

```
In [1]: import pandas as pd
cust_df = pd.read_csv("Unit06 iris.csv")
cust_df
```

```
Out[1]:
```

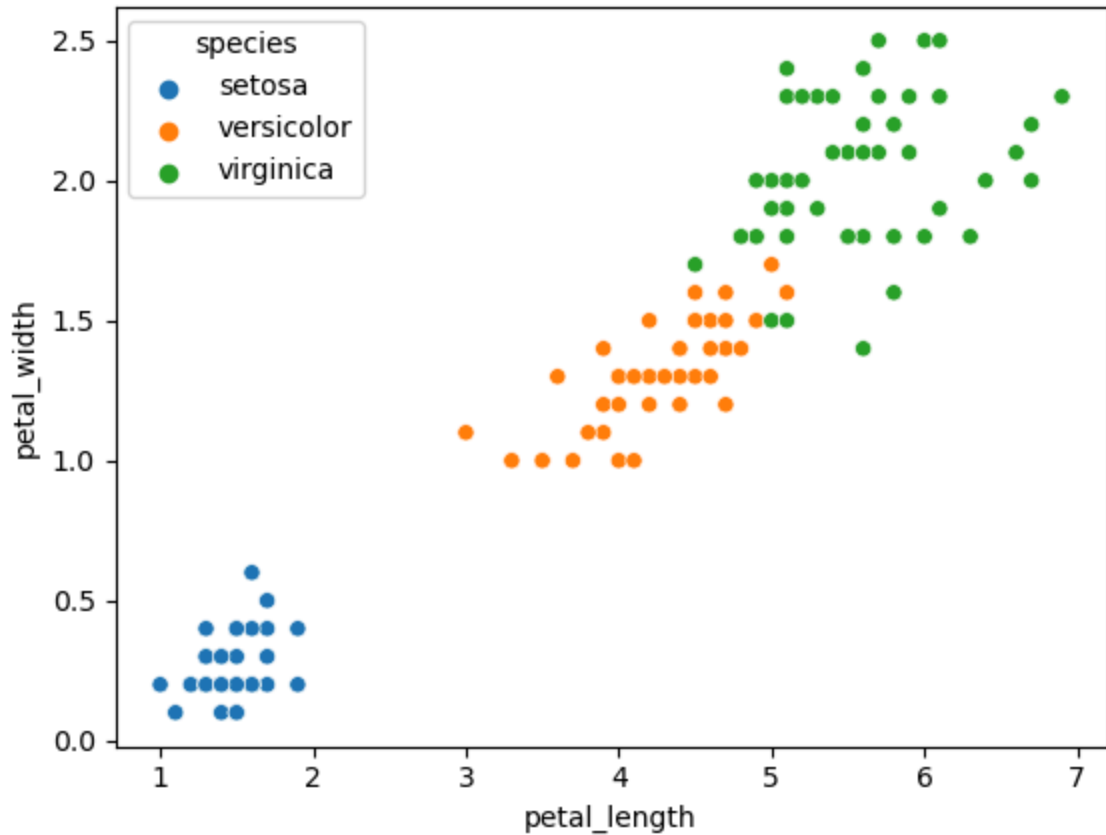
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
In [5]: sns.scatterplot(data=cust_df, x='petal_length', y='petal_width', hue='species')
```

C:\Users\Pan Mour\anaconda3\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\Pan Mour\anaconda3\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\Pan Mour\anaconda3\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\Pan Mour\anaconda3\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\Pan Mour\anaconda3\Lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):

```
Out[5]: <Axes: xlabel='petal_length', ylabel='petal_width'>
```



```
In [6]: cust_df = cust_df.drop('species', axis=1)
cust_df
```

```
Out[6]:
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
In [7]: from sklearn.preprocessing import StandardScaler
import numpy as np
```

```
X = cust_df.values[:,:]  
X = np.nan_to_num(X)  
Clus_dataSet = StandardScaler().fit_transform(X)  
Clus_dataSet
```

