```python
from sklearn import datasets
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage

iris= datasets.load_iris()

print(type(iris))
```

```
<class 'sklearn.utils._bunch.Bunch'>
```

```python
iris.keys()
```

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR',
'feature_names', 'filename', 'data_module'])
```

```python
iris['target_names']
```

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```python
iris.feature_names
```

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

```python
df=pd.DataFrame(iris.data,columns=iris.feature_names)

df['target']=iris.target

df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\
n    {\n      \"column\": \"sepal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.828066127977863,\n        \"min\": 4.3,\n        \"max\": 7.9,\n
\"num_unique_values\": 35,\n        \"samples\": [\n          6.2,\n
4.5,\n          5.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.4358662849366982,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
\"samples\": [\n          2.3,\n          4.0,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length (cm)\",\n
```

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.7652982332594662,\n   \"min\": 1.0,\n   \"max\": 6.9,\n
\"num_unique_values\": 43,\n   \"samples\": [\n   6.7,\n
3.8,\n   3.7\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.7622376689603465,\n   \"min\":
0.1,\n   \"max\": 2.5,\n   \"num_unique_values\": 22,\n
\"samples\": [\n   0.2,\n   1.2,\n   1.3\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 0,\n
\"min\": 0,\n   \"max\": 2,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   0,\n   1,\n   2\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
dict_target_name={0:'setosa',1:'versicolor',2:'virginica'}

df['target_names']=df['target'].map(dict_target_name)

df.head()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 150,\n \"fields\": [\
n   {\n   \"column\": \"sepal length (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.828066127977863,\n   \"min\": 4.3,\n   \"max\": 7.9,\n
\"num_unique_values\": 35,\n   \"samples\": [\n   6.2,\n
4.5,\n   5.6\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sepal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.4358662849366982,\n   \"min\":
2.0,\n   \"max\": 4.4,\n   \"num_unique_values\": 23,\n
\"samples\": [\n   2.3,\n   4.0,\n   3.5\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"petal length (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.7652982332594662,\n   \"min\": 1.0,\n   \"max\": 6.9,\n
\"num_unique_values\": 43,\n   \"samples\": [\n   6.7,\n
3.8,\n   3.7\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.7622376689603465,\n   \"min\":
0.1,\n   \"max\": 2.5,\n   \"num_unique_values\": 22,\n
\"samples\": [\n   0.2,\n   1.2,\n   1.3\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"target\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 0,\n
\"min\": 0,\n   \"max\": 2,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   0,\n   1,\n   2\n   ],\n

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"target_names\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n   \"samples\": [\n
\"setosa\",\n   \"versicolor\",\n   \"virginica\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   target             150 non-null    int64
 5   target_names       150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
df.describe()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 8,\n \"fields\": [\n
{\n   \"column\": \"sepal length (cm)\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 51.24711349471842,\n
\"min\": 0.828066127977863,\n   \"max\": 150.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
5.843333333333334,\n   5.8,\n   150.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"sepal width (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
52.08617800869865,\n   \"min\": 0.4358662849366982,\n
\"max\": 150.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   3.0573333333333337,\n   3.0,\n
150.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal length (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 51.83521261418364,\n   \"min\":
1.0,\n   \"max\": 150.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   3.7580000000000005,\n   4.35,\n
150.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"petal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 52.63664824261751,\n   \"min\":
0.1,\n   \"max\": 150.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   1.1993333333333336,\n   1.3,\n

```
150.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"target\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 52.69404575122032,\n        \"min\": 0.0,\n        \"max\":
150.0,\n        \"num_unique_values\": 5,\n        \"samples\": [\n
1.0,\n          2.0,\n          0.8192319205190405\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

Data visualization

```
  df.columns
```

```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'target', 'target_names'],
      dtype='object')
```

```
df.shape
```

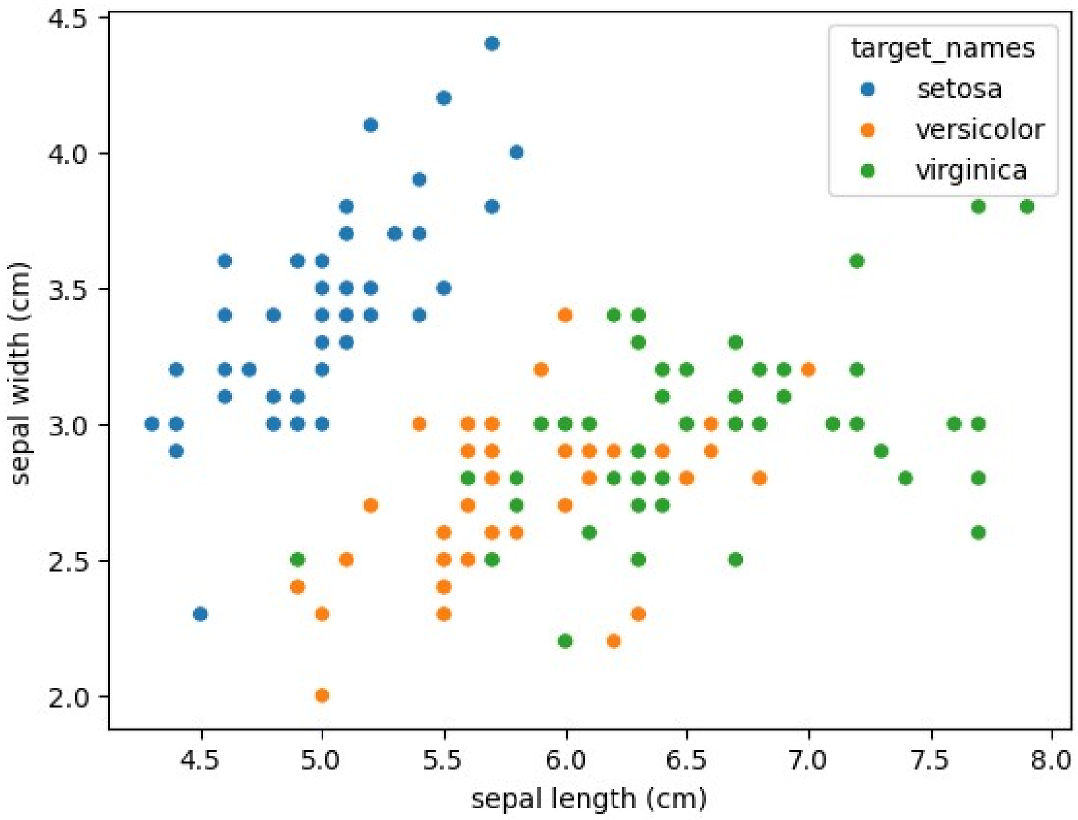
```
(150, 6)
```

```
#number of null values
df.isnull().sum()
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
target_names         0
dtype: int64
```

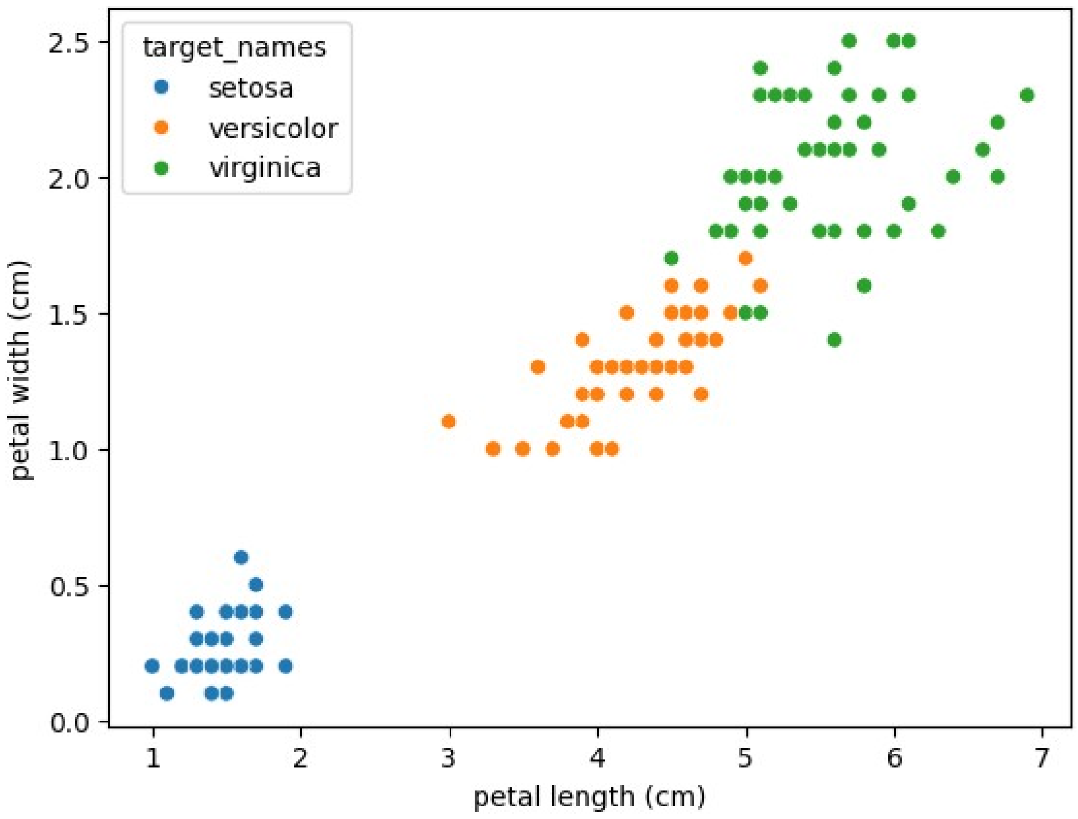
```
sns.scatterplot(data=df,x='sepal length (cm)',y='sepal width
(cm)',hue='target_names')
```

