3]: (200, 3)

Import K Means Cluster

[4]:
```
from sklearn.cluster import KMeans

fig = plt.figure(figsize=(10, 8))
WCSS = []
for i in range(1, 11):
    clf = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
    clf.fit(df_sel)
    WCSS.append(clf.inertia_) # inertia is another name for WCSS

plt.plot(range(1, 11), WCSS)
plt.title('The Elbow Method')
plt.ylabel('WCSS')
plt.xlabel('Number of Clusters')
plt.show()
```



Choosing n clusters

- *From ELBOW method the optimum value is 6 clusters*

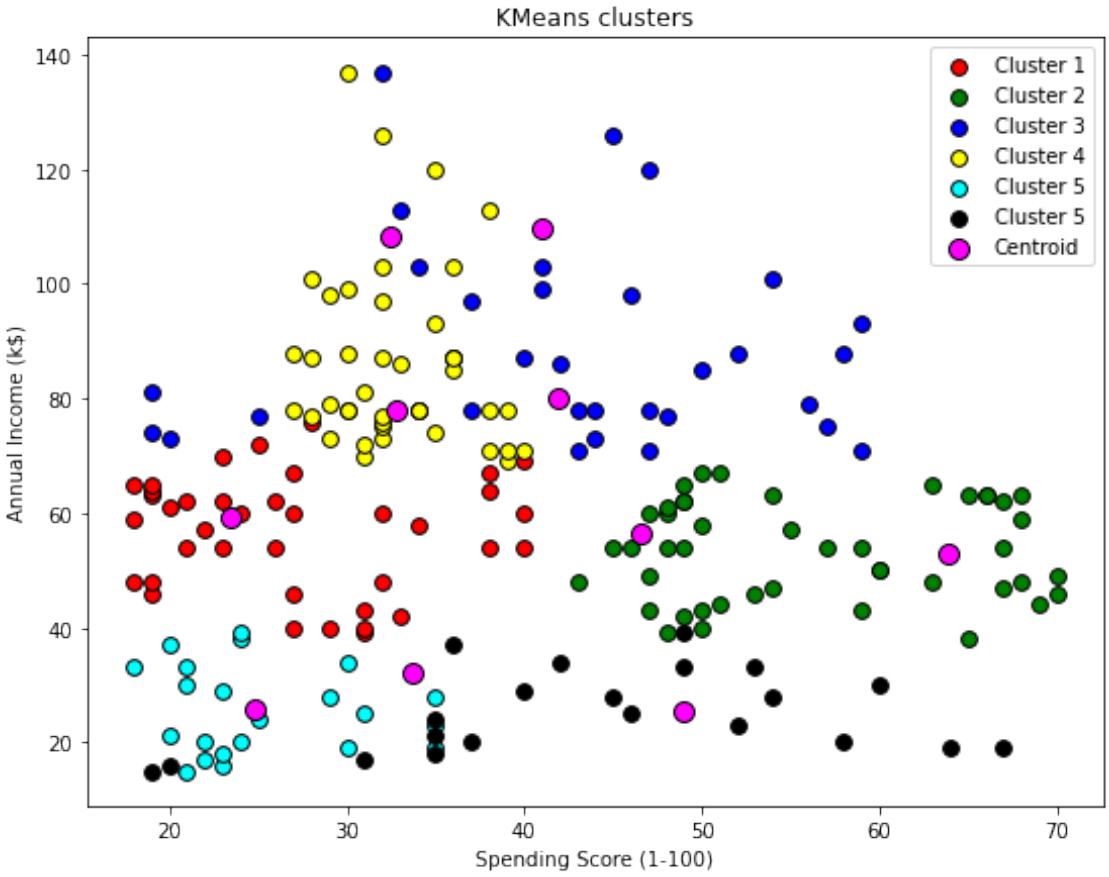
[5]:
```
clsters = KMeans(n_clusters=6, init='k-means++', max_iter=300, n_init=10, ␣
↪random_state=0)
nclsters_y = clsters.fit_predict(df_sel)
```

[6]:
```
nclsters_y
```

[6]:
```
array([5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4,
       5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 1, 4, 1, 0,
       5, 4, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 3, 0, 3, 2, 3, 2, 3, 2, 3,
       0, 3, 2, 3, 2, 3, 2, 3, 2, 3, 0, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3], dtype=int32)
```

[7]:
```
fig = plt.figure(figsize=(9, 7))
plt.scatter(df_sel[nclsters_y == 0, 0], df_sel[nclsters_y == 0, 1],␣
↪color='red', s=60, label='Cluster 1', edgecolors='black')
plt.scatter(df_sel[nclsters_y == 1, 0], df_sel[nclsters_y == 1, 1],␣
↪color='green', s=60, label='Cluster 2', edgecolors='black')
plt.scatter(df_sel[nclsters_y == 2, 0], df_sel[nclsters_y == 2, 1],␣
↪color='blue',s=60, label='Cluster 3', edgecolors='black')
plt.scatter(df_sel[nclsters_y == 3, 0], df_sel[nclsters_y == 3, 1],␣
↪color='yellow', s=60, label='Cluster 4', edgecolors='black')
plt.scatter(df_sel[nclsters_y == 4, 0], df_sel[nclsters_y == 4, 1],␣
↪color='cyan', s=60, label='Cluster 5', edgecolors='black')
plt.scatter(df_sel[nclsters_y == 5, 0], df_sel[nclsters_y == 5, 1],␣
↪color='black', s=60, label='Cluster 5', edgecolors='black')

# cluster centres
plt.scatter(clf.cluster_centers_[:, 0], clf.cluster_centers_[:, 1],␣
↪color='magenta', s=100, label='Centroid',edgecolors='black')
plt.legend()
plt.title('KMeans clusters')
plt.ylabel('Annual Income (k$)')
plt.xlabel('Spending Score (1-100)')
plt.show()
```



[8]:
```
#In this dataframe three columns are Age          Annual Income (k$)
 ↪         Spending Score (1-100)
#so, I am changing to common columns as x0, x1, and x2

def colmns_rename(dff, prefix='x'):

    dff = dff.copy()
    dff.columns = [prefix + str(i) for i in dff.columns]

    return dff

# creating dataFrame of df_sel
df_new = pd.DataFrame(df_sel)
df_new.head(5)
```

[8]:

	0	1	2
0	19	15	39
1	21	15	81
2	20	16	6

```
3  23  16  77
4  31  17  40
```

[9]:
```
#Rename column names
df_new = colmns_rename(df_new)
df_new.head(5)
```

[9]:
```
   x0  x1  x2
0  19  15  39
1  21  15  81
2  20  16   6
3  23  16  77
4  31  17  40
```

[10]:
```
#adding the nclsters_y as a y_columns
df_new['yy'] = nclsters_y
df_new.head()
```

[10]:
```
   x0  x1  x2  yy
0  19  15  39   5
1  21  15  81   4
2  20  16   6   5
3  23  16  77   4
4  31  17  40   5
```

[11]:
```
%run -i '~/Desktop/Analysis/Work/ML_EIT/Github/cluster3_3d.py'
cluster3_3d(df_new)
```

[12]:
```
%run -i '~/Desktop/Analysis/Work/ML_EIT/Github/clusters4_3d.py'
clusters4_3d(df_new)
```

[ ]:
```

```