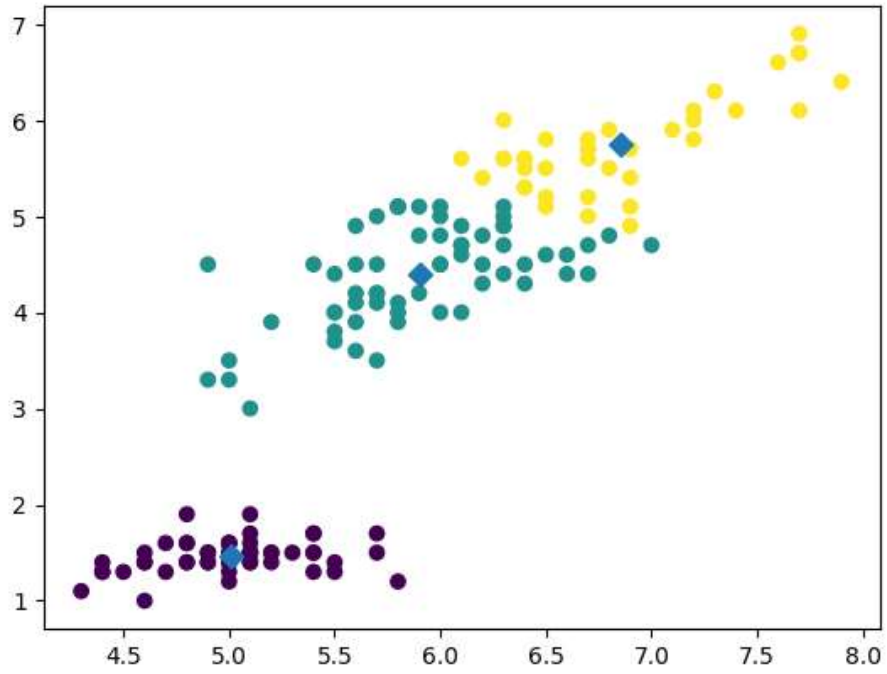
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.
```

```
['wineuci', 'seed-from-uci']
```

```
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans


iris=load_iris()
samples=iris.data
model=KMeans(n_clusters=3)
model.fit(samples)
labels=model.predict(samples)
print(labels)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2
 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2
 2 1]
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
```

```
# plotting sepal length and petal length
xs=samples[:,0]
ys=samples[:,2]
plt.scatter(xs,ys,c=labels)
centroids=model.cluster_centers_
centroids_x=centroids[:,0]
centroids_y=centroids[:,2]
plt.scatter(centroids_x,centroids_y,marker='D',s=50)
plt.show()
```




```
seeds=pd.read_csv('../input/seed-from-uci/Seed_Data.csv')
seeds.head()
```

	A	P	C	LK	WK	A_Coef	LKG	target
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	0
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	0
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	0
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	0
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	0

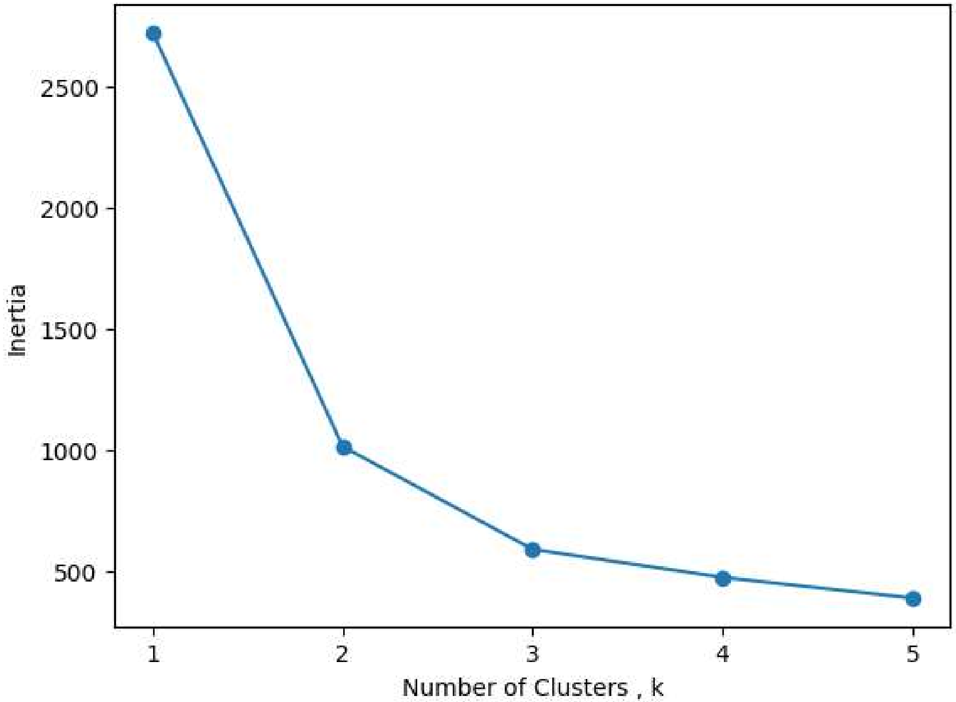
```
seed_dict={
    0:'Kama',
    1:'Rosa',
    2:'Canadian'
}
seed_list=seeds['target'].map(seed_dict).tolist()
```