```
Out[7]: array([[ -9.00681170e-01,  1.03205722e+00, -1.34127240e+00,
                -1.31297673e+00],
               [-1.14301691e+00, -1.24957601e-01, -1.34127240e+00,
                -1.31297673e+00],
               [-1.38535265e+00,  3.37848329e-01, -1.39813811e+00,
                -1.31297673e+00],
               [-1.50652052e+00,  1.06445364e-01, -1.28440670e+00,
                -1.31297673e+00],
               [-1.02184904e+00,  1.26346019e+00, -1.34127240e+00,
                -1.31297673e+00],
               [-5.37177559e-01,  1.95766909e+00, -1.17067529e+00,
                -1.05003079e+00],
               [-1.50652052e+00,  8.00654259e-01, -1.34127240e+00,
                -1.18150376e+00],
               [-1.02184904e+00,  8.00654259e-01, -1.28440670e+00,
                -1.31297673e+00],
               [-1.74885626e+00, -3.56360566e-01, -1.34127240e+00,
                -1.31297673e+00],
               [-1.14301691e+00,  1.06445364e-01, -1.28440670e+00,
                -1.44444970e+00],
               [-5.37177559e-01,  1.49486315e+00, -1.28440670e+00,
                -1.31297673e+00],
               [-1.26418478e+00,  8.00654259e-01, -1.22754100e+00,
                -1.31297673e+00],
               [-1.26418478e+00, -1.24957601e-01, -1.34127240e+00,
                -1.44444970e+00],
               [-1.87002413e+00, -1.24957601e-01, -1.51186952e+00,
                -1.44444970e+00],
               [-5.25060772e-02,  2.18907205e+00, -1.45500381e+00,
                -1.31297673e+00],
               [-1.73673948e-01,  3.11468391e+00, -1.28440670e+00,
                -1.05003079e+00],
               [-5.37177559e-01,  1.95766909e+00, -1.39813811e+00,
                -1.05003079e+00],
               [-9.00681170e-01,  1.03205722e+00, -1.34127240e+00,
                -1.18150376e+00],
               [-1.73673948e-01,  1.72626612e+00, -1.17067529e+00,
                -1.18150376e+00],
               [-9.00681170e-01,  1.72626612e+00, -1.28440670e+00,
                -1.18150376e+00],
               [-5.37177559e-01,  8.00654259e-01, -1.17067529e+00,
                -1.31297673e+00],
               [-9.00681170e-01,  1.49486315e+00, -1.28440670e+00,
                -1.05003079e+00],
               [-1.50652052e+00,  1.26346019e+00, -1.56873522e+00,
                -1.31297673e+00],
               [-9.00681170e-01,  5.69251294e-01, -1.17067529e+00,
                -9.18557817e-01],
               [-1.26418478e+00,  8.00654259e-01, -1.05694388e+00,
                -1.31297673e+00],
               [-1.02184904e+00, -1.24957601e-01, -1.22754100e+00,
                -1.31297673e+00],
               [-1.02184904e+00,  8.00654259e-01, -1.22754100e+00,
                -1.05003079e+00],
               [-7.79513300e-01,  1.03205722e+00, -1.28440670e+00,
                -1.31297673e+00],
```

[-7.79513300e-01, 8.00654259e-01, -1.34127240e+00,
-1.31297673e+00],
[-1.38535265e+00, 3.37848329e-01, -1.22754100e+00,
-1.31297673e+00],
[-1.26418478e+00, 1.06445364e-01, -1.22754100e+00,
-1.31297673e+00],
[-5.37177559e-01, 8.00654259e-01, -1.28440670e+00,
-1.05003079e+00],
[-7.79513300e-01, 2.42047502e+00, -1.28440670e+00,
-1.44444970e+00],
[-4.16009689e-01, 2.65187798e+00, -1.34127240e+00,
-1.31297673e+00],
[-1.14301691e+00, 1.06445364e-01, -1.28440670e+00,
-1.44444970e+00],
[-1.02184904e+00, 3.37848329e-01, -1.45500381e+00,
-1.31297673e+00],
[-4.16009689e-01, 1.03205722e+00, -1.39813811e+00,
-1.31297673e+00],