```
<Axes: xlabel='sepal length (cm)', ylabel='sepal width (cm)'>
```

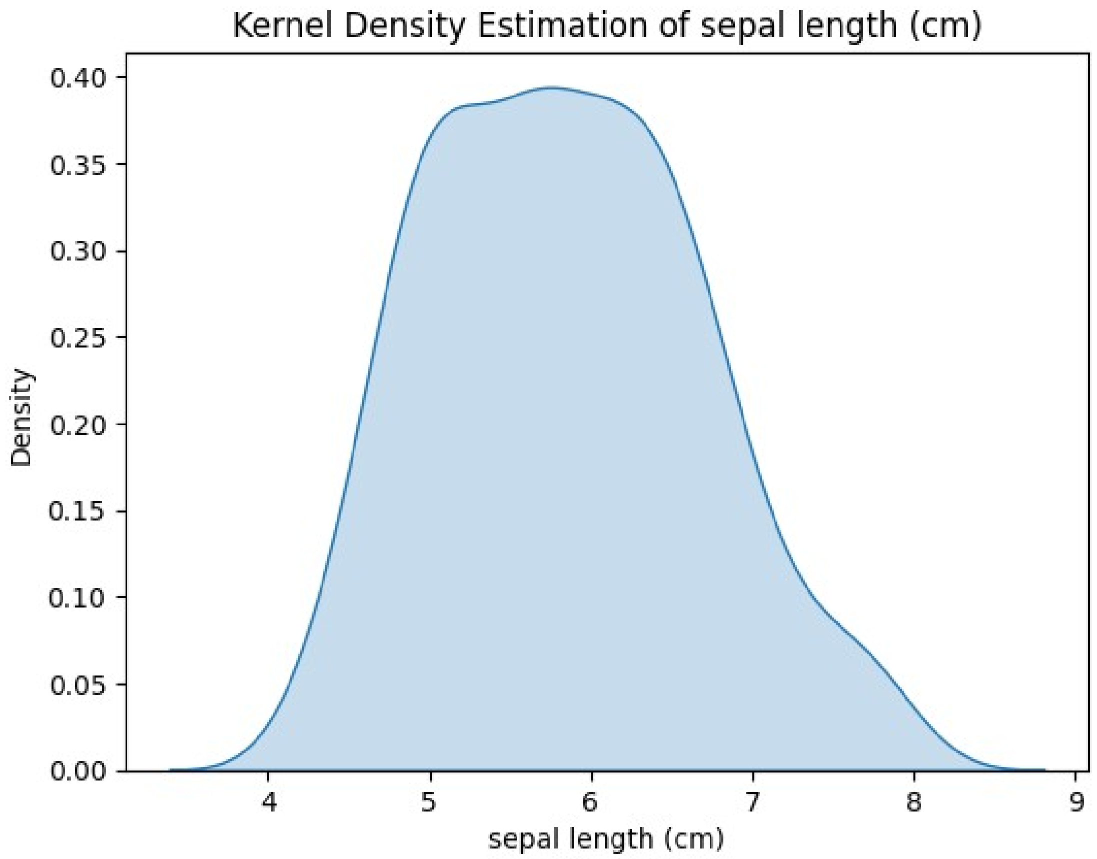


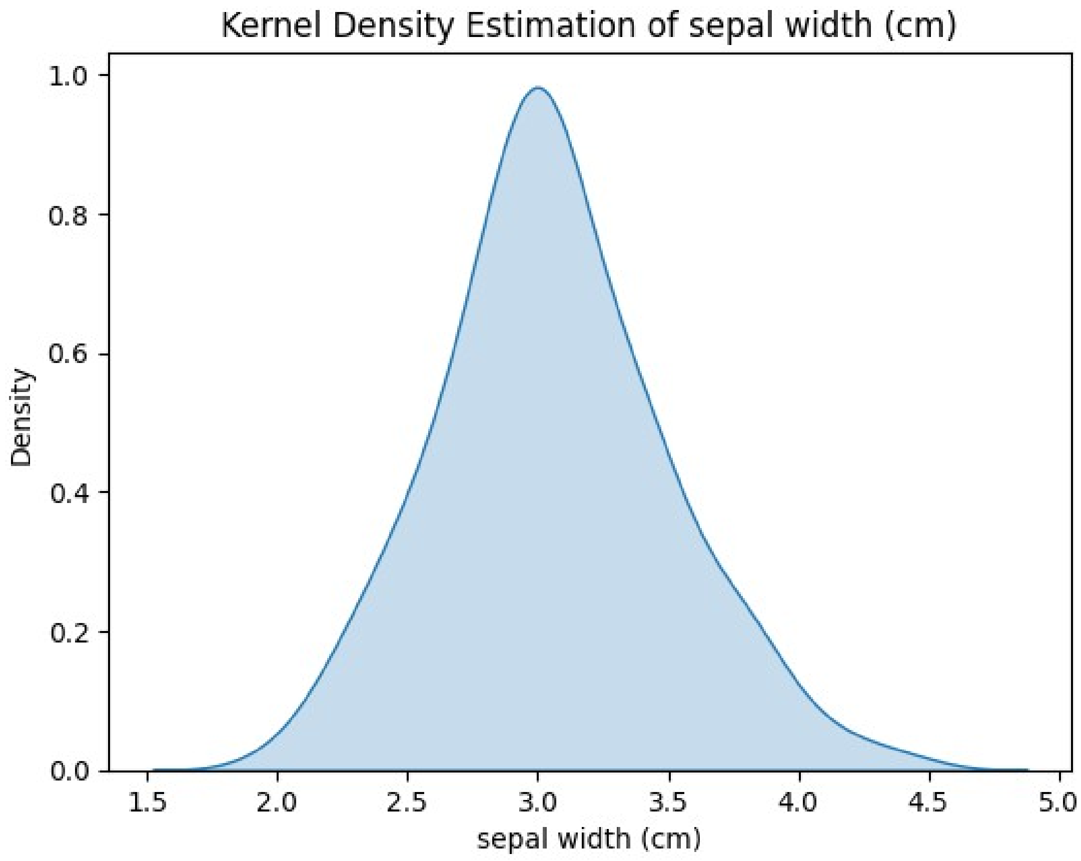
```
sns.scatterplot(data=df,x='petal length (cm)',y='petal width
(cm)',hue='target_names')
```

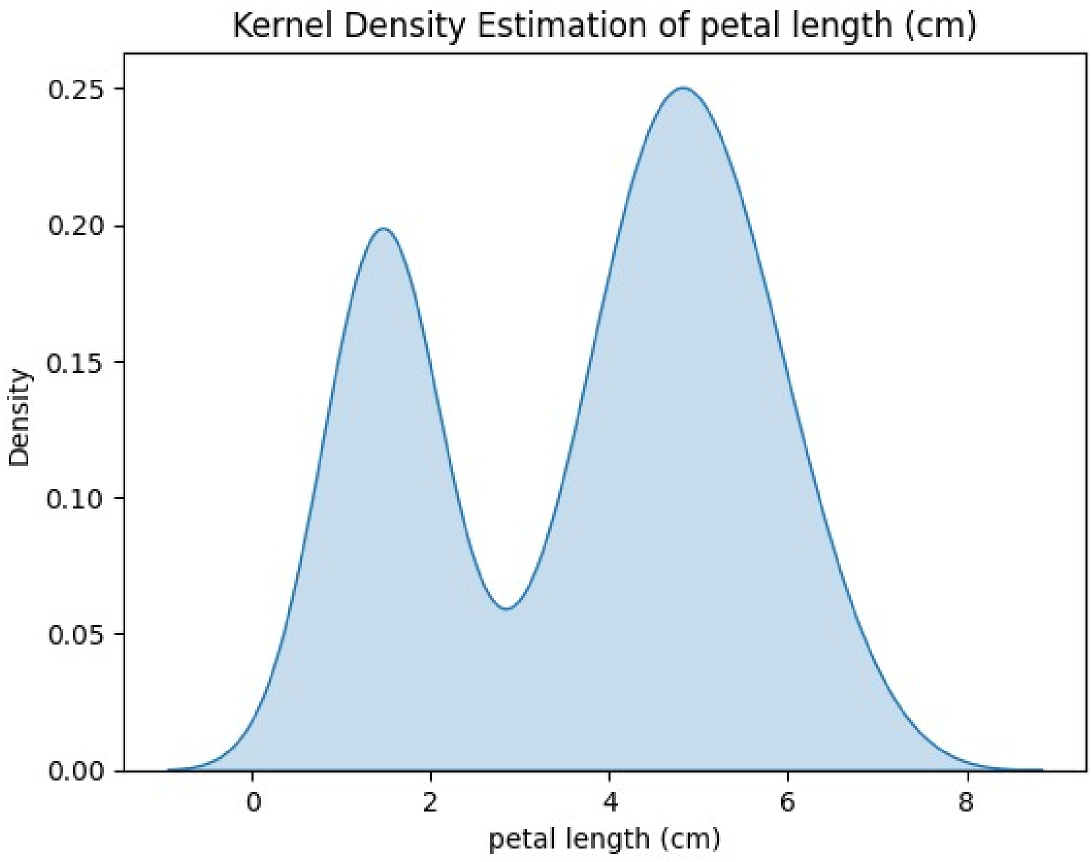
```
<Axes: xlabel='petal length (cm)', ylabel='petal width (cm)'>
```

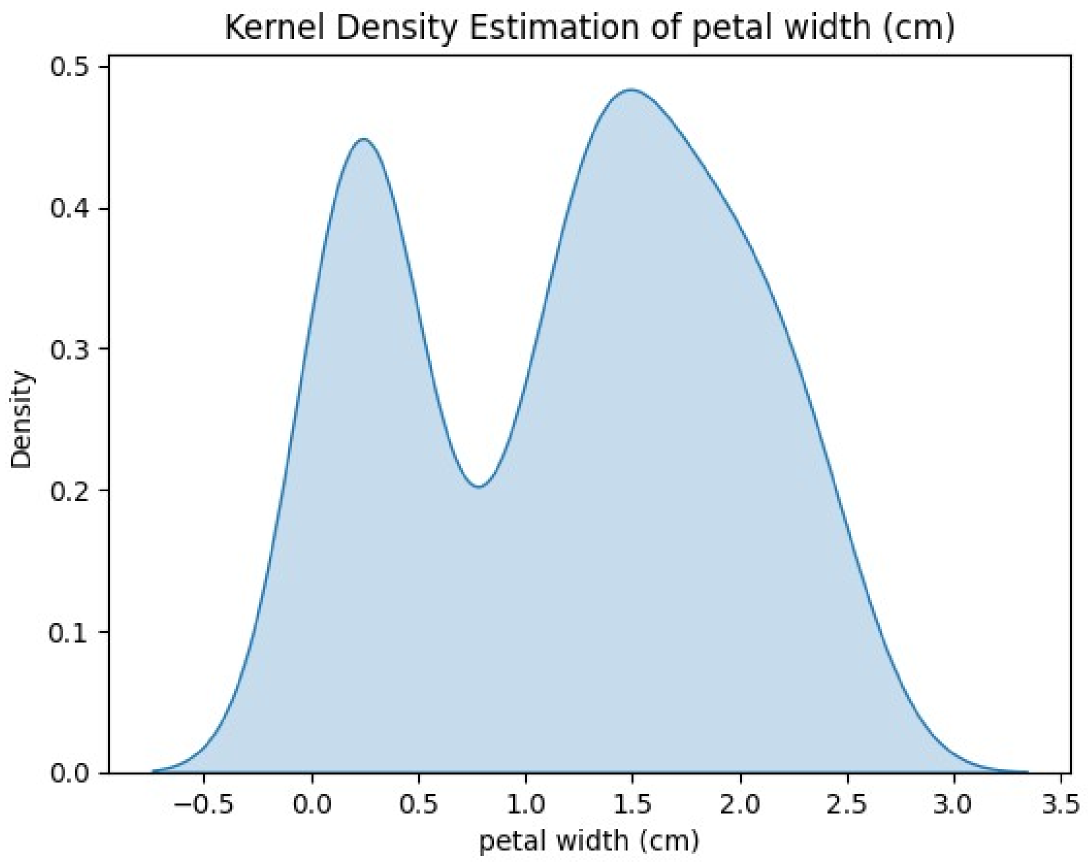