```
# Preparing seeds for clustering by dropping the target column

seeds=seeds.drop(['target'],axis=1)
seeds.head()
```

	A	P	C	LK	WK	A_Coef	LKG
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175

```
ks=range(1,6)
inertias=[]

for k in ks:
    model=KMeans(n_clusters=k)
    model.fit(seeds)
    inertias.append(model.inertia_)

plt.plot(ks,inertias,'-o')
plt.xlabel('Number of Clusters , k')
plt.ylabel('Inertia')
plt.xticks(ks)
plt.show()

# Inertia decreases from 3 to 4 very slowly , so 3 can be a good choice
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
```



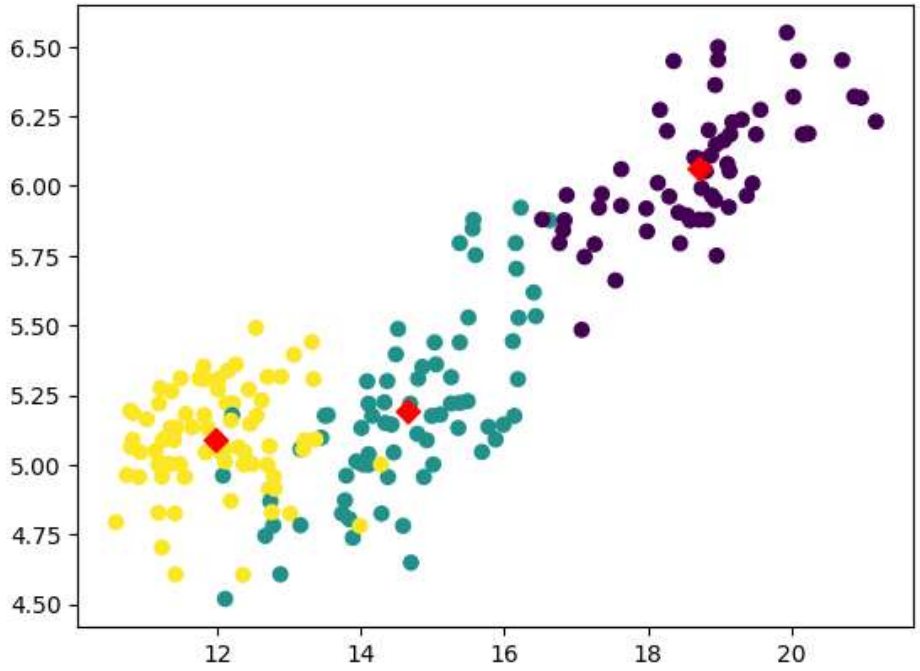