[-1.14301691e+00, 1.06445364e-01, -1.28440670e+00,
-1.44444970e+00],
[-1.74885626e+00, -1.24957601e-01, -1.39813811e+00,
-1.31297673e+00],
[-9.00681170e-01, 8.00654259e-01, -1.28440670e+00,
-1.31297673e+00],
[-1.02184904e+00, 1.03205722e+00, -1.39813811e+00,
-1.18150376e+00],
[-1.62768839e+00, -1.74477836e+00, -1.39813811e+00,
-1.18150376e+00],
[-1.74885626e+00, 3.37848329e-01, -1.39813811e+00,
-1.31297673e+00],
[-1.02184904e+00, 1.03205722e+00, -1.22754100e+00,
-7.87084847e-01],
[-9.00681170e-01, 1.72626612e+00, -1.05694388e+00,
-1.05003079e+00],
[-1.26418478e+00, -1.24957601e-01, -1.34127240e+00,
-1.18150376e+00],
[-9.00681170e-01, 1.72626612e+00, -1.22754100e+00,
-1.31297673e+00],
[-1.50652052e+00, 3.37848329e-01, -1.34127240e+00,
-1.31297673e+00],
[-6.58345429e-01, 1.49486315e+00, -1.28440670e+00,
-1.31297673e+00],
[-1.02184904e+00, 5.69251294e-01, -1.34127240e+00,
-1.31297673e+00],
[1.40150837e+00, 3.37848329e-01, 5.35295827e-01,
2.64698913e-01],
[6.74501145e-01, 3.37848329e-01, 4.21564419e-01,
3.96171883e-01],
[1.28034050e+00, 1.06445364e-01, 6.49027235e-01,
3.96171883e-01],
[-4.16009689e-01, -1.74477836e+00, 1.37235899e-01,
1.33225943e-01],
[7.95669016e-01, -5.87763531e-01, 4.78430123e-01,
3.96171883e-01],
[-1.73673948e-01, -5.87763531e-01, 4.21564419e-01,
1.33225943e-01],

[5.53333275e-01, 5.69251294e-01, 5.35295827e-01,
5.27644853e-01],
[-1.14301691e+00, -1.51337539e+00, -2.60824029e-01,
-2.61192967e-01],
[9.16836886e-01, -3.56360566e-01, 4.78430123e-01,
1.33225943e-01],
[-7.79513300e-01, -8.19166497e-01, 8.03701950e-02,
2.64698913e-01],
[-1.02184904e+00, -2.43898725e+00, -1.47092621e-01,
-2.61192967e-01],
[6.86617933e-02, -1.24957601e-01, 2.50967307e-01,
3.96171883e-01],
[1.89829664e-01, -1.97618132e+00, 1.37235899e-01,
-2.61192967e-01],
[3.10997534e-01, -3.56360566e-01, 5.35295827e-01,
2.64698913e-01],
[-2.94841818e-01, -3.56360566e-01, -9.02269170e-02,
1.33225943e-01],
[1.03800476e+00, 1.06445364e-01, 3.64698715e-01,
2.64698913e-01],
[-2.94841818e-01, -1.24957601e-01, 4.21564419e-01,
3.96171883e-01],
[-5.25060772e-02, -8.19166497e-01, 1.94101603e-01,
-2.61192967e-01],
[4.32165405e-01, -1.97618132e+00, 4.21564419e-01,
3.96171883e-01],
[-2.94841818e-01, -1.28197243e+00, 8.03701950e-02,
-1.29719997e-01],
[6.86617933e-02, 3.37848329e-01, 5.92161531e-01,
7.90590793e-01],
[3.10997534e-01, -5.87763531e-01, 1.37235899e-01,
1.33225943e-01],
[5.53333275e-01, -1.28197243e+00, 6.49027235e-01,
3.96171883e-01],
[3.10997534e-01, -5.87763531e-01, 5.35295827e-01,
1.75297293e-03],
[6.74501145e-01, -3.56360566e-01, 3.07833011e-01,
1.33225943e-01],
[9.16836886e-01, -1.24957601e-01, 3.64698715e-01,
2.64698913e-01],
[1.15917263e+00, -5.87763531e-01, 5.92161531e-01,
2.64698913e-01],
[1.03800476e+00, -1.24957601e-01, 7.05892939e-01,
6.59117823e-01],
[1.89829664e-01, -3.56360566e-01, 4.21564419e-01,
3.96171883e-01],
[-1.73673948e-01, -1.05056946e+00, -1.47092621e-01,
-2.61192967e-01],
[-4.16009689e-01, -1.51337539e+00, 2.35044910e-02,
-1.29719997e-01],
[-4.16009689e-01, -1.51337539e+00, -3.33612130e-02,
-2.61192967e-01],
[-5.25060772e-02, -8.19166497e-01, 8.03701950e-02,
1.75297293e-03],
[1.89829664e-01, -8.19166497e-01, 7.62758643e-01,
5.27644853e-01],

[-5.37177559e-01, -1.24957601e-01, 4.21564419e-01,
3.96171883e-01],
[1.89829664e-01, 8.00654259e-01, 4.21564419e-01,
5.27644853e-01],
[1.03800476e+00, 1.06445364e-01, 5.35295827e-01,
3.96171883e-01],
[5.53333275e-01, -1.74477836e+00, 3.64698715e-01,
1.33225943e-01],
