Lecture23

April 30, 2024

[1]:
```
# Machine Learning
# Clustering
# Classification
# Regression
```

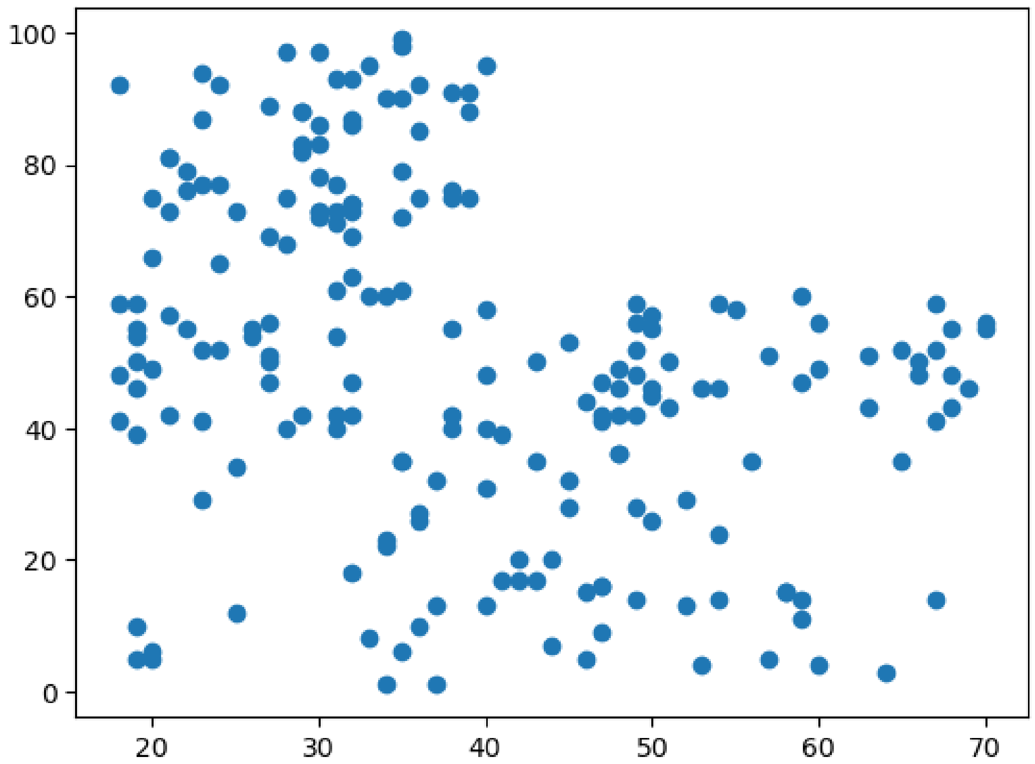
1 Clustering

[6]:
```
import pandas as pd
df=pd.read_csv('datafiles/customers.csv')
df.head()
```

[6]:
```
   CustomerID  Gender  Age  AnnualIncome_in_k  SpendingScore
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
```

[8]:
```
tmpdf=df[['Age','SpendingScore']]
```

[10]:
```
import matplotlib.pyplot as plt
plt.scatter(tmpdf.Age, tmpdf.SpendingScore);
```



[14]:
```
from sklearn.cluster import KMeans
my_kmeans=KMeans(3)
my_kmeans.fit(tmpdf)
my_kmeans.cluster_centers_
```