```
seeds.head()
```

	A	P	C	LK	WK	A_Coef	LKG
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175

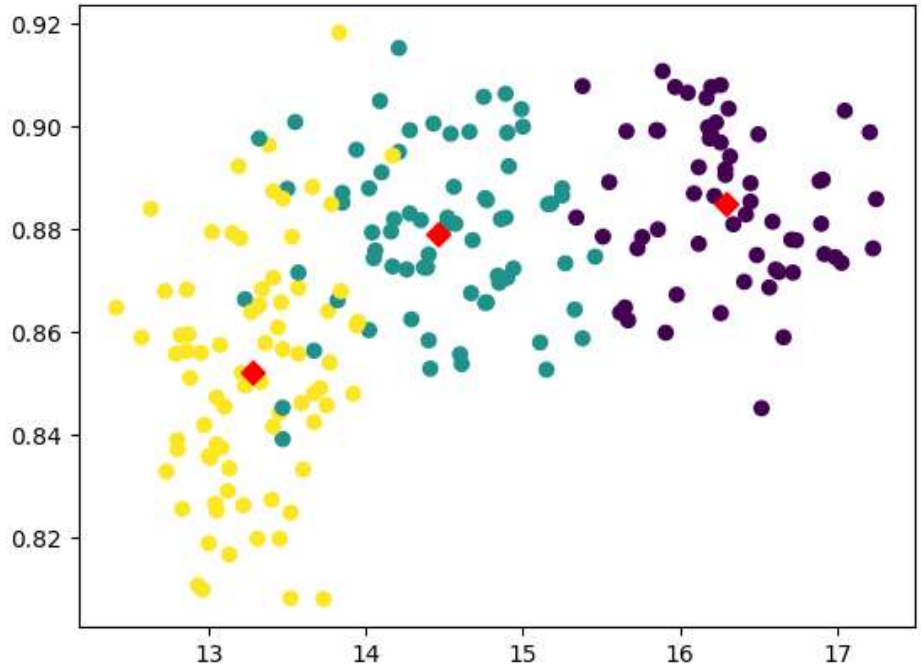
```
model=KMeans(n_clusters=3)
seed_labels=model.fit_predict(seeds)
centroids=model.cluster_centers_
centroids
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
array([[18.72180328, 16.29737705,  0.88508689,  6.20893443,  3.72267213,
         3.60359016,  6.06609836],
       [14.64847222, 14.46041667,  0.87916667,  5.56377778,  3.27790278,
         2.64893333,  5.19231944],
       [11.96441558, 13.27480519,  0.8522    ,  5.22928571,  2.87292208,
         4.75974026,  5.08851948]])
```

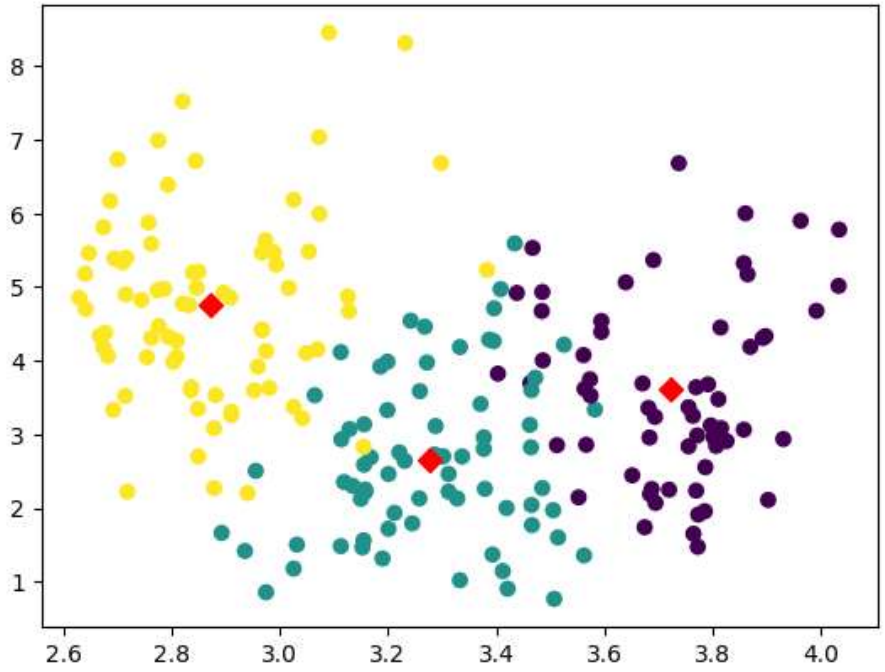
```
# A vs LKG
xs_A=seeds.iloc[:,0]
ys_LKG=seeds.iloc[:,-1]
centroids_xs_A=centroids[:,0]
centroids_ys_LKG=centroids[:,-1]
plt.scatter(xs_A,ys_LKG,c=seed_labels)
plt.scatter(centroids_xs_A,centroids_ys_LKG,marker='D',s=50,c='red')
plt.show()
```