[-2.94841818e-01, -1.24957601e-01, 1.94101603e-01,
1.33225943e-01],
[-4.16009689e-01, -1.28197243e+00, 1.37235899e-01,
1.33225943e-01],
[-4.16009689e-01, -1.05056946e+00, 3.64698715e-01,
1.75297293e-03],
[3.10997534e-01, -1.24957601e-01, 4.78430123e-01,
2.64698913e-01],
[-5.25060772e-02, -1.05056946e+00, 1.37235899e-01,
1.75297293e-03],
[-1.02184904e+00, -1.74477836e+00, -2.60824029e-01,
-2.61192967e-01],
[-2.94841818e-01, -8.19166497e-01, 2.50967307e-01,
1.33225943e-01],
[-1.73673948e-01, -1.24957601e-01, 2.50967307e-01,
1.75297293e-03],
[-1.73673948e-01, -3.56360566e-01, 2.50967307e-01,
1.33225943e-01],
[4.32165405e-01, -3.56360566e-01, 3.07833011e-01,
1.33225943e-01],
[-9.00681170e-01, -1.28197243e+00, -4.31421141e-01,
-1.29719997e-01],
[-1.73673948e-01, -5.87763531e-01, 1.94101603e-01,
1.33225943e-01],
[5.53333275e-01, 5.69251294e-01, 1.27454998e+00,
1.71090158e+00],
[-5.25060772e-02, -8.19166497e-01, 7.62758643e-01,
9.22063763e-01],
[1.52267624e+00, -1.24957601e-01, 1.21768427e+00,
1.18500970e+00],
[5.53333275e-01, -3.56360566e-01, 1.04708716e+00,
7.90590793e-01],
[7.95669016e-01, -1.24957601e-01, 1.16081857e+00,
1.31648267e+00],
[2.12851559e+00, -1.24957601e-01, 1.61574420e+00,
1.18500970e+00],
[-1.14301691e+00, -1.28197243e+00, 4.21564419e-01,
6.59117823e-01],
[1.76501198e+00, -3.56360566e-01, 1.44514709e+00,
7.90590793e-01],
[1.03800476e+00, -1.28197243e+00, 1.16081857e+00,
7.90590793e-01],
[1.64384411e+00, 1.26346019e+00, 1.33141568e+00,
1.71090158e+00],
[7.95669016e-01, 3.37848329e-01, 7.62758643e-01,
1.05353673e+00],
[6.74501145e-01, -8.19166497e-01, 8.76490051e-01,
9.22063763e-01],

[1.15917263e+00, -1.24957601e-01, 9.90221459e-01,
1.18500970e+00],
[-1.73673948e-01, -1.28197243e+00, 7.05892939e-01,
1.05353673e+00],
[-5.25060772e-02, -5.87763531e-01, 7.62758643e-01,
1.57942861e+00],
[6.74501145e-01, 3.37848329e-01, 8.76490051e-01,
1.44795564e+00],
[7.95669016e-01, -1.24957601e-01, 9.90221459e-01,
7.90590793e-01],
[2.24968346e+00, 1.72626612e+00, 1.67260991e+00,
1.31648267e+00],
[2.24968346e+00, -1.05056946e+00, 1.78634131e+00,
1.44795564e+00],
[1.89829664e-01, -1.97618132e+00, 7.05892939e-01,
3.96171883e-01],
[1.28034050e+00, 3.37848329e-01, 1.10395287e+00,
1.44795564e+00],
[-2.94841818e-01, -5.87763531e-01, 6.49027235e-01,
1.05353673e+00],
[2.24968346e+00, -5.87763531e-01, 1.67260991e+00,
1.05353673e+00],
[5.53333275e-01, -8.19166497e-01, 6.49027235e-01,
7.90590793e-01],
[1.03800476e+00, 5.69251294e-01, 1.10395287e+00,
1.18500970e+00],
[1.64384411e+00, 3.37848329e-01, 1.27454998e+00,
7.90590793e-01],
[4.32165405e-01, -5.87763531e-01, 5.92161531e-01,
7.90590793e-01],
[3.10997534e-01, -1.24957601e-01, 6.49027235e-01,
7.90590793e-01],
[6.74501145e-01, -5.87763531e-01, 1.04708716e+00,
1.18500970e+00],
[1.64384411e+00, -1.24957601e-01, 1.16081857e+00,
5.27644853e-01],
[1.88617985e+00, -5.87763531e-01, 1.33141568e+00,
9.22063763e-01],
[2.49201920e+00, 1.72626612e+00, 1.50201279e+00,
1.05353673e+00],
[6.74501145e-01, -5.87763531e-01, 1.04708716e+00,
1.31648267e+00],
[5.53333275e-01, -5.87763531e-01, 7.62758643e-01,
3.96171883e-01],
[3.10997534e-01, -1.05056946e+00, 1.04708716e+00,
2.64698913e-01],
[2.24968346e+00, -1.24957601e-01, 1.33141568e+00,
1.44795564e+00],
[5.53333275e-01, 8.00654259e-01, 1.04708716e+00,
1.57942861e+00],
[6.74501145e-01, 1.06445364e-01, 9.90221459e-01,
7.90590793e-01],
[1.89829664e-01, -1.24957601e-01, 5.92161531e-01,
7.90590793e-01],
[1.28034050e+00, 1.06445364e-01, 9.33355755e-01,
1.18500970e+00],


