Data Science For Everyone Using Python

Manaranjan Pradhan
manaranjan@enablecloud.com

This notebook is given as part of ***Data Science for everyone*** *workshop.*
(Forwarding this document to others is strictly prohibited.)

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sn
%matplotlib inline
```

In [2]:

```
beer = pd.read_csv( "beer.csv" )
```

In [4]:

```
beer
```

Out[4]:

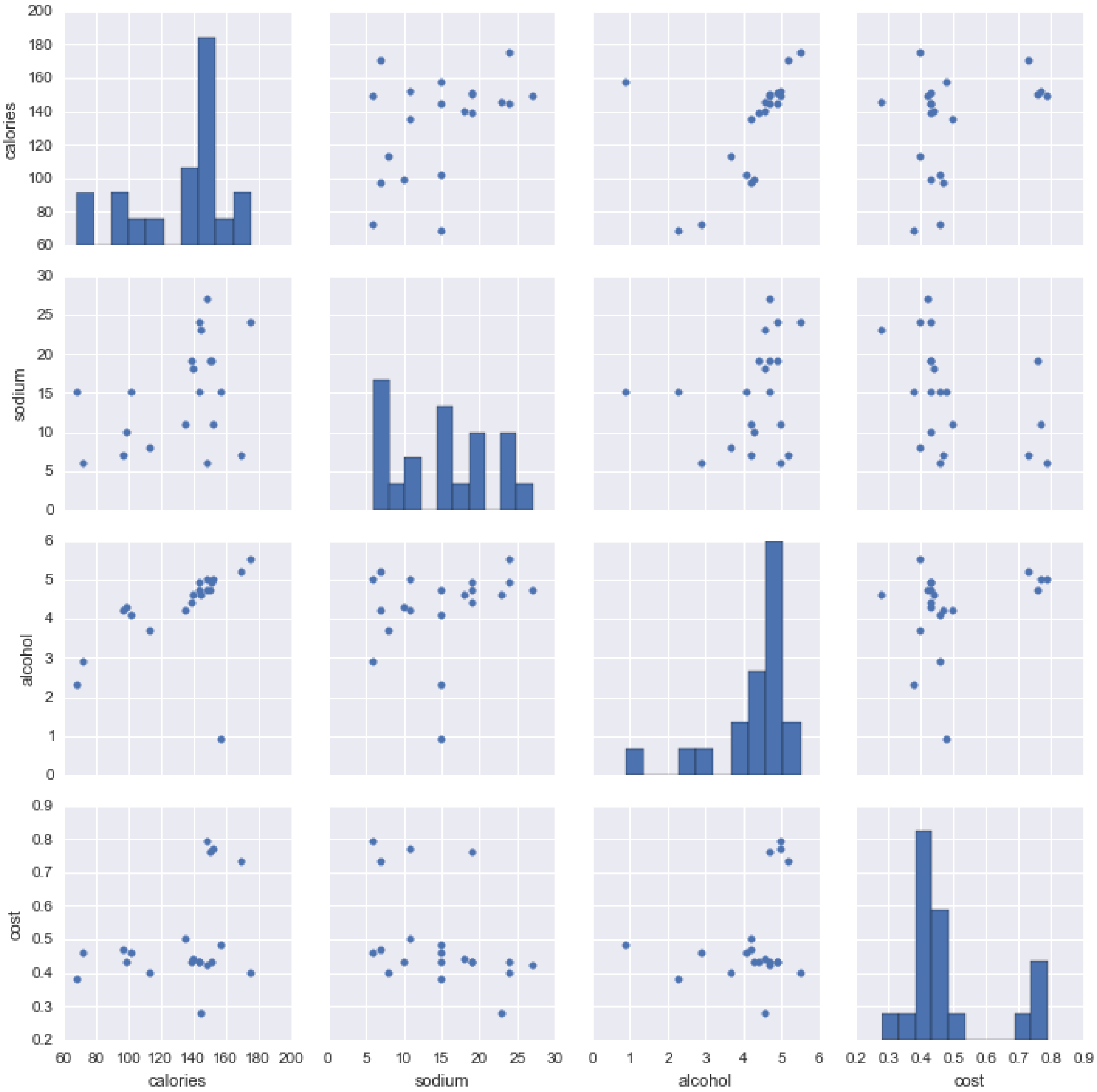
	name	calories	sodium	alcohol	cost
0	Budweiser	144	15	4.7	0.43
1	Schlitz	151	19	4.9	0.43
2	Lowenbrau	157	15	0.9	0.48
3	Kronenbourg	170	7	5.2	0.73
4	Heineken	152	11	5.0	0.77
5	Old_Milwaukee	145	23	4.6	0.28
6	Augsberger	175	24	5.5	0.40
7	Srohs_Bohemian_Style	149	27	4.7	0.42
8	Miller_Lite	99	10	4.3	0.43
9	Budweiser_Light	113	8	3.7	0.40
10	Coors	140	18	4.6	0.44
11	Coors_Light	102	15	4.1	0.46
12	Michelob_Light	135	11	4.2	0.50
13	Becks	150	19	4.7	0.76
14	Kirin	149	6	5.0	0.79
15	Pabst_Extra_Light	68	15	2.3	0.38
16	Hamms	139	19	4.4	0.43
17	Heilemans_Old_Style	144	24	4.9	0.43
18	Olympia_Goled_Light	72	6	2.9	0.46
19	Schlitz_Light	97	7	4.2	0.47

In [7]:

```
sn.pairplot( beer )
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x8a2ba8>
```



In [9]:

```
from sklearn.cluster import KMeans
```

In [10]:

```
beer.columns
```

Out[10]:

```
Index(['name', 'calories', 'sodium', 'alcohol', 'cost'], dtype='objec
t')
```

In [11]:

```
X = beer[['calories', 'sodium', 'alcohol', 'cost']]
clusters = KMeans(3)  # 3 clusters
clusters.fit( X )
```

Out[11]:

```
KMeans(copy_x=True, init='k-means++', max_iter=300, n_clusters=3, n_i
nit=10,
    n_jobs=1, precompute_distances='auto', random_state=None, tol=0.0
001,
    verbose=0)
```

In [12]:

```
clusters.cluster_centers_
```

Out[12]:

```
array([[ 150.        ,   17.        ,    4.52142857,    0.52071429],
       [  70.        ,   10.5       ,    2.6       ,    0.42      ],
       [ 102.75      ,   10.        ,    4.075     ,    0.44      ]])
```

In [13]:

```
clusters.labels_
```

Out[13]:

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

In [14]:

```
beer["cluster_id"] = clusters.labels_
```

In [15]:

```
beer
```

Out[15]:

	name	calories	sodium	alcohol	cost	cluster_id
0	Budweiser	144	15	4.7	0.43	0
1	Schlitz	151	19	4.9	0.43	0
2	Lowenbrau	157	15	0.9	0.48	0
3	Kronenbourg	170	7	5.2	0.73	0
4	Heineken	152	11	5.0	0.77	0
5	Old_Milwaukee	145	23	4.6	0.28	0
6	Augsberger	175	24	5.5	0.40	0
7	Srohs_Bohemian_Style	149	27	4.7	0.42	0
8	Miller_Lite	99	10	4.3	0.43	2
9	Budweiser_Light	113	8	3.7	0.40	2
10	Coors	140	18	4.6	0.44	0
11	Coors_Light	102	15	4.1	0.46	2
12	Michelob_Light	135	11	4.2	0.50	0
13	Becks	150	19	4.7	0.76	0
14	Kirin	149	6	5.0	0.79	0
15	Pabst_Extra_Light	68	15	2.3	0.38	1
16	Hamms	139	19	4.4	0.43	0
17	Heilemans_Old_Style	144	24	4.9	0.43	0
18	Olympia_Goled_Light	72	6	2.9	0.46	1
19	Schlitz_Light	97	7	4.2	0.47	2

In [16]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform( X )
```

In [17]:

```
clusters = KMeans(3)  # 3 clusters
clusters.fit( X )
```

Out[17]:

```
KMeans(copy_x=True, init='k-means++', max_iter=300, n_clusters=3, n_i
nit=10,
    n_jobs=1, precompute_distances='auto', random_state=None, tol=0.0
001,
    verbose=0)
```

In [18]:

```
beer["cluster_new"] = clusters.labels_
```

In [20]:

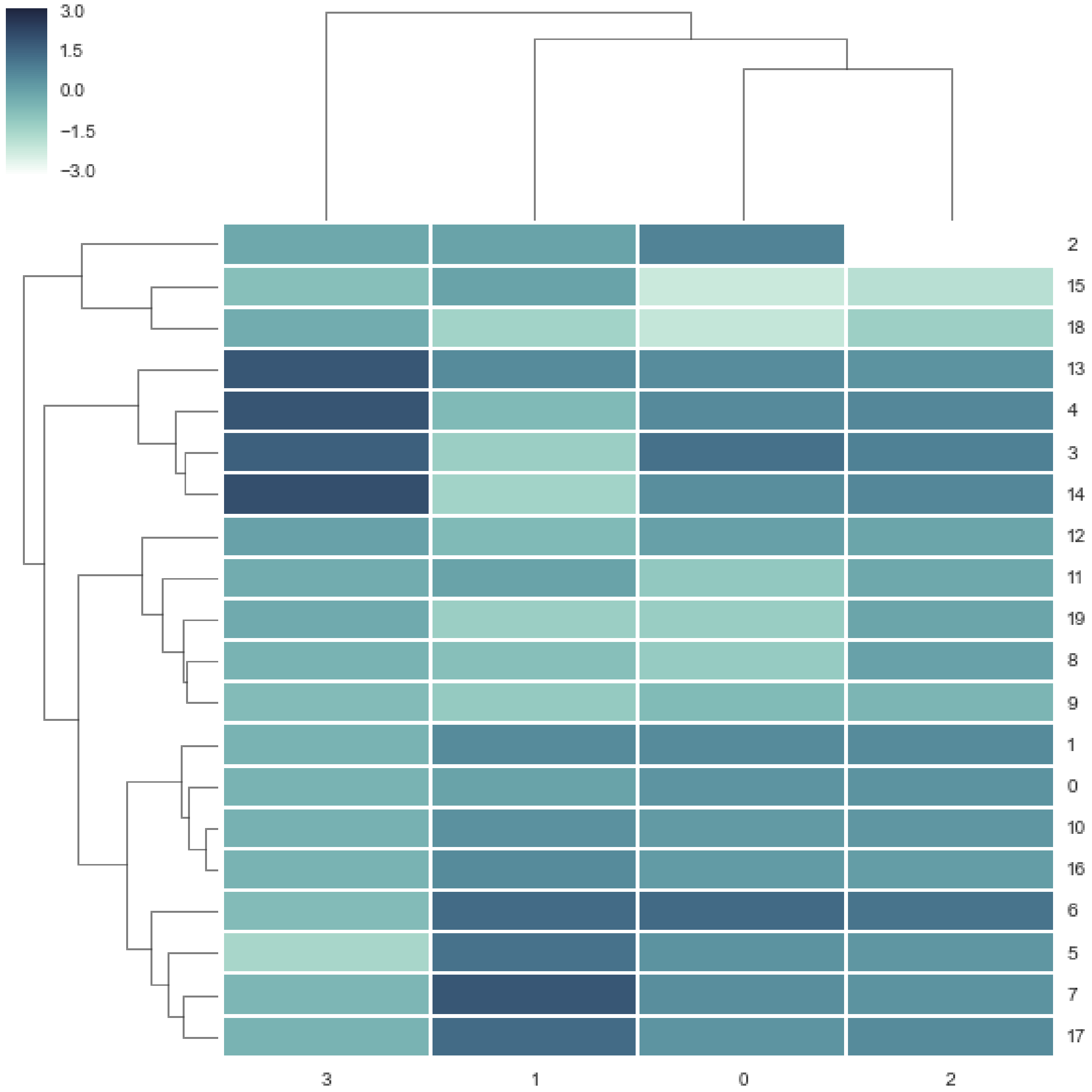
```
beer
```

Out[20]:

	name	calories	sodium	alcohol	cost	cluster_id	cluster_
0	Budweiser	144	15	4.7	0.43	0	1
1	Schlitz	151	19	4.9	0.43	0	1
2	Lowenbrau	157	15	0.9	0.48	0	1
3	Kronenbourg	170	7	5.2	0.73	0	1
4	Heineken	152	11	5.0	0.77	0	1
5	Old_Milwaukee	145	23	4.6	0.28	0	1
6	Augsberger	175	24	5.5	0.40	0	1
7	Srohs_Bohemian_Style	149	27	4.7	0.42	0	1
8	Miller_Lite	99	10	4.3	0.43	2	0
9	Budweiser_Light	113	8	3.7	0.40	2	0
10	Coors	140	18	4.6	0.44	0	1
11	Coors_Light	102	15	4.1	0.46	2	0
12	Michelob_Light	135	11	4.2	0.50	0	1
13	Becks	150	19	4.7	0.76	0	1
14	Kirin	149	6	5.0	0.79	0	1
15	Pabst_Extra_Light	68	15	2.3	0.38	1	2
16	Hamms	139	19	4.4	0.43	0	1
17	Heilemans_Old_Style	144	24	4.9	0.43	0	1
18	Olympia_Goled_Light	72	6	2.9	0.46	1	2
19	Schlitz_Light	97	7	4.2	0.47	2	0

In [21]:

```
cmap = sn.cubehelix_palette(as_cmap=True, rot=-.3, light=1)
g = sn.clustermap(X_scaled, cmap=cmap, linewidths=.5)
```



In [24]:

```
clusters = KMeans(4)  # 3 clusters
clusters.fit( X )
beer["cluster_final"] = clusters.labels_
```

In [25]:

```
beer[['name', 'calories', 'sodium', 'alcohol', 'cost', 'cluster_final']]
```

Out[25]:

	name	calories	sodium	alcohol	cost	cluster_final
0	Budweiser	144	15	4.7	0.43	0
1	Schlitz	151	19	4.9	0.43	0
2	Lowenbrau	157	15	0.9	0.48	0
3	Kronenbourg	170	7	5.2	0.73	3
4	Heineken	152	11	5.0	0.77	0
5	Old_Milwaukee	145	23	4.6	0.28	0
6	Augsberger	175	24	5.5	0.40	3
7	Srohs_Bohemian_Style	149	27	4.7	0.42	0
8	Miller_Lite	99	10	4.3	0.43	1
9	Budweiser_Light	113	8	3.7	0.40	1
10	Coors	140	18	4.6	0.44	0
11	Coors_Light	102	15	4.1	0.46	1
12	Michelob_Light	135	11	4.2	0.50	0
13	Becks	150	19	4.7	0.76	0
14	Kirin	149	6	5.0	0.79	0
15	Pabst_Extra_Light	68	15	2.3	0.38	2
16	Hamms	139	19	4.4	0.43	0
17	Heilemans_Old_Style	144	24	4.9	0.43	0
18	Olympia_Goled_Light	72	6	2.9	0.46	2
19	Schlitz_Light	97	7	4.2	0.47	1

In [31]:

```
beer_0 = beer[['name', 'calories', 'sodium', 'alcohol', 'cost', 'cluster_final']]
[beer.cluster_final == 0]
```

In [33]:

```
beer_0
```

Out[33]:

	name	calories	sodium	alcohol	cost	cluster_final
0	Budweiser	144	15	4.7	0.43	0
1	Schlitz	151	19	4.9	0.43	0
2	Lowenbrau	157	15	0.9	0.48	0
4	Heineken	152	11	5.0	0.77	0
5	Old_Milwaukee	145	23	4.6	0.28	0
7	Srohs_Bohemian_Style	149	27	4.7	0.42	0
10	Coors	140	18	4.6	0.44	0
12	Michelob_Light	135	11	4.2	0.50	0
13	Becks	150	19	4.7	0.76	0
14	Kirin	149	6	5.0	0.79	0
16	Hamms	139	19	4.4	0.43	0
17	Heilemans_Old_Style	144	24	4.9	0.43	0

In [34]:

```
beer_0.mean()
```

Out[34]:

```
calories         146.250000
sodium            17.250000
alcohol            4.383333
cost               0.513333
cluster_final      0.000000
dtype: float64
```

In [36]:

```
beer_1 = beer[['name', 'calories', 'sodium', 'alcohol', 'cost', 'cluster_final']]
[beer.cluster_final == 1]
```

In [38]:

```
beer_1
```

Out[38]:

	name	calories	sodium	alcohol	cost	cluster_final
8	Miller_Lite	99	10	4.3	0.43	1
9	Budweiser_Light	113	8	3.7	0.40	1
11	Coors_Light	102	15	4.1	0.46	1
19	Schlitz_Light	97	7	4.2	0.47	1

In [39]:

```
beer_1.mean()
```

Out[39]:

```
calories         102.750
sodium            10.000
alcohol            4.075
cost               0.440
cluster_final      1.000
dtype: float64
```

In [40]:

```
beer_2 = beer[['name', 'calories', 'sodium', 'alcohol', 'cost', 'cluster_final']]
[beer.cluster_final == 2]
```

In [41]:

```
beer_2
```

Out[41]:

	name	calories	sodium	alcohol	cost	cluster_final
15	Pabst_Extra_Light	68	15	2.3	0.38	2
18	Olympia_Goled_Light	72	6	2.9	0.46	2

In [42]:

```
beer_2.mean()
```

Out[42]:

```
calories         70.00
sodium           10.50
alcohol           2.60
cost              0.42
cluster_final     2.00
dtype: float64
```

In [43]:

```
beer_3 = beer[['name', 'calories', 'sodium', 'alcohol', 'cost', 'cluster_final']]
[beer.cluster_final == 3]
```

In [44]:

```
beer_3
```

Out[44]:

	name	calories	sodium	alcohol	cost	cluster_final
3	Kronenbourg	170	7	5.2	0.73	3
6	Augsberger	175	24	5.5	0.40	3

In [45]:

```
beer_3.mean()
```

Out[45]:

```
calories         172.500
sodium            15.500
alcohol            5.350
cost               0.565
cluster_final      3.000
dtype: float64
```

In [ ]:

```

```