```
# P vs C
xs_A=seeds.iloc[:,1]
ys_LKG=seeds.iloc[:,2]
centroids_xs_A=centroids[:,1]
centroids_ys_LKG=centroids[:,2]
plt.scatter(xs_A,ys_LKG,c=seed_labels)
plt.scatter(centroids_xs_A,centroids_ys_LKG,marker='D',s=50,c='red')
plt.show()
```




```
# WK vs A_Coeff
xs_A=seeds.iloc[:,4]
ys_LKG=seeds.iloc[:,5]
centroids_xs_A=centroids[:,4]
centroids_ys_LKG=centroids[:,5]
plt.scatter(xs_A,ys_LKG,c=seed_labels)
plt.scatter(centroids_xs_A,centroids_ys_LKG,marker='D',s=50,c='red')
plt.show()
```




```
wine=pd.read_csv('../input/wineuci/Wine.csv',header=None)
wine.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

```
wine.columns=['Class','Alcohol','Malic_acid','Ash','Alcalinity_of_ash','Magnesium','Total_phenols','Flavanoids','Nonflavanoid_phenols','Proanthocyanins','Col
```

```
wine.head()
```

	Class	Alcohol	Malic_acid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyanins	Color_intensity	Hue
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04

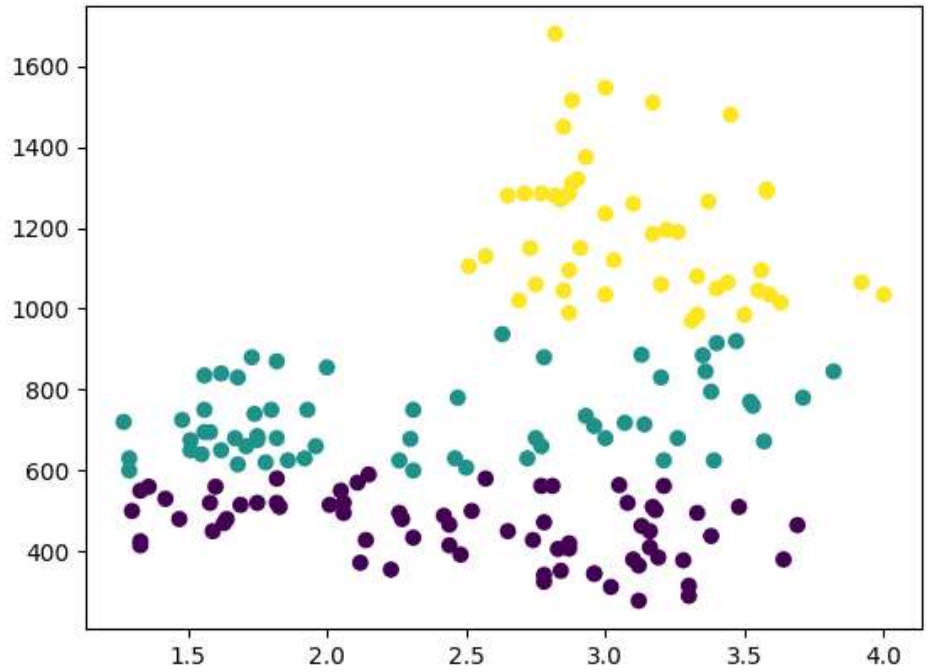
```
wine_class=wine['Class']
wine.drop('Class',axis=1,inplace=True)


model=KMeans(n_clusters=3)
labels=model.fit_predict(wine)

df=pd.DataFrame({'labels':labels , 'class':wine_class})
ct=pd.crosstab(df['labels'],df['class'])
ct
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
```

class	1	2	3
labels			
0	0	50	19
1	13	20	29
2	46	1	0

```
xs=wine.loc[:,'OD280']
ys=wine.loc[:,'Proline']
plt.scatter(xs,ys,c=labels)
```