```
In [8]: from sklearn.cluster import KMeans
```

```
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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.
```

```
In [9]: cust_df["Clus_km"] = labels
cust_df
```

```
Out[9]:
```

	sepal_length	sepal_width	petal_length	petal_width	Clus_km
0	5.1	3.5	1.4	0.2	1
1	4.9	3.0	1.4	0.2	1
2	4.7	3.2	1.3	0.2	1
3	4.6	3.1	1.5	0.2	1
4	5.0	3.6	1.4	0.2	1
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	0
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	0

150 rows × 5 columns

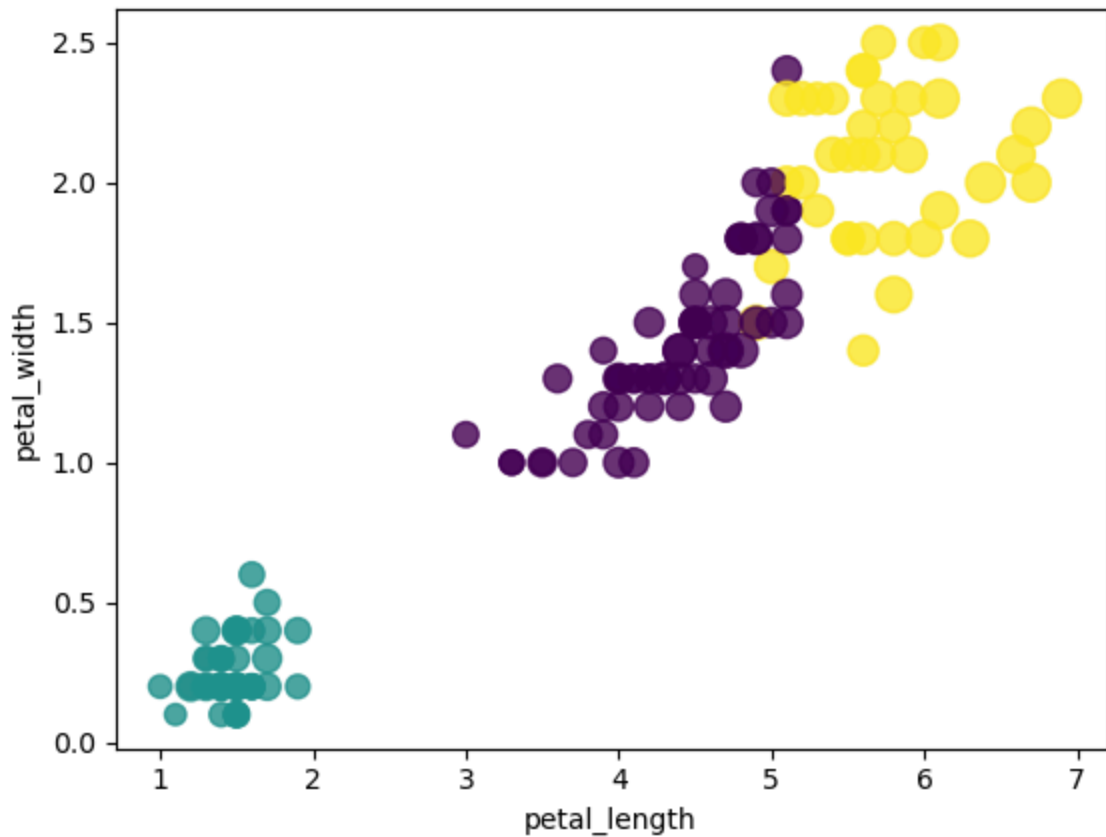
```
In [10]: cust_df.groupby('Clus_km').mean()
```