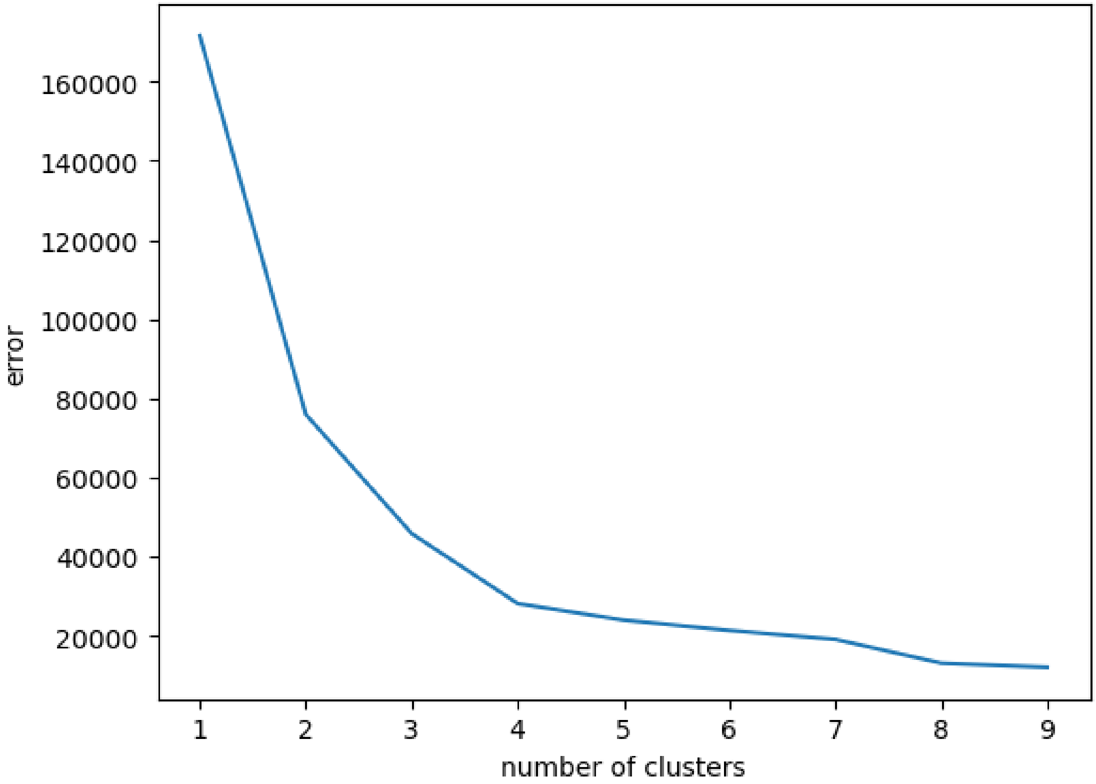
[14]:
```
array([[29.56451613, 80.74193548],
       [43.02173913, 47.59782609],
       [43.02173913, 14.23913043]])
```

[20]:
```
my_kmeans.inertia_
```

[20]:
```
45844.53681626927
```

[24]:
```
errors=[]
for n in range(1,10):
    my_kmeans = KMeans(n)
    my_kmeans.fit(tmpdf)
    errors.append(my_kmeans.inertia_)
```

[28]:
```
plt.plot(range(1,10), errors)
plt.xlabel('number of clusters')
plt.ylabel('error')
plt.show()
```



[29]:
```
my_kmeans = KMeans(4)
my_kmeans.fit(tmpdf)
```

[29]:
```
KMeans(n_clusters=4)
```

[30]:
```
tmpdf['clusters'] = my_kmeans.labels_
```

```
/tmp/ipykernel_1538/959676023.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  tmpdf['clusters'] = my_kmeans.labels_
```

[34]:
```
tmpdf
```

[34]:
```
     Age  SpendingScore  clusters
0     19             39         2
1     21             81         1
```