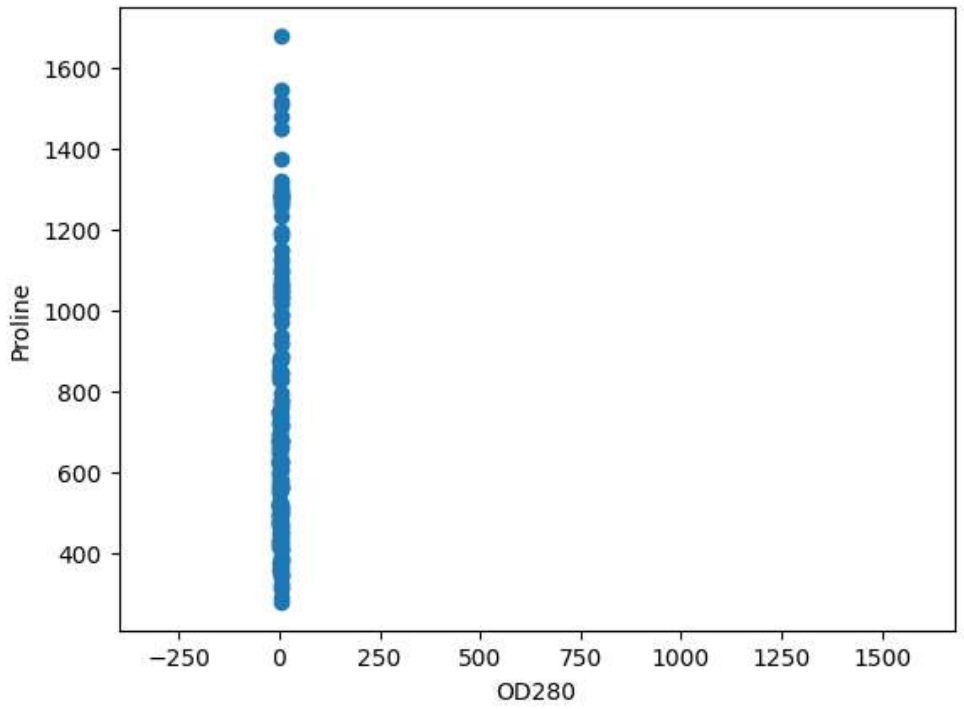
```
<matplotlib.collections.PathCollection at 0x788fbe55b7c0>
```




```
wine.var(axis=0)
```

```
Alcohol                     0.659062
Malic_acid                  1.248015
Ash                         0.075265
Alcalinity_of_ash          11.152686
Magnesium                 203.989335
Total_phenols               0.391690
Flavanoids                  0.997719
Nonflavanoid_phenols        0.015489
Proanthocyanins             0.327595
Color_intensity             5.374449
Hue                         0.052245
OD280                       0.504086
Proline                 99166.717355
dtype: float64
```

```
# Variance comparison between Proline and OD280
plt.scatter(wine['OD280'],wine['Proline'])
plt.xlim(-400,max(wine['Proline']))
plt.xlabel('OD280')
plt.ylabel('Proline')
plt.show()
```



Applying Standard Scaler (then KMeans in sklearn Pipeline)