```
Out[10]:
```

	sepal_length	sepal_width	petal_length	petal_width
Clus_km				
0	5.901613	2.748387	4.393548	1.433871
1	5.006000	3.418000	1.464000	0.244000
2	6.850000	3.073684	5.742105	2.071053

```
In [12]: import matplotlib.pyplot as plt

area = np.pi * ( X[:, 0])**2
plt.scatter(X[:, 2], X[:, 3], s=area, c=labels.astype(np.float64), alpha=0.8)
plt.xlabel('petal_length', fontsize=10)
plt.ylabel('petal_width', fontsize=10)
plt.show()
```



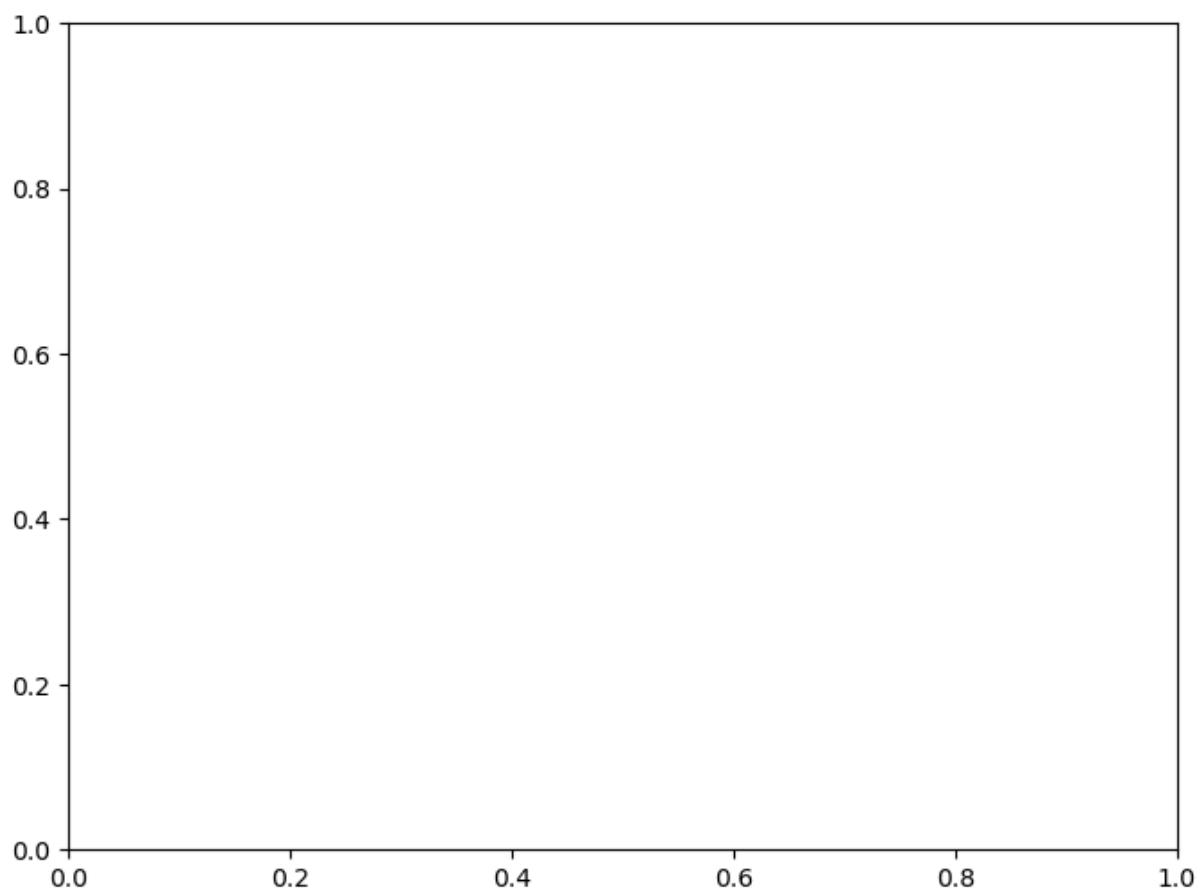
```
In [13]: from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(1, figsize=(8, 6))
plt.clf()
ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)