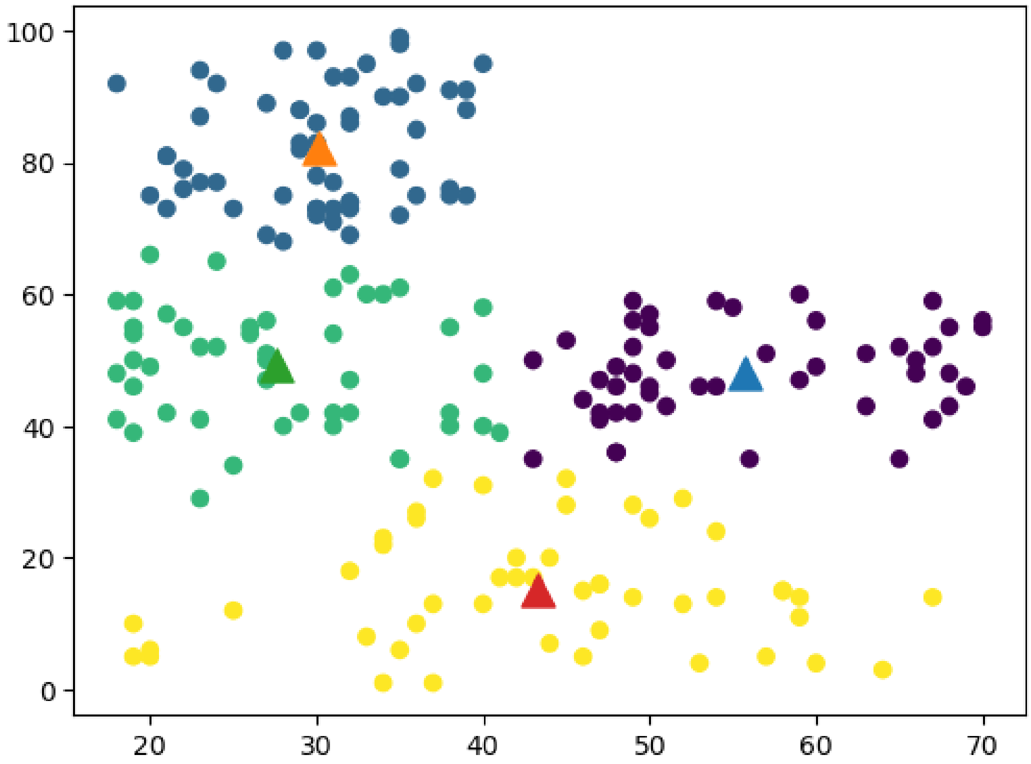
```
2     20             6         3
3     23            77         1
4     31            40         2
..    …            …        …
195   35            79         1
196   45            28         3
197   32            74         1
198   32            18         3
199   30            83         1

[200 rows x 3 columns]
```

[41]:

```
# c0= tmpdf.loc[tmpdf.clusters==0]
# c1= tmpdf.loc[tmpdf.clusters==1]
# c2= tmpdf.loc[tmpdf.clusters==2]
# c3= tmpdf.loc[tmpdf.clusters==3]
# plt.scatter(c0['Age'], c0['SpendingScore'])
# plt.scatter(c1['Age'], c1['SpendingScore'])
# plt.scatter(c2['Age'], c2['SpendingScore'])
# plt.scatter(c3['Age'], c3['SpendingScore'])
plt.scatter(tmpdf.Age, tmpdf.SpendingScore,c=tmpdf.clusters)
plt.scatter(my_kmeans.cluster_centers_[0,0],my_kmeans.cluster_centers_[0,1], s=150, marker='^')
plt.scatter(my_kmeans.cluster_centers_[1,0],my_kmeans.cluster_centers_[1,1], s=150, marker='^')
plt.scatter(my_kmeans.cluster_centers_[2,0],my_kmeans.cluster_centers_[2,1], s=150, marker='^')
plt.scatter(my_kmeans.cluster_centers_[3,0],my_kmeans.cluster_centers_[3,1], s=150, marker='^')
```

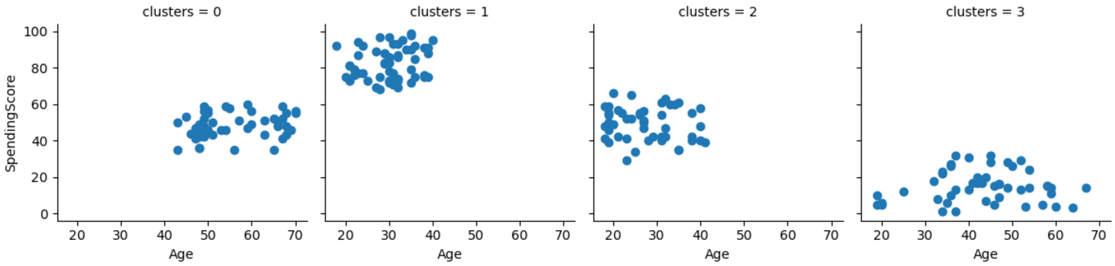
[41]:

```
<matplotlib.collections.PathCollection at 0x7fd4b689c1d0>
```



```
[44]: import seaborn as sns
      sns.FacetGrid(data=tmpdf, col='clusters').map(plt.scatter, 'Age',␣
       ↪'SpendingScore')
```

```
[44]: <seaborn.axisgrid.FacetGrid at 0x7fd4b63f54d0>
```



```
[ ]:
```

2 Data Classification

```
[ ]:
```

```
[ ]:
```

3