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
scaler=StandardScaler()
kmeans=KMeans(n_clusters=3)
pipeline=make_pipeline(scaler,kmeans)
pipeline.fit(wine)
labels=pipeline.predict(wine)
```

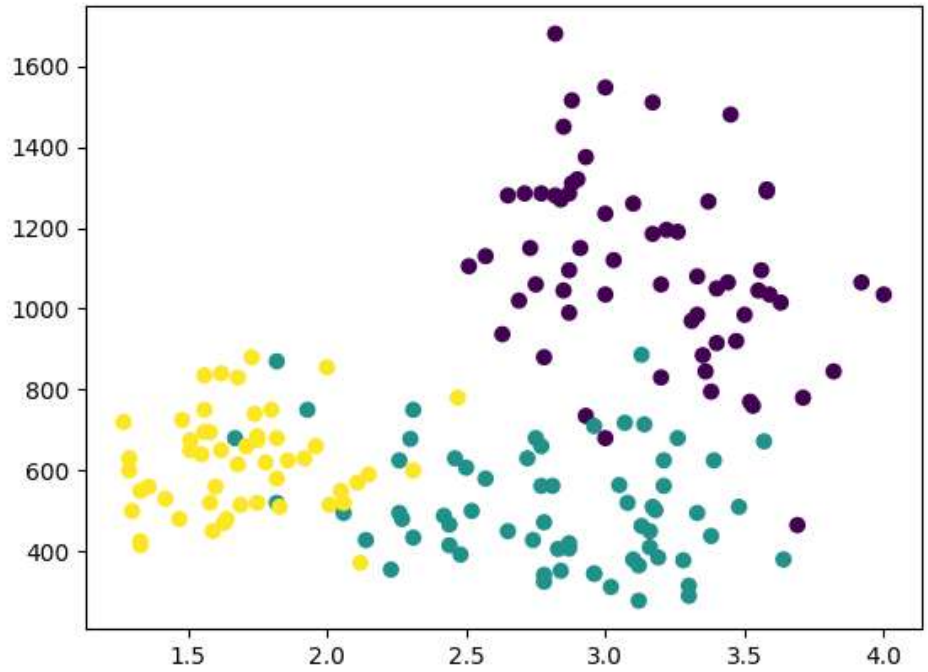
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
```

```
df=pd.DataFrame({'labels':labels,'class':wine_class   })
ct=pd.crosstab(df['labels'],df['class'])
ct
# After scaling we get tight clusters
```

class	1	2	3
labels			
0	59	3	0
1	0	65	0
2	0	3	48

```
xs=wine.loc[:,'OD280']
ys=wine.loc[:,'Proline']
plt.scatter(xs,ys,c=labels)
```

```
<matplotlib.collections.PathCollection at 0x788fbe5e24d0>
```



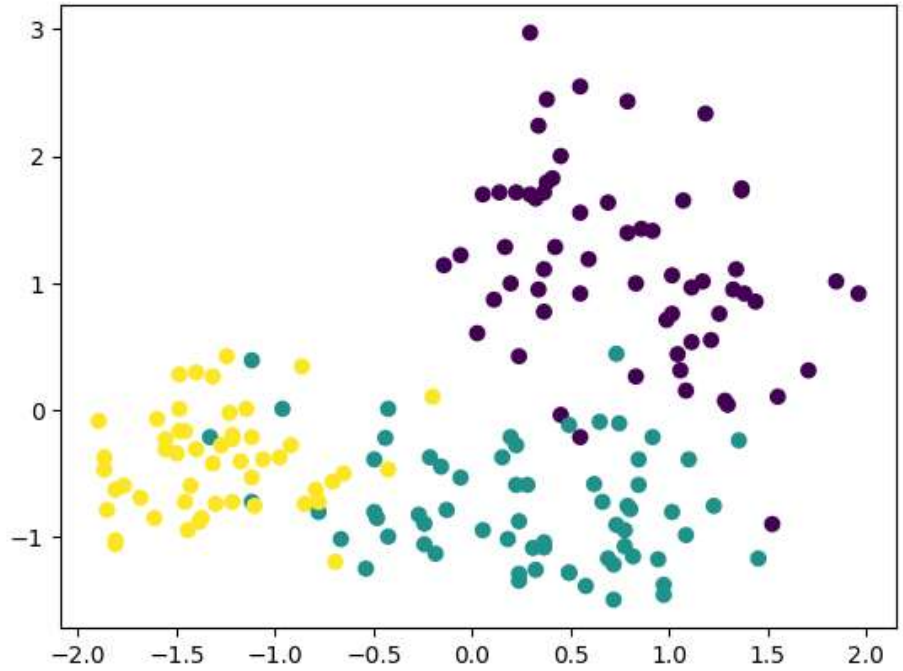

```
scaled_wine=scaler.fit_transform(wine)
scaled_wine.var(axis=0)
```

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
scaled_wine=pd.DataFrame(scaled_wine)
scaled_wine.columns=['Alcohol','Malic_acid','Ash','Alcalinity_of_ash','Magnesium','Total_phenols','Flavanoids','Nonflavanoid_phenols','Proanthocyanins','Colo
```