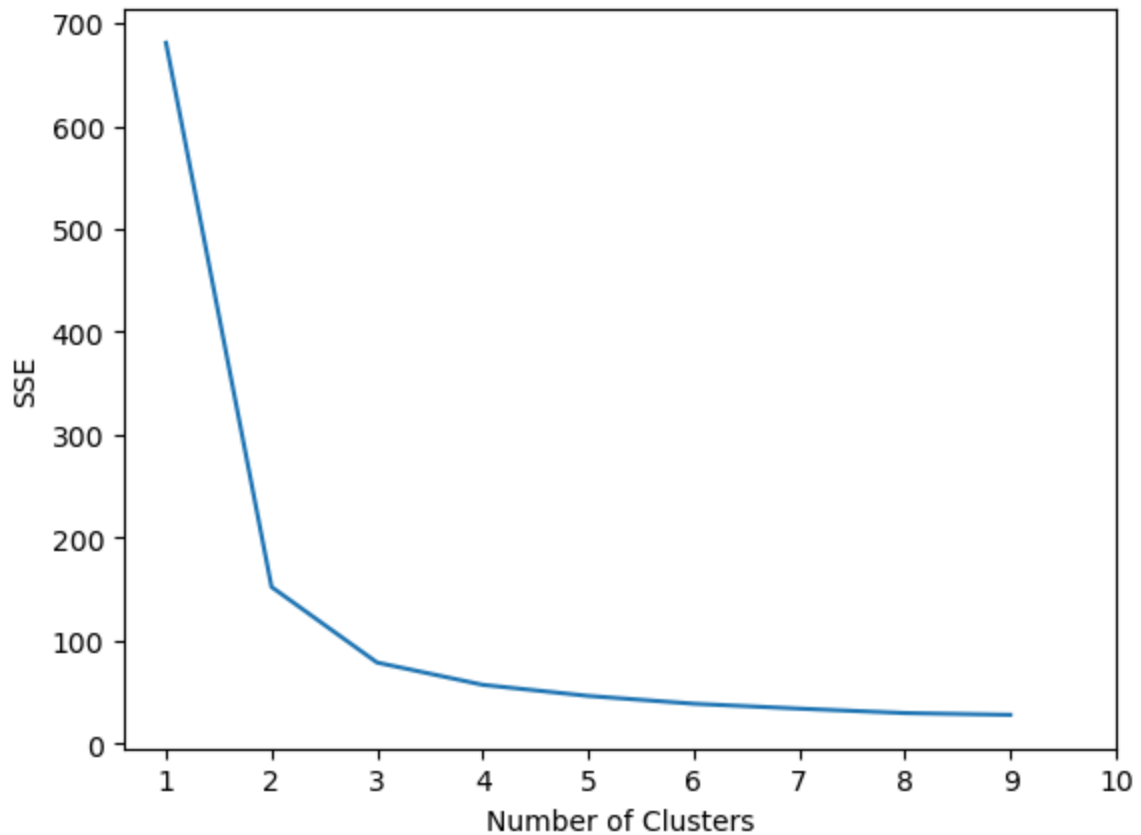
plt.cla()
ax.set_xlabel('petal_length')
ax.set_ylabel('petal_width')

ax.scatter(X[:, 2], X[:, 3], c= labels.astype(np.float64))
```

```
Out[13]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x277e86bce90>
```