```
for feature in df.columns[:-2]:
   sns.kdeplot(data=df, x=feature, fill=True)
   plt.xlabel(feature)
   plt.ylabel('Density')
   plt.title(f'Kernel Density Estimation of {feature}')
   plt.show()
```

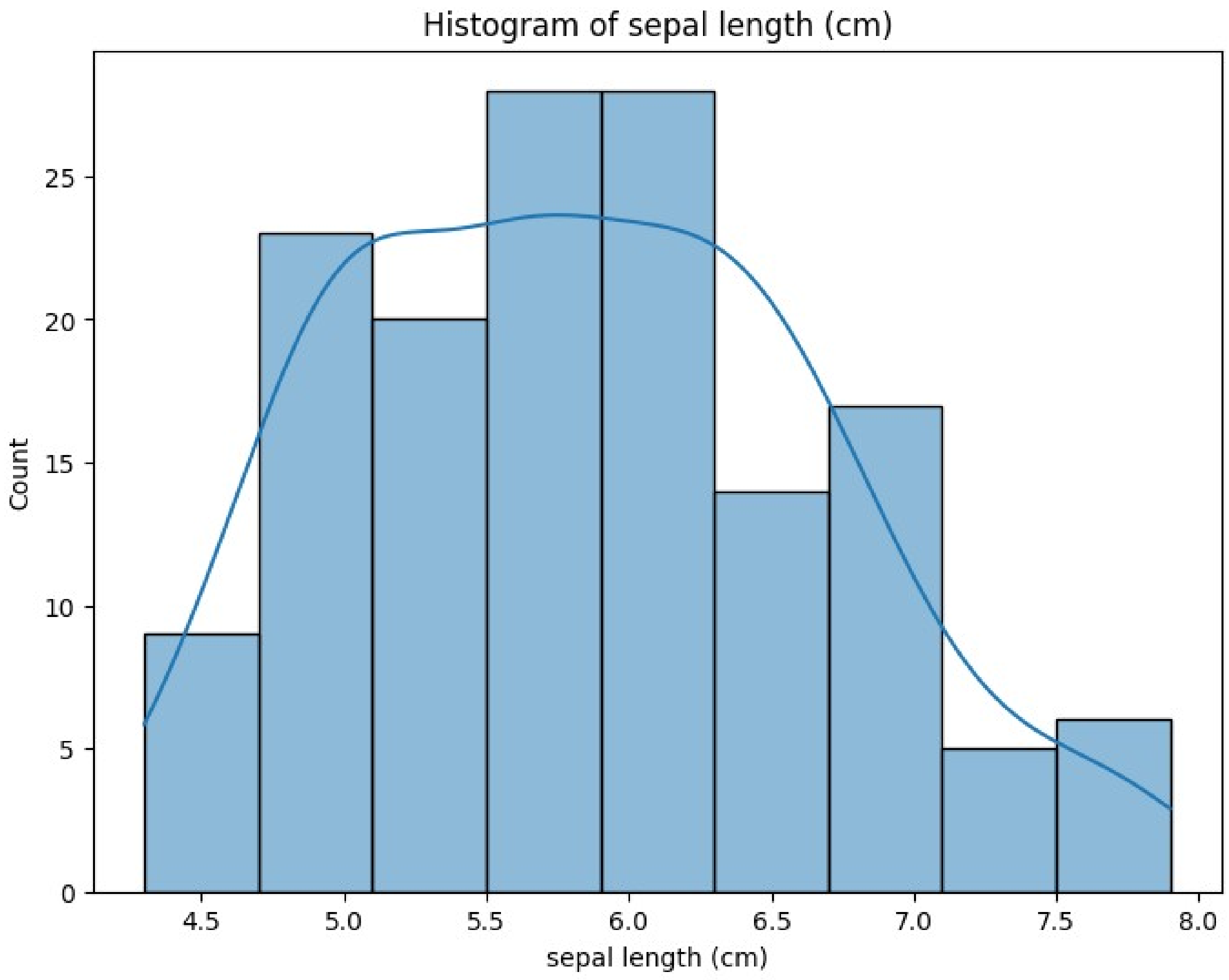

Kernel Density Estimation of sepal length (cm)
Density
sepal length (cm)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
4
5
6
7
8
9

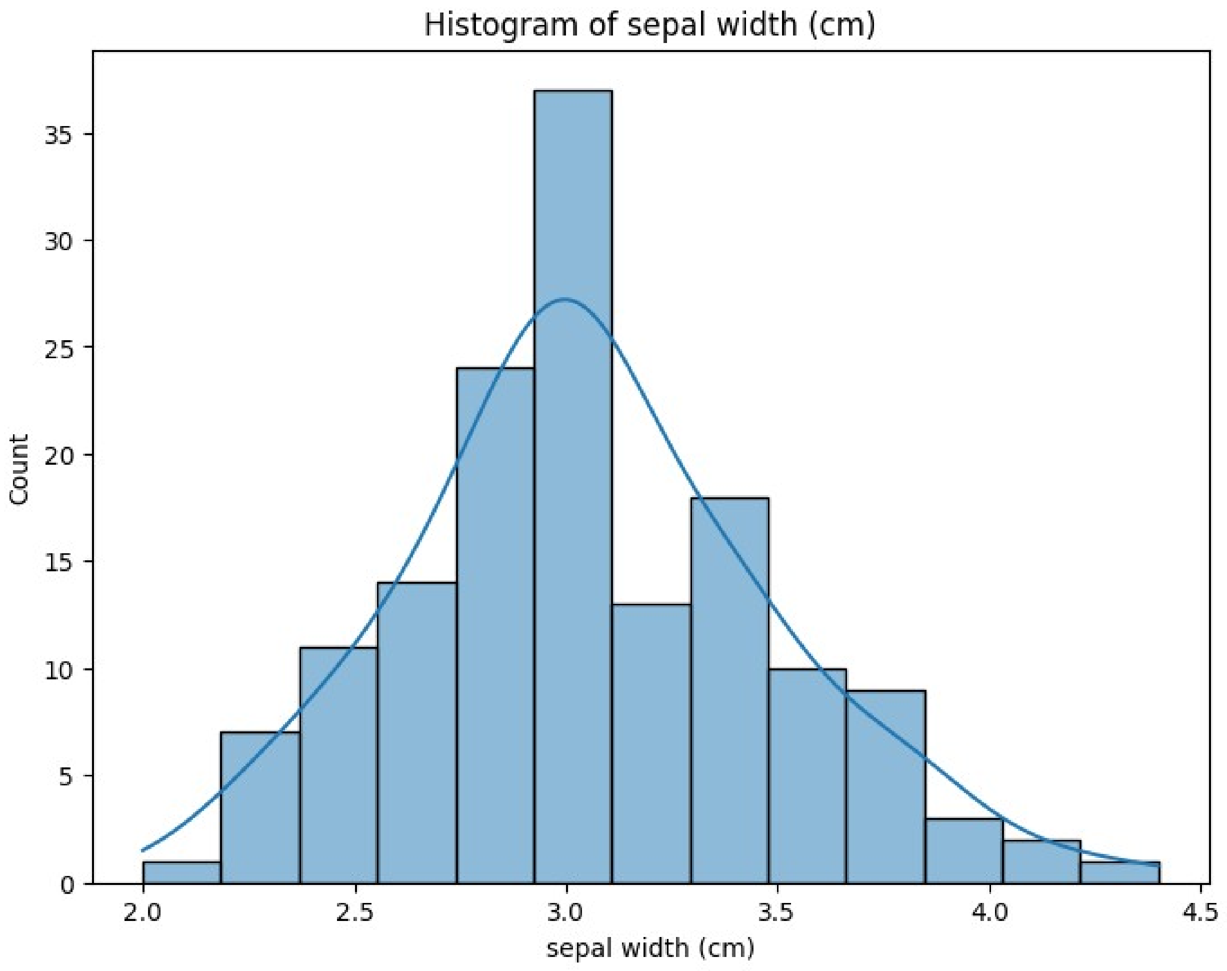

Kernel Density Estimation of sepal width (cm)
Density
sepal width (cm)
0.0
0.2
0.4
0.6
0.8
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

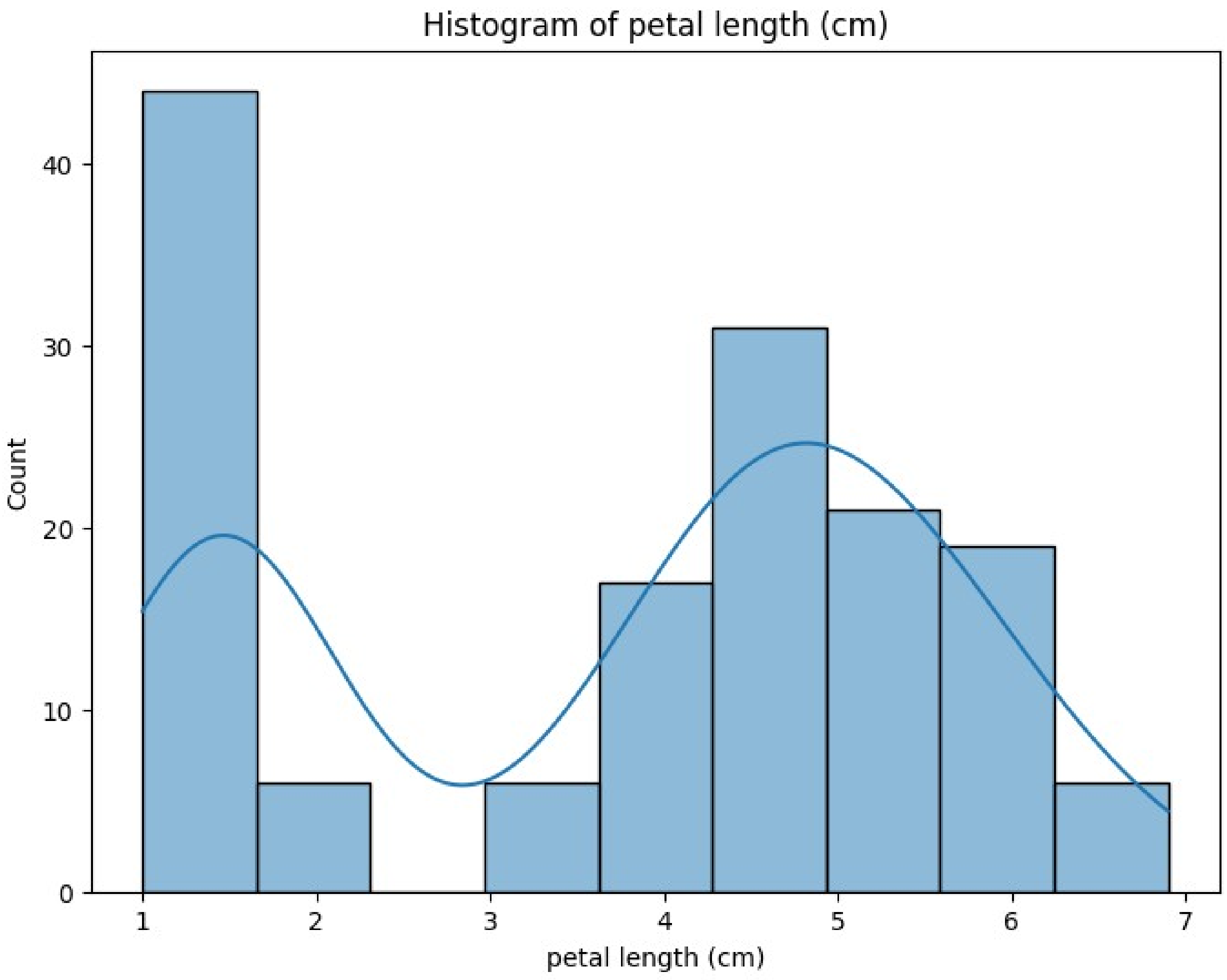

Kernel Density Estimation of petal length (cm)
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
petal length (cm)



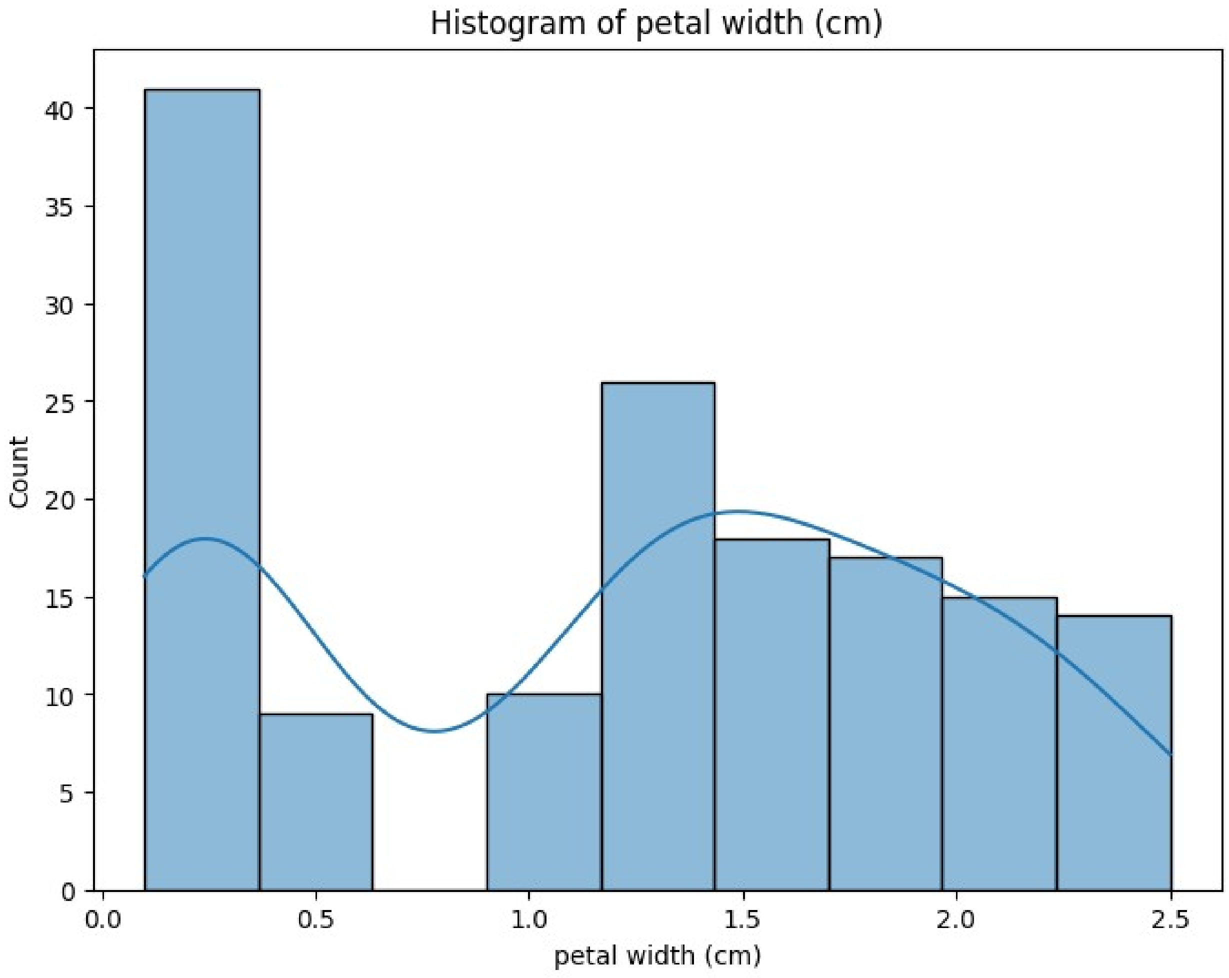
```
for feature in df.columns[:-2]:
    plt.figure(figsize=(8, 6))
    sns.histplot(data=df, x=feature, kde=True)
    plt.xlabel(feature)
    plt.ylabel('Count')
    plt.title(f'Histogram of {feature}')
    plt.show()
```


Histogram of sepal length (cm)
Count
sepal length (cm)
0
5
10
15
20
25
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

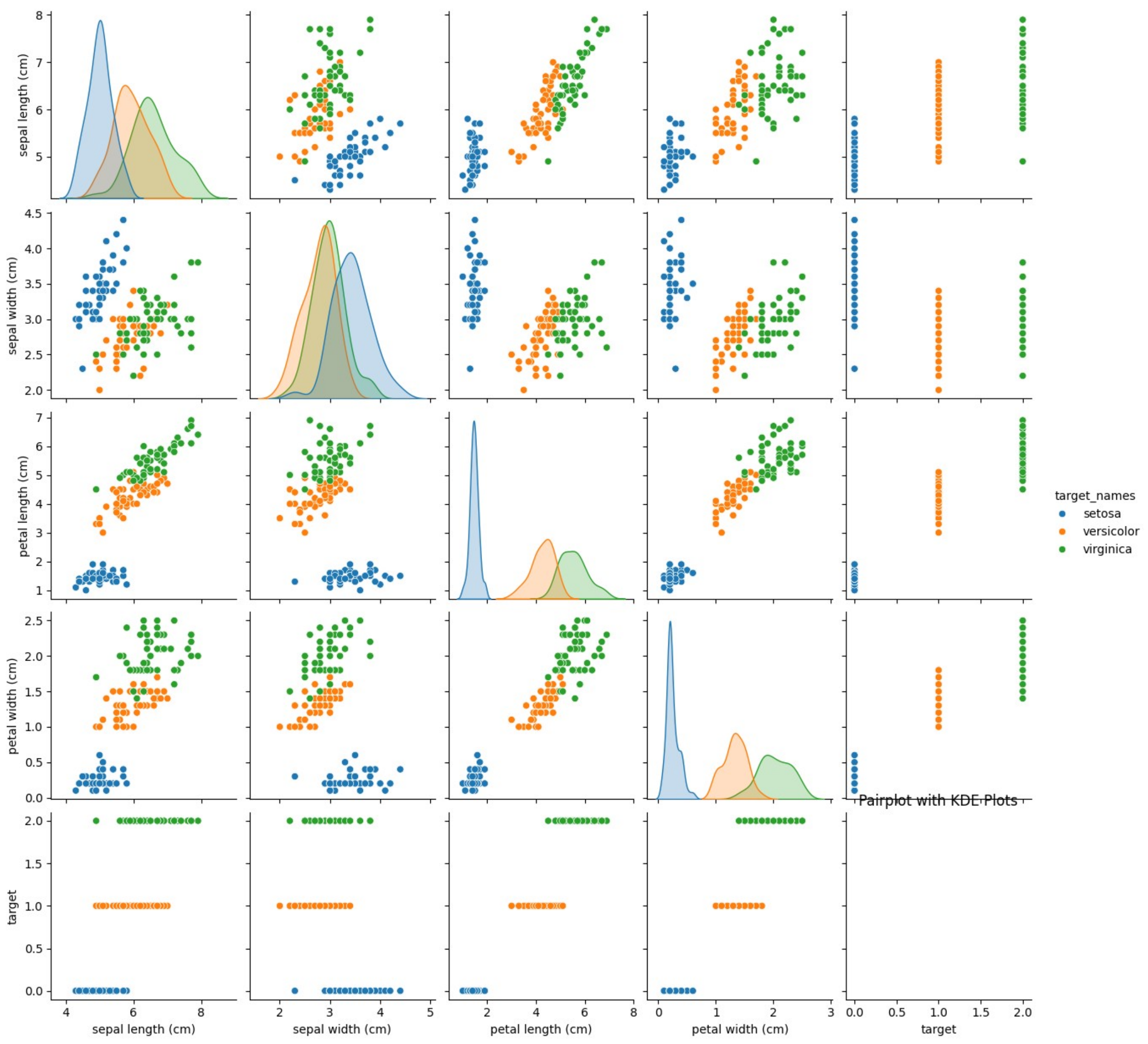

Histogram of sepal width (cm)
Count
sepal width (cm)
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5



Histogram of petal length (cm)
Count
0
10
20
30
40
1
2
3
4
5
6
7
petal length (cm)



```
sns.pairplot(data=df, hue='target_names', diag_kind='kde')
plt.title('Pairplot with KDE Plots')
plt.show()
```



Get features

```
features = df.drop(columns=["target",'target_names'],axis=1)# features

features
```

{"summary":"{\n \"name\": \"features\",\n \"rows\": 150,\n
\"fields\": [\n   {\n   \"column\": \"sepal length (cm)\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.828066127977863,\n   \"min\": 4.3,\n   \"max\": 7.9,\n
\"num_unique_values\": 35,\n   \"samples\": [\n   6.2,\n
4.5,\n   5.6\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sepal width (cm)\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.4358662849366982,\n   \"min\":

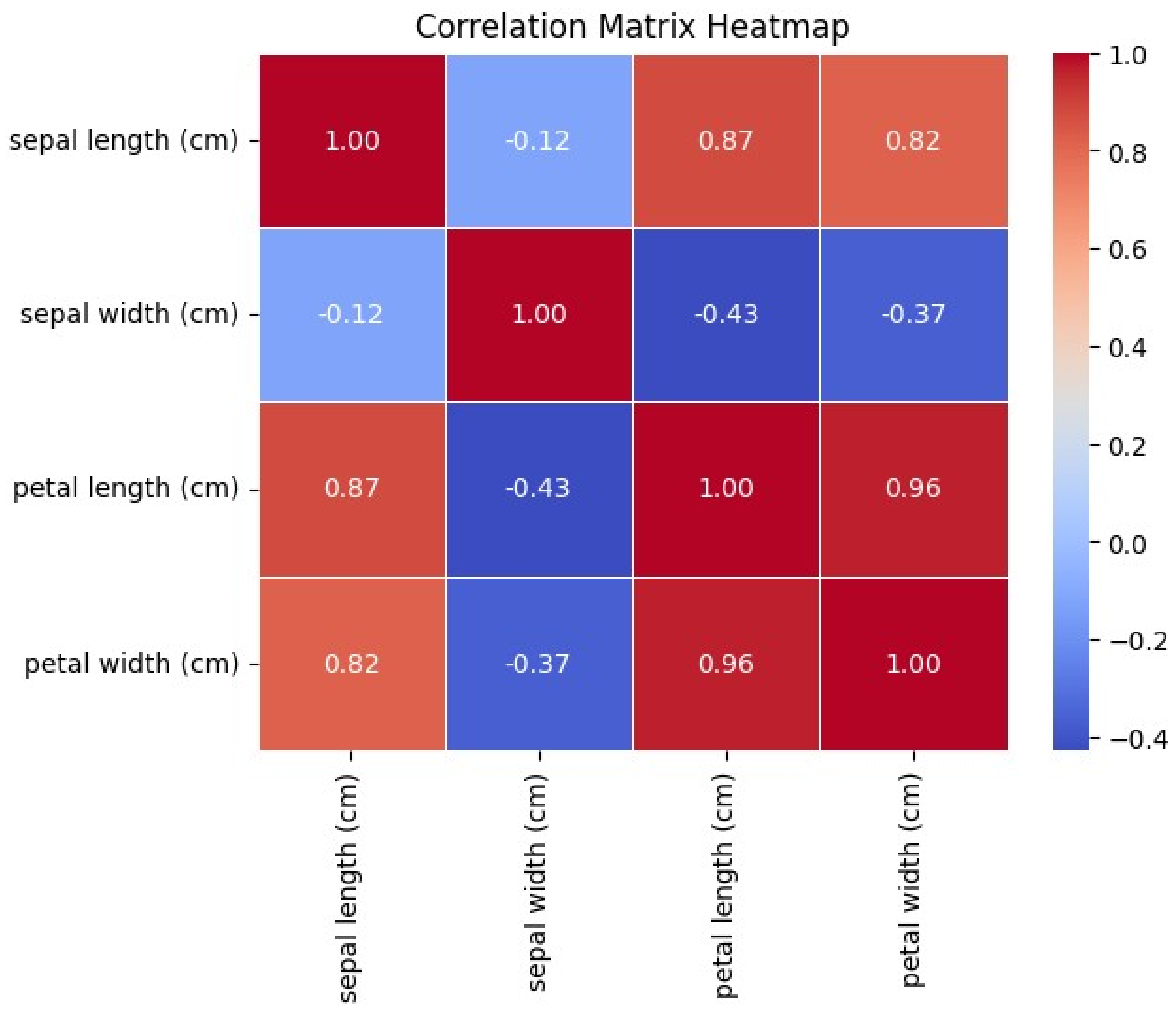
```
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
\"samples\": [\n          2.3,\n          4.0,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length (cm)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7652982332594662,\n        \"min\": 1.0,\n        \"max\": 6.9,\n
\"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n
3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal width (cm)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.7622376689603465,\n        \"min\":
0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n
\"samples\": [\n          0.2,\n          1.2,\n          1.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"features"}
```

Scaling

```
scaler = StandardScaler()

# Scale the numerical columns
scaled_features = scaler.fit_transform(features)
scaled_features
```

```
array([[-9.00681170e-01,  1.01900435e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00, -1.31979479e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.38535265e+00,  3.28414053e-01, -1.39706395e+00,
        -1.31544430e+00],
       [-1.50652052e+00,  9.82172869e-02, -1.28338910e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  1.24920112e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-5.37177559e-01,  1.93979142e+00, -1.16971425e+00,
        -1.05217993e+00],
       [-1.50652052e+00,  7.88807586e-01, -1.34022653e+00,
        -1.18381211e+00],
       [-1.02184904e+00,  7.88807586e-01, -1.28338910e+00,
        -1.31544430e+00],
       [-1.74885626e+00, -3.62176246e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00,  9.82172869e-02, -1.28338910e+00,
        -1.44707648e+00],
       [-5.37177559e-01,  1.47939788e+00, -1.28338910e+00,
        -1.31544430e+00],
       [-1.26418478e+00,  7.88807586e-01, -1.22655167e+00,
        -1.31544430e+00],
       [-1.26418478e+00, -1.31979479e-01, -1.34022653e+00,
        -1.44707648e+00],
       [-1.87002413e+00, -1.31979479e-01, -1.51073881e+00,
        -1.44707648e+00],
       [-5.25060772e-02,  2.16998818e+00, -1.45390138e+00,
        -1.31544430e+00],
       [-1.73673948e-01,  3.09077525e+00, -1.28338910e+00,
        -1.05217993e+00],
       [-5.37177559e-01,  1.93979142e+00, -1.39706395e+00,
        -1.05217993e+00],
       [-9.00681170e-01,  1.01900435e+00, -1.34022653e+00,
        -1.18381211e+00],
       [-1.73673948e-01,  1.70959465e+00, -1.16971425e+00,
        -1.18381211e+00],
       [-9.00681170e-01,  1.70959465e+00, -1.28338910e+00,
        -1.18381211e+00],
       [-5.37177559e-01,  7.88807586e-01, -1.16971425e+00,
        -1.31544430e+00],
       [-9.00681170e-01,  1.47939788e+00, -1.28338910e+00,
        -1.05217993e+00],
       [-1.50652052e+00,  1.24920112e+00, -1.56757623e+00,
        -1.31544430e+00],
       [-9.00681170e-01,  5.58610819e-01, -1.16971425e+00,
        -9.20547742e-01],
       [-1.26418478e+00,  7.88807586e-01, -1.05603939e+00,
        -1.31544430e+00],
       [-1.02184904e+00, -1.31979479e-01, -1.22655167e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  7.88807586e-01, -1.22655167e+00,
        -1.05217993e+00],
       [-7.79513300e-01,  1.01900435e+00, -1.28338910e+00,
        -1.31544430e+00],
       [-7.79513300e-01,  7.88807586e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.38535265e+00,  3.28414053e-01, -1.22655167e+00,
        -1.31544430e+00],
       [-1.26418478e+00,  9.82172869e-02, -1.22655167e+00,
        -1.31544430e+00],
       [-5.37177559e-01,  7.88807586e-01, -1.28338910e+00,
        -1.05217993e+00],
       [-7.79513300e-01,  2.40018495e+00, -1.28338910e+00,
        -1.44707648e+00],
       [-4.16009689e-01,  2.63038172e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00,  9.82172869e-02, -1.28338910e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  3.28414053e-01, -1.45390138e+00,
        -1.31544430e+00],
       [-4.16009689e-01,  1.01900435e+00, -1.39706395e+00,
         -1.31544430e+00],
        [-1.14301691e+00,  1.24920112e+00, -1.34022653e+00,
         -1.44707648e+00],
        [-1.74885626e+00, -1.31979479e-01, -1.39706395e+00,
         -1.31544430e+00],
        [-9.00681170e-01,  7.88807586e-01, -1.28338910e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  1.01900435e+00, -1.39706395e+00,
         -1.18381211e+00],
        [-1.62768839e+00, -1.74335684e+00, -1.39706395e+00,
         -1.18381211e+00],
        [-1.74885626e+00,  3.28414053e-01, -1.39706395e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  1.01900435e+00, -1.22655167e+00,
         -7.88915558e-01],
        [-9.00681170e-01,  1.70959465e+00, -1.05603939e+00,
         -1.05217993e+00],
        [-1.26418478e+00, -1.31979479e-01, -1.34022653e+00,
         -1.18381211e+00],
        [-9.00681170e-01,  1.70959465e+00, -1.22655167e+00,
         -1.31544430e+00],
        [-1.50652052e+00,  3.28414053e-01, -1.34022653e+00,
         -1.31544430e+00],
        [-6.58345429e-01,  1.47939788e+00, -1.28338910e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  5.58610819e-01, -1.34022653e+00,
         -1.31544430e+00],
        [ 1.40150837e+00,  3.28414053e-01,  5.35408562e-01,
          2.64141916e-01],
        [ 6.74501145e-01,  3.28414053e-01,  4.21733708e-01,
          3.95774101e-01],
        [ 1.28034050e+00,  9.82172869e-02,  6.49083415e-01,
          3.95774101e-01],
        [-4.16009689e-01, -1.74335684e+00,  1.37546573e-01,
          1.32509732e-01],
        [ 7.95669016e-01, -5.92373012e-01,  4.78571135e-01,
          3.95774101e-01],
        [-1.73673948e-01, -5.92373012e-01,  4.21733708e-01,
          1.32509732e-01],
        [ 5.53333275e-01,  5.58610819e-01,  5.35408562e-01,
          5.27406285e-01],
        [-1.14301691e+00, -1.51316008e+00, -2.60315415e-01,
         -2.62386821e-01],
        [ 9.16836886e-01, -3.62176246e-01,  4.78571135e-01,
          1.32509732e-01],
        [-7.79513300e-01, -8.22569778e-01,  8.07091462e-02,
          2.64141916e-01],
        [-1.02184904e+00, -2.43394714e+00, -1.46640561e-01,
         -2.62386821e-01],
        [ 6.86617933e-02, -1.31979479e-01,  2.51221427e-01,
          3.95774101e-01],
        [ 1.89829664e-01, -1.97355361e+00,  1.37546573e-01,
         -2.62386821e-01],
        [ 3.10997534e-01, -3.62176246e-01,  5.35408562e-01,
          2.64141916e-01],
        [-2.94841818e-01, -3.62176246e-01, -8.98031345e-02,
          1.32509732e-01],
        [ 1.03800476e+00,  9.82172869e-02,  3.64896281e-01,
          2.64141916e-01],
        [-2.94841818e-01, -1.31979479e-01,  4.21733708e-01,
          3.95774101e-01],
        [-5.25060772e-02, -8.22569778e-01,  1.94384000e-01,
         -2.62386821e-01],
        [ 4.32165405e-01, -1.97355361e+00,  4.21733708e-01,
          3.95774101e-01],
        [-2.94841818e-01, -1.28296331e+00,  8.07091462e-02,
         -1.30754636e-01],
        [ 6.86617933e-02,  3.28414053e-01,  5.92245988e-01,
          7.90670654e-01],
        [ 3.10997534e-01, -5.92373012e-01,  1.37546573e-01,
          1.32509732e-01],
        [ 5.53333275e-01, -1.28296331e+00,  6.49083415e-01,
          3.95774101e-01],
        [ 3.10997534e-01, -5.92373012e-01,  5.35408562e-01,
          8.77547895e-04],
        [ 6.74501145e-01, -3.62176246e-01,  3.08058854e-01,
          1.32509732e-01],
        [ 9.16836886e-01, -1.31979479e-01,  3.64896281e-01,
          2.64141916e-01],
        [ 1.15917263e+00, -5.92373012e-01,  5.92245988e-01,
          2.64141916e-01],
        [ 1.03800476e+00, -1.31979479e-01,  7.05920842e-01,
          6.59038469e-01],
        [ 1.89829664e-01, -3.62176246e-01,  4.21733708e-01,
          3.95774101e-01],
        [-1.73673948e-01, -1.05276654e+00, -1.46640561e-01,
         -2.62386821e-01],
        [-4.16009689e-01, -1.51316008e+00,  2.38717193e-02,
         -1.30754636e-01],
        [-4.16009689e-01, -1.51316008e+00, -3.29657076e-02,
         -2.62386821e-01],
        [-5.25060772e-02, -8.22569778e-01,  8.07091462e-02,
          8.77547895e-04],
        [ 1.89829664e-01, -8.22569778e-01,  7.62758269e-01,
          5.27406285e-01],
        [-5.37177559e-01, -1.31979479e-01,  4.21733708e-01,
          3.95774101e-01],
        [ 1.89829664e-01,  7.88807586e-01,  4.21733708e-01,
          5.27406285e-01],
        [ 1.03800476e+00,  9.82172869e-02,  5.35408562e-01,
          3.95774101e-01],
        [ 5.53333275e-01, -1.74335684e+00,  3.64896281e-01,
          1.32509732e-01],
        [-2.94841818e-01, -1.31979479e-01,  1.94384000e-01,
          1.32509732e-01],
        [-4.16009689e-01, -1.28296331e+00,  1.37546573e-01,
          1.32509732e-01],
        [-4.16009689e-01, -1.05276654e+00,  3.64896281e-01,
          8.77547895e-04],
        [ 3.10997534e-01, -1.31979479e-01,  4.78571135e-01,
          2.64141916e-01],
        [-5.25060772e-02, -1.05276654e+00,  1.37546573e-01,
          8.77547895e-04],
        [-1.02184904e+00, -1.74335684e+00, -2.60315415e-01,
         -2.62386821e-01],
        [-2.94841818e-01, -8.22569778e-01,  2.51221427e-01,
          1.32509732e-01],
        [-1.73673948e-01, -1.31979479e-01,  2.51221427e-01,
          8.77547895e-04],
        [-1.73673948e-01, -3.62176246e-01,  2.51221427e-01,
          1.32509732e-01],
        [ 4.32165405e-01, -3.62176246e-01,  3.08058854e-01,
          1.32509732e-01],
        [-9.00681170e-01, -1.28296331e+00, -4.30827696e-01,
         -1.30754636e-01],
        [-1.73673948e-01, -5.92373012e-01,  1.94384000e-01,
          1.32509732e-01],
        [ 5.53333275e-01,  5.58610819e-01,  1.27429511e+00,
          1.71209594e+00],
        [-5.25060772e-02, -8.22569778e-01,  7.62758269e-01,
          9.22302838e-01],
        [ 1.52267624e+00, -1.31979479e-01,  1.21745768e+00,
          1.18556721e+00],
        [ 5.53333275e-01, -3.62176246e-01,  1.04694540e+00,
          7.90670654e-01],
        [ 7.95669016e-01, -1.31979479e-01,  1.16062026e+00,
          1.31719939e+00],
        [ 2.12851559e+00, -1.31979479e-01,  1.61531967e+00,
          1.18556721e+00],
        [-1.14301691e+00, -1.28296331e+00,  4.21733708e-01,
          6.59038469e-01],
        [ 1.76501198e+00, -3.62176246e-01,  1.44480739e+00,
          7.90670654e-01],
        [ 1.03800476e+00, -1.28296331e+00,  1.16062026e+00,
          7.90670654e-01],
        [ 1.64384411e+00,  1.24920112e+00,  1.33113254e+00,
          1.71209594e+00],
       [ 7.95669016e-01,  3.28414053e-01,  7.62758269e-01,
         1.05393502e+00],
       [ 6.74501145e-01, -8.22569778e-01,  8.76433123e-01,
         9.22302838e-01],
       [ 1.15917263e+00, -1.31979479e-01,  9.90107977e-01,
         1.18556721e+00],
       [-1.73673948e-01, -1.28296331e+00,  7.05920842e-01,
         1.05393502e+00],
       [-5.25060772e-02, -5.92373012e-01,  7.62758269e-01,
         1.58046376e+00],
       [ 6.74501145e-01,  3.28414053e-01,  8.76433123e-01,
         1.44883158e+00],
       [ 7.95669016e-01, -1.31979479e-01,  9.90107977e-01,
         7.90670654e-01],
       [ 2.24968346e+00,  1.70959465e+00,  1.67215710e+00,
         1.31719939e+00],
       [ 2.24968346e+00, -1.05276654e+00,  1.78583195e+00,
         1.44883158e+00],
       [ 1.89829664e-01, -1.97355361e+00,  7.05920842e-01,
         3.95774101e-01],
       [ 1.28034050e+00,  3.28414053e-01,  1.10378283e+00,
         1.44883158e+00],
       [-2.94841818e-01, -5.92373012e-01,  6.49083415e-01,
         1.05393502e+00],
       [ 2.24968346e+00, -5.92373012e-01,  1.67215710e+00,
         1.05393502e+00],
       [ 5.53333275e-01, -8.22569778e-01,  6.49083415e-01,
         7.90670654e-01],
       [ 1.03800476e+00,  5.58610819e-01,  1.10378283e+00,
         1.18556721e+00],
       [ 1.64384411e+00,  3.28414053e-01,  1.27429511e+00,
         7.90670654e-01],
       [ 4.32165405e-01, -5.92373012e-01,  5.92245988e-01,
         7.90670654e-01],
       [ 3.10997534e-01, -1.31979479e-01,  6.49083415e-01,
         7.90670654e-01],
       [ 6.74501145e-01, -5.92373012e-01,  1.04694540e+00,
         1.18556721e+00],
       [ 1.64384411e+00, -1.31979479e-01,  1.16062026e+00,
         5.27406285e-01],
       [ 1.88617985e+00, -5.92373012e-01,  1.33113254e+00,
         9.22302838e-01],
       [ 2.49201920e+00,  1.70959465e+00,  1.50164482e+00,
         1.05393502e+00],
       [ 6.74501145e-01, -5.92373012e-01,  1.04694540e+00,
         1.31719939e+00],
       [ 5.53333275e-01, -5.92373012e-01,  7.62758269e-01,
         3.95774101e-01],
       [ 3.10997534e-01, -1.05276654e+00,  1.04694540e+00,
         2.64141916e-01],
       [ 2.24968346e+00, -1.31979479e-01,  1.33113254e+00,
         1.44883158e+00],
       [ 5.53333275e-01,  7.88807586e-01,  1.04694540e+00,
         1.58046376e+00],
       [ 6.74501145e-01,  9.82172869e-02,  9.90107977e-01,
         7.90670654e-01],
       [ 1.89829664e-01, -1.31979479e-01,  5.92245988e-01,
         7.90670654e-01],
       [ 1.28034050e+00,  9.82172869e-02,  9.33270550e-01,
         1.18556721e+00],
       [ 1.03800476e+00,  9.82172869e-02,  1.04694540e+00,
         1.58046376e+00],
       [ 1.28034050e+00,  9.82172869e-02,  7.62758269e-01,
         1.44883158e+00],
       [-5.25060772e-02, -8.22569778e-01,  7.62758269e-01,
         9.22302838e-01],
       [ 1.15917263e+00,  3.28414053e-01,  1.21745768e+00,
         1.44883158e+00],
       [ 1.03800476e+00,  5.58610819e-01,  1.10378283e+00,
         1.71209594e+00],
       [ 1.03800476e+00, -1.31979479e-01,  8.19595696e-01,
         1.44883158e+00],
       [ 5.53333275e-01, -1.28296331e+00,  7.05920842e-01,
         9.22302838e-01],
       [ 7.95669016e-01, -1.31979479e-01,  8.19595696e-01,
         1.05393502e+00],
       [ 4.32165405e-01,  7.88807586e-01,  9.33270550e-01,
         1.44883158e+00],
       [ 6.86617933e-02, -1.31979479e-01,  7.62758269e-01,
         7.90670654e-01]])
```

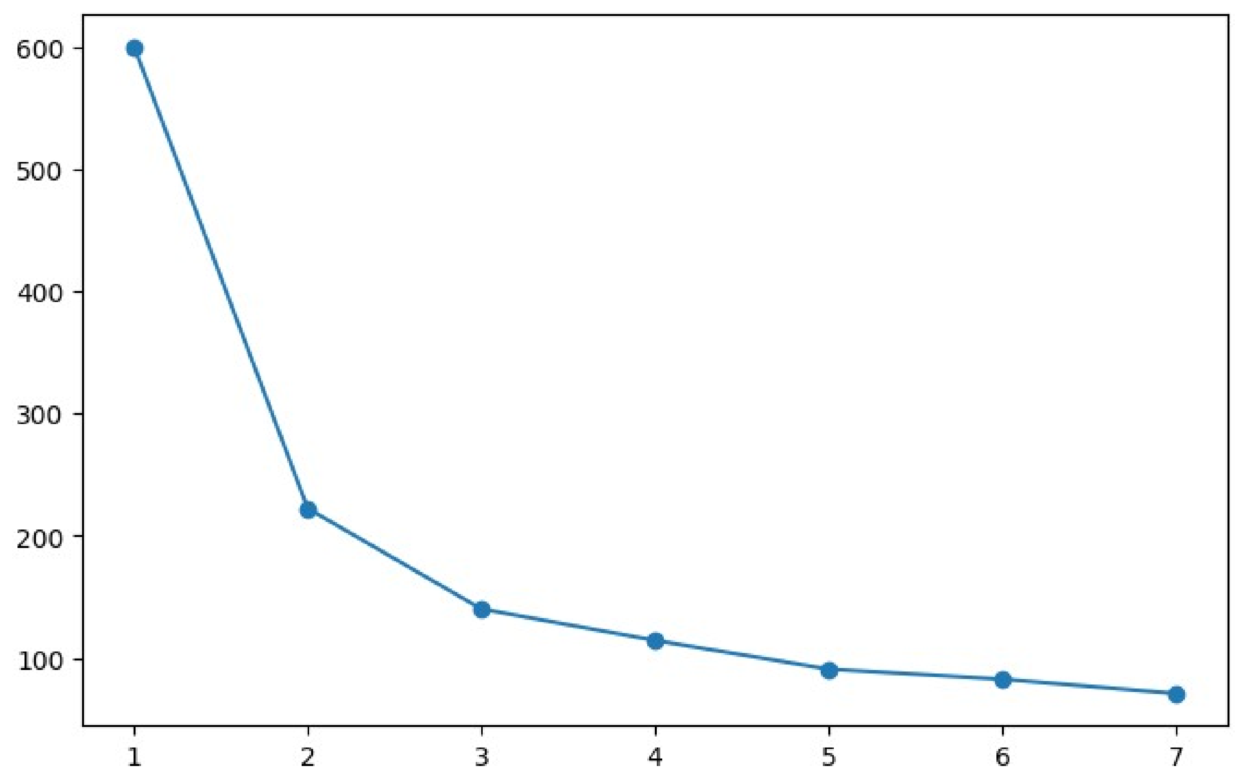
```python
correlation_matrix = features.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
linewidths=0.5, fmt='.2f')
plt.title("Correlation Matrix Heatmap")
plt.show()
```



Elbow method

Using the elbow method to determine the optimal number of clusters for k-means clustering

```
inertia=[]
k_value=range(1,8)
for k in k_value:
  kmeans=KMeans(n_clusters=k)
  kmeans.fit(scaled_features)
  inertia.append(kmeans.inertia_)

plt.figure(figsize=(8,5))
plt.plot(k_value, inertia, marker='o')
plt.grid=True
plt.show()
```



KMeans

K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

How KMeans Clustering Works KMeans clustering is an unsupervised machine learning algorithm used to group a set of data points into distinct clusters based on their similarities.

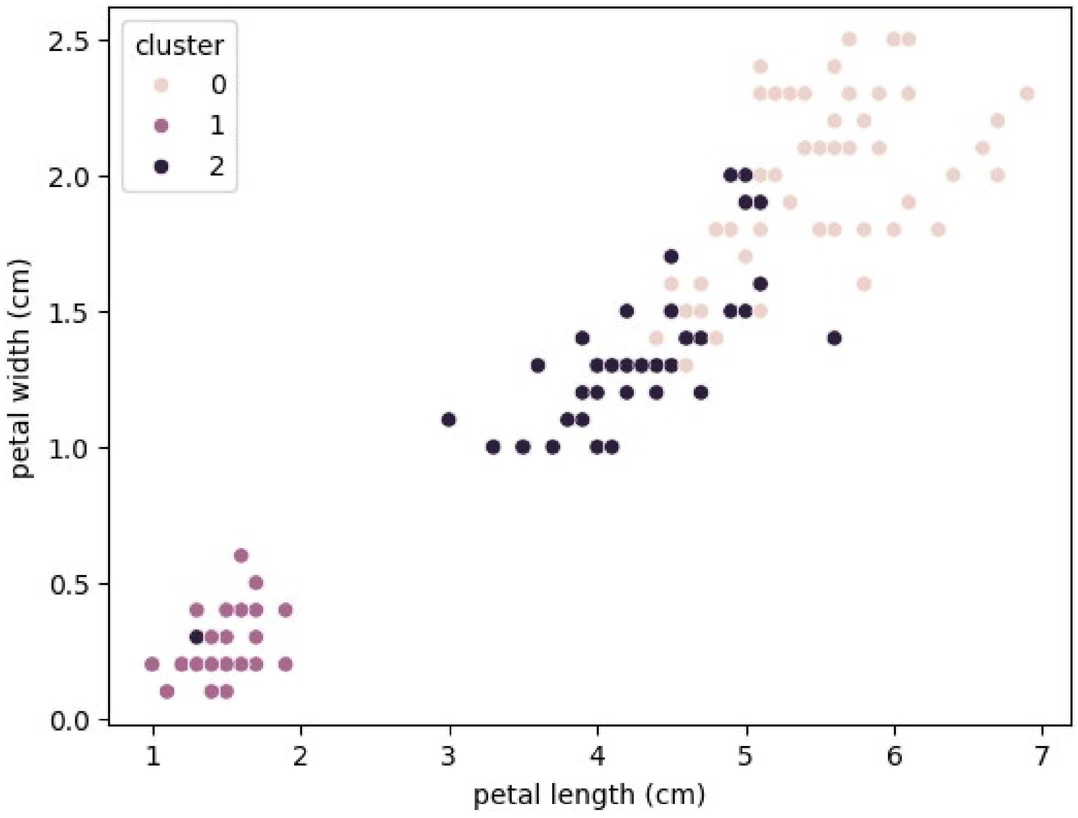
- Initialization: The algorithm starts by randomly selecting K centroids, where K is the number of clusters desired.
- Assignment Step: Each data point is assigned to the nearest centroid, forming clusters. Update Step: The centroids are recalculated as the mean of all data points in each cluster.
- Repeat: The assignment and update steps are repeated until the centroids no longer change or a maximum number of iterations is reached.

Why KMeans Clustering is Suitable for the Iris Dataset The Iris dataset contains 150 samples of flowers, categorized into three species: setosa, versicolor, and virginica, based on four features (sepal length, sepal width, petal length, and petal width). KMeans clustering is suitable for this dataset because:

- Natural Grouping: The Iris dataset contains inherent clusters (species of flowers), and KMeans aims to partition data into groups based on similarity, making it ideal for identifying these clusters.
- Feature Space: The dataset has multiple numeric features, which KMeans uses effectively to calculate distances (typically Euclidean distance) and form clusters.

- Dimensionality: With only four features, the dataset is relatively low-dimensional, making KMeans computationally efficient.
- Three Classes: Since the number of species is known to be three, we can set K=3 for the KMeans algorithm, which aligns with the actual number of clusters in the data.

```
kmeans=KMeans(n_clusters=3)
df['cluster'] = kmeans.fit_predict(scaled_features)

sns.scatterplot(data=df,x='petal length (cm)',y='petal width
(cm)',hue='cluster')

<Axes: xlabel='petal length (cm)', ylabel='petal width (cm)'>
```



```
df[['target','cluster']]
```

{"summary":"{\n \"name\": \"df[['target','cluster']]\",\n \"rows\":
150,\n \"fields\": [\n   {\n   \"column\": \"target\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0,\n   \"min\": 0,\n   \"max\": 2,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0,\n
1,\n   2\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"cluster\",\n   \"properties\": {\n   \"dtype\": \"int32\",\n
\"num_unique_values\": 3,\n   \"samples\": [\n   1,\n

```
2,\n          0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

Hierarchical Cluster

Hierarchical clustering is an unsupervised machine learning algorithm that builds a hierarchy of clusters either by:

- Agglomerative Approach (Bottom-up): Each data point starts as its own cluster. Pairs of clusters are merged step by step based on some distance metric (e.g., Euclidean distance) until all data points are grouped into a single cluster or the desired number of clusters is reached.
- Divisive Approach (Top-down): All data points start in a single cluster. The cluster is recursively split into smaller clusters until each data point forms its own cluster or a desired cluster count is reached. A dendrogram is often used to visualize the hierarchy, showing how clusters merge or split at different distance levels.

Why Hierarchical Clustering is Suitable for the Iris Dataset

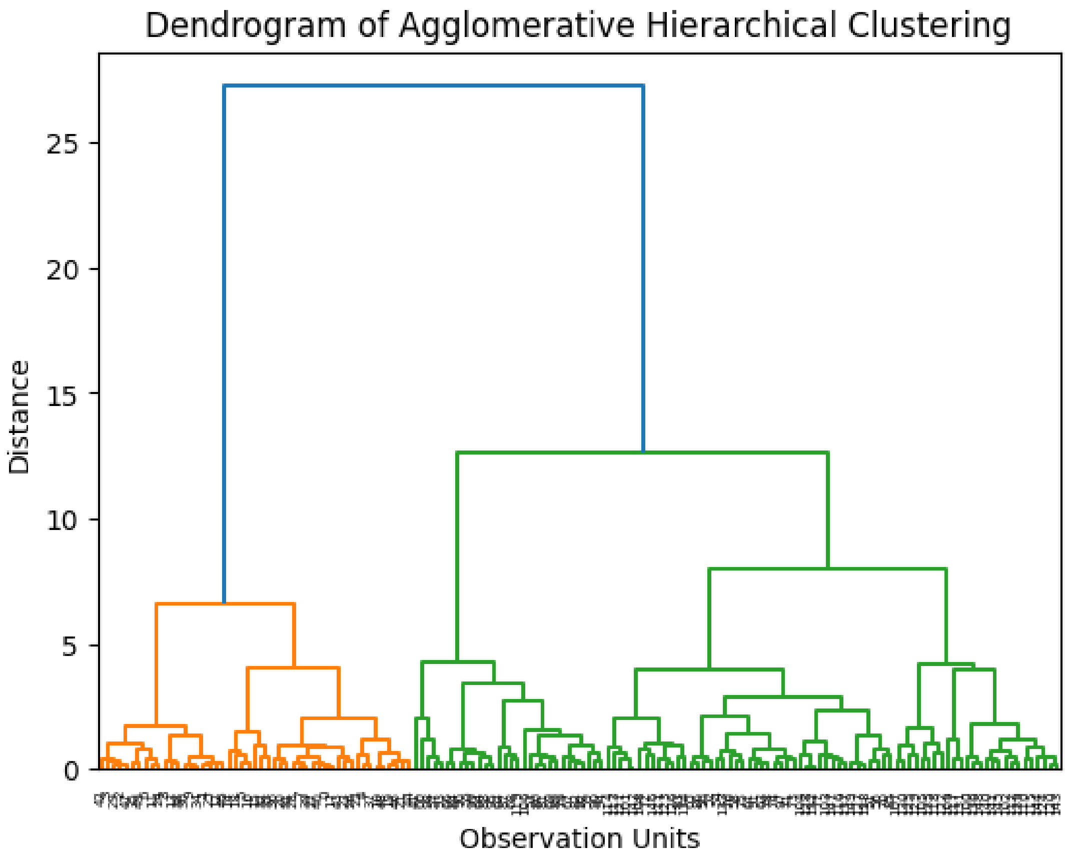
- Flexible Cluster Formation: Hierarchical clustering does not require the user to predefine the number of clusters (unlike KMeans). The Iris dataset might exhibit a nested or hierarchical structure among the species, and hierarchical clustering can reveal this.
- Visual Insight via Dendrogram: A dendrogram can give insights into how different species of flowers are related in terms of their feature similarities, showing potential subclusters within species.
- Small Dataset: The Iris dataset contains only 150 samples, making it computationally feasible to perform hierarchical clustering. Hierarchical clustering can be computationally expensive for large datasets, but this is not a concern with Iris.
- Natural Clustering: Similar to KMeans, the Iris dataset has natural clusters corresponding to the species of flowers. Hierarchical clustering can help detect these groupings without prior knowledge of the number of species.

```
scaler = StandardScaler()

# Scale the numerical columns
scaled_features = scaler.fit_transform(features)
scaled_features
```

```
array([[-9.00681170e-01,  1.01900435e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00, -1.31979479e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.38535265e+00,  3.28414053e-01, -1.39706395e+00,
        -1.31544430e+00],
       [-1.50652052e+00,  9.82172869e-02, -1.28338910e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  1.24920112e+00, -1.34022653e+00,
        -1.31544430e+00],
       [-5.37177559e-01,  1.93979142e+00, -1.16971425e+00,
        -1.05217993e+00],
       [-1.50652052e+00,  7.88807586e-01, -1.34022653e+00,
        -1.18381211e+00],
       [-1.02184904e+00,  7.88807586e-01, -1.28338910e+00,
        -1.31544430e+00],
       [-1.74885626e+00, -3.62176246e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.14301691e+00,  9.82172869e-02, -1.28338910e+00,
        -1.44707648e+00],
       [-5.37177559e-01,  1.47939788e+00, -1.28338910e+00,
        -1.31544430e+00],
       [-1.26418478e+00,  7.88807586e-01, -1.22655167e+00,
        -1.31544430e+00],
       [-1.26418478e+00, -1.31979479e-01, -1.34022653e+00,
        -1.44707648e+00],
       [-1.87002413e+00, -1.31979479e-01, -1.51073881e+00,
        -1.44707648e+00],
       [-5.25060772e-02,  2.16998818e+00, -1.45390138e+00,
        -1.31544430e+00],
       [-1.73673948e-01,  3.09077525e+00, -1.28338910e+00,
        -1.05217993e+00],
       [-5.37177559e-01,  1.93979142e+00, -1.39706395e+00,
        -1.05217993e+00],
       [-9.00681170e-01,  1.01900435e+00, -1.34022653e+00,
        -1.18381211e+00],
       [-1.73673948e-01,  1.70959465e+00, -1.16971425e+00,
        -1.18381211e+00],
       [-9.00681170e-01,  1.70959465e+00, -1.28338910e+00,
        -1.18381211e+00],
       [-5.37177559e-01,  7.88807586e-01, -1.16971425e+00,
        -1.31544430e+00],
       [-9.00681170e-01,  1.47939788e+00, -1.28338910e+00,
        -1.05217993e+00],
       [-1.50652052e+00,  1.24920112e+00, -1.56757623e+00,
        -1.31544430e+00],
       [-9.00681170e-01,  5.58610819e-01, -1.16971425e+00,
        -9.20547742e-01],
       [-1.26418478e+00,  7.88807586e-01, -1.05603939e+00,
        -1.31544430e+00],
       [-1.02184904e+00, -1.31979479e-01, -1.22655167e+00,
        -1.31544430e+00],
       [-1.02184904e+00,  7.88807586e-01, -1.22655167e+00,
        -1.05217993e+00],
       [-7.79513300e-01,  1.01900435e+00, -1.28338910e+00,
        -1.31544430e+00],
       [-7.79513300e-01,  7.88807586e-01, -1.34022653e+00,
        -1.31544430e+00],
       [-1.38535265e+00,  3.28414053e-01, -1.22655167e+00,
         -1.31544430e+00],
        [-1.26418478e+00,  9.82172869e-02, -1.22655167e+00,
         -1.31544430e+00],
        [-5.37177559e-01,  7.88807586e-01, -1.28338910e+00,
         -1.05217993e+00],
        [-7.79513300e-01,  2.40018495e+00, -1.28338910e+00,
         -1.44707648e+00],
        [-4.16009689e-01,  2.63038172e+00, -1.34022653e+00,
         -1.31544430e+00],
        [-1.14301691e+00,  9.82172869e-02, -1.28338910e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  3.28414053e-01, -1.45390138e+00,
         -1.31544430e+00],
        [-4.16009689e-01,  1.01900435e+00, -1.39706395e+00,
         -1.31544430e+00],
        [-1.14301691e+00,  1.24920112e+00, -1.34022653e+00,
         -1.44707648e+00],
        [-1.74885626e+00, -1.31979479e-01, -1.39706395e+00,
         -1.31544430e+00],
        [-9.00681170e-01,  7.88807586e-01, -1.28338910e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  1.01900435e+00, -1.39706395e+00,
         -1.18381211e+00],
        [-1.62768839e+00, -1.74335684e+00, -1.39706395e+00,
         -1.18381211e+00],
        [-1.74885626e+00,  3.28414053e-01, -1.39706395e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  1.01900435e+00, -1.22655167e+00,
         -7.88915558e-01],
        [-9.00681170e-01,  1.70959465e+00, -1.05603939e+00,
         -1.05217993e+00],
        [-1.26418478e+00, -1.31979479e-01, -1.34022653e+00,
         -1.18381211e+00],
        [-9.00681170e-01,  1.70959465e+00, -1.22655167e+00,
         -1.31544430e+00],
        [-1.50652052e+00,  3.28414053e-01, -1.34022653e+00,
         -1.31544430e+00],
        [-6.58345429e-01,  1.47939788e+00, -1.28338910e+00,
         -1.31544430e+00],
        [-1.02184904e+00,  5.58610819e-01, -1.34022653e+00,
         -1.31544430e+00],
        [ 1.40150837e+00,  3.28414053e-01,  5.35408562e-01,
          2.64141916e-01],
        [ 6.74501145e-01,  3.28414053e-01,  4.21733708e-01,
          3.95774101e-01],
        [ 1.28034050e+00,  9.82172869e-02,  6.49083415e-01,
          3.95774101e-01],
        [-4.16009689e-01, -1.74335684e+00,  1.37546573e-01,
          1.32509732e-01],
        [ 7.95669016e-01, -5.92373012e-01,  4.78571135e-01,
          3.95774101e-01],
        [-1.73673948e-01, -5.92373012e-01,  4.21733708e-01,
          1.32509732e-01],
        [ 5.53333275e-01,  5.58610819e-01,  5.35408562e-01,
          5.27406285e-01],
        [-1.14301691e+00, -1.51316008e+00, -2.60315415e-01,
         -2.62386821e-01],
        [ 9.16836886e-01, -3.62176246e-01,  4.78571135e-01,
          1.32509732e-01],
        [-7.79513300e-01, -8.22569778e-01,  8.07091462e-02,
          2.64141916e-01],
        [-1.02184904e+00, -2.43394714e+00, -1.46640561e-01,
         -2.62386821e-01],
        [ 6.86617933e-02, -1.31979479e-01,  2.51221427e-01,
          3.95774101e-01],
        [ 1.89829664e-01, -1.97355361e+00,  1.37546573e-01,
         -2.62386821e-01],
        [ 3.10997534e-01, -3.62176246e-01,  5.35408562e-01,
          2.64141916e-01],
        [-2.94841818e-01, -3.62176246e-01, -8.98031345e-02,
          1.32509732e-01],
        [ 1.03800476e+00,  9.82172869e-02,  3.64896281e-01,
          2.64141916e-01],
        [-2.94841818e-01, -1.31979479e-01,  4.21733708e-01,
          3.95774101e-01],
        [-5.25060772e-02, -8.22569778e-01,  1.94384000e-01,
         -2.62386821e-01],
        [ 4.32165405e-01, -1.97355361e+00,  4.21733708e-01,
          3.95774101e-01],
        [-2.94841818e-01, -1.28296331e+00,  8.07091462e-02,
         -1.30754636e-01],
        [ 6.86617933e-02,  3.28414053e-01,  5.92245988e-01,
          7.90670654e-01],
        [ 3.10997534e-01, -5.92373012e-01,  1.37546573e-01,
          1.32509732e-01],
        [ 5.53333275e-01, -1.28296331e+00,  6.49083415e-01,
          3.95774101e-01],
        [ 3.10997534e-01, -5.92373012e-01,  5.35408562e-01,
          8.77547895e-04],
        [ 6.74501145e-01, -3.62176246e-01,  3.08058854e-01,
          1.32509732e-01],
        [ 9.16836886e-01, -1.31979479e-01,  3.64896281e-01,
          2.64141916e-01],
        [ 1.15917263e+00, -5.92373012e-01,  5.92245988e-01,
          2.64141916e-01],
        [ 1.03800476e+00, -1.31979479e-01,  7.05920842e-01,
          6.59038469e-01],
        [ 1.89829664e-01, -3.62176246e-01,  4.21733708e-01,
          3.95774101e-01],
        [-1.73673948e-01, -1.05276654e+00, -1.46640561e-01,
         -2.62386821e-01],
        [-4.16009689e-01, -1.51316008e+00,  2.38717193e-02,
         -1.30754636e-01],
        [-4.16009689e-01, -1.51316008e+00, -3.29657076e-02,
         -2.62386821e-01],
        [-5.25060772e-02, -8.22569778e-01,  8.07091462e-02,
          8.77547895e-04],
        [ 1.89829664e-01, -8.22569778e-01,  7.62758269e-01,
          5.27406285e-01],
        [-5.37177559e-01, -1.31979479e-01,  4.21733708e-01,
          3.95774101e-01],
        [ 1.89829664e-01,  7.88807586e-01,  4.21733708e-01,
          5.27406285e-01],
        [ 1.03800476e+00,  9.82172869e-02,  5.35408562e-01,
          3.95774101e-01],
        [ 5.53333275e-01, -1.74335684e+00,  3.64896281e-01,
          1.32509732e-01],
        [-2.94841818e-01, -1.31979479e-01,  1.94384000e-01,
          1.32509732e-01],
        [-4.16009689e-01, -1.28296331e+00,  1.37546573e-01,
          1.32509732e-01],
        [-4.16009689e-01, -1.05276654e+00,  3.64896281e-01,
          8.77547895e-04],
        [ 3.10997534e-01, -1.31979479e-01,  4.78571135e-01,
          2.64141916e-01],
        [-5.25060772e-02, -1.05276654e+00,  1.37546573e-01,
          8.77547895e-04],
        [-1.02184904e+00, -1.74335684e+00, -2.60315415e-01,
         -2.62386821e-01],
        [-2.94841818e-01, -8.22569778e-01,  2.51221427e-01,
          1.32509732e-01],
        [-1.73673948e-01, -1.31979479e-01,  2.51221427e-01,
          8.77547895e-04],
        [-1.73673948e-01, -3.62176246e-01,  2.51221427e-01,
          1.32509732e-01],
        [ 4.32165405e-01, -3.62176246e-01,  3.08058854e-01,
          1.32509732e-01],
        [-9.00681170e-01, -1.28296331e+00, -4.30827696e-01,
         -1.30754636e-01],
        [-1.73673948e-01, -5.92373012e-01,  1.94384000e-01,
          1.32509732e-01],
        [ 5.53333275e-01,  5.58610819e-01,  1.27429511e+00,
          1.71209594e+00],
        [-5.25060772e-02, -8.22569778e-01,  7.62758269e-01,
          9.22302838e-01],
        [ 1.52267624e+00, -1.31979479e-01,  1.21745768e+00,
          1.18556721e+00],
       [ 5.53333275e-01, -3.62176246e-01,  1.04694540e+00,
         7.90670654e-01],
       [ 7.95669016e-01, -1.31979479e-01,  1.16062026e+00,
         1.31719939e+00],
       [ 2.12851559e+00, -1.31979479e-01,  1.61531967e+00,
         1.18556721e+00],
       [-1.14301691e+00, -1.28296331e+00,  4.21733708e-01,
         6.59038469e-01],
       [ 1.76501198e+00, -3.62176246e-01,  1.44480739e+00,
         7.90670654e-01],
       [ 1.03800476e+00, -1.28296331e+00,  1.16062026e+00,
         7.90670654e-01],
       [ 1.64384411e+00,  1.24920112e+00,  1.33113254e+00,
         1.71209594e+00],
       [ 7.95669016e-01,  3.28414053e-01,  7.62758269e-01,
         1.05393502e+00],
       [ 6.74501145e-01, -8.22569778e-01,  8.76433123e-01,
         9.22302838e-01],
       [ 1.15917263e+00, -1.31979479e-01,  9.90107977e-01,
         1.18556721e+00],
       [-1.73673948e-01, -1.28296331e+00,  7.05920842e-01,
         1.05393502e+00],
       [-5.25060772e-02, -5.92373012e-01,  7.62758269e-01,
         1.58046376e+00],
       [ 6.74501145e-01,  3.28414053e-01,  8.76433123e-01,
         1.44883158e+00],
       [ 7.95669016e-01, -1.31979479e-01,  9.90107977e-01,
         7.90670654e-01],
       [ 2.24968346e+00,  1.70959465e+00,  1.67215710e+00,
         1.31719939e+00],
       [ 2.24