```
lables=KMeans(n_clusters=3).fit_predict(scaled_wine)
xs=scaled_wine.loc[:,'OD280']
ys=scaled_wine.loc[:,'Proline']
plt.scatter(xs,ys,c=labels)
```

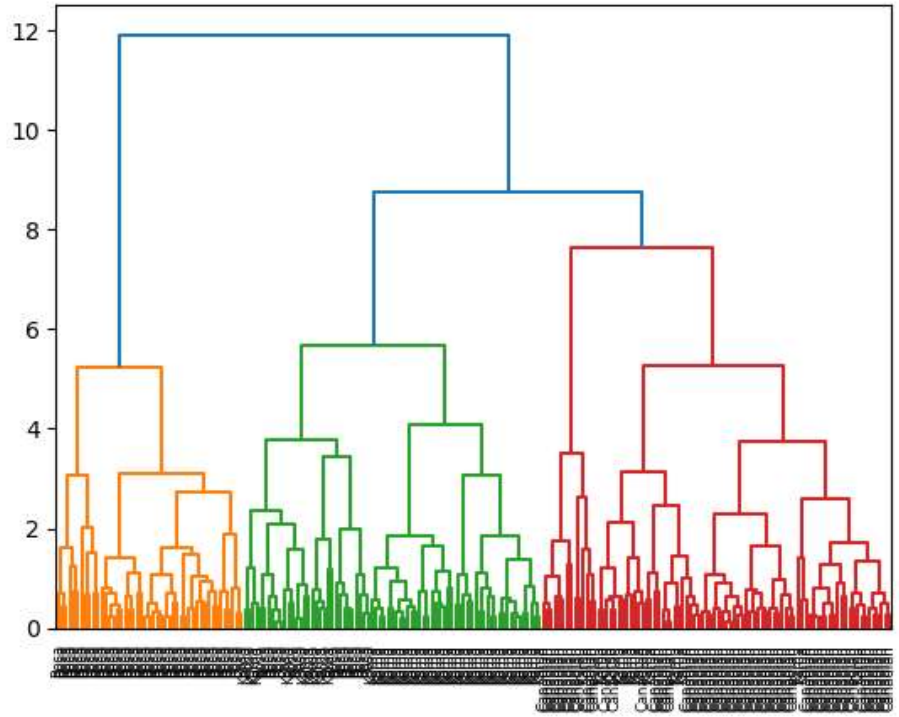
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4
  warnings.warn(
<matplotlib.collections.PathCollection at 0x788f817f94e0>
```



Visualization with hierarchical clustering and t-SNE

Hierarchical Clustering

```
from scipy.cluster.hierarchy import linkage,dendrogram , fcluster


mergings=linkage(seeds,method='complete')
#plt.figure(figsize=(50,50))
dendrogram(mergings,labels=seed_list,leaf_rotation=90,leaf_font_size=6)
plt.show()
```




```
labels=fcluster(mergings,6,criterion='distance')
print(labels)
```

```
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 4 2 2 2 4 2 2 4 4 2 2 4 2 2 2 2 2 2
 2 2 3 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 2 2 2 4 2 2 2 1
 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1
 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 4 3 3 4 4 3 4 4
 4 4 4 3 4 4 4 3 4 4 4 4 4 4 4 4 3 4 4 4 4 4 3 3 4 4 4 4 4 4 3 4 4 4 4 4 3
 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 3 4 4]
```

```
df=pd.DataFrame({'labels':labels,
             'seed_variety':seed_list})

ct=pd.crosstab(df['labels'],df['seed_variety'])
ct
```

seed_variety	Canadian	Kama	Rosa
labels			
1	0	0	47
2	0	52	23

```
from sklearn.datasets import load_iris
from sklearn.manifold import TSNE
iris=load_iris()
data=iris.data
species=iris.target


import seaborn as sns
model=TSNE(learning_rate=100)
transformed=model.fit_transform(data)
xs=transformed[:,0]
ys=transformed[:,1]
plt.scatter(xs,ys,c=species)
#plt.legend(species)
#sns.scatterplot(xs,ys,hue=species)
plt.show()
```