```
In [14]: sse = []
         for k in range(1, 10):
             kmeans = KMeans(n_clusters=k, init= "k-means++", n_init = 12, max_iter = 300)
             kmeans.fit(X)
             sse.append(kmeans.inertia_)
```

```
In [16]: from sklearn.metrics import silhouette_score, silhouette_samples

silhouette_coefficients = []
# Notice you start at 2 clusters for silhouette coefficient
for k in range(2, 10):
    kmeans = KMeans(n_clusters=k, init= "k-means++", n_init = 12, max_iter = 300)
    kmeans.fit(X)
    score = silhouette_score(X, kmeans.labels_)
    silhouette_coefficients.append(score)

silhouette_coefficients
```

```

C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(
C:\Users\Pan Mour\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: 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(

```

```

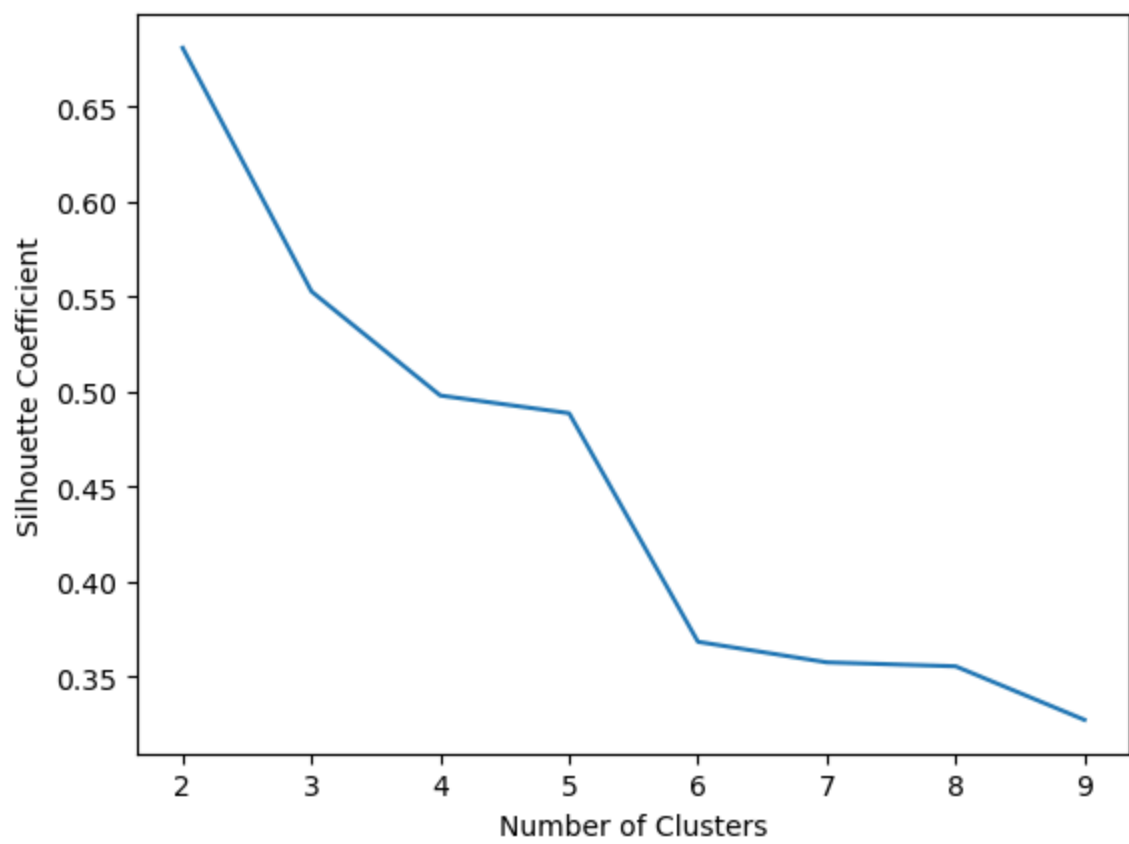
Out[16]: [0.6808136202936816,
          0.5525919445499757,
          0.4978256901095472,
          0.4885175508886279,
          0.36820569682713084,
          0.3573935435168054,
          0.3553085739523937,
          0.32712812133901203]

```

```

In [17]: plt.plot(range(2, 10), silhouette_coefficients)
plt.xticks(range(2, 10))
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Coefficient")
plt.show()

```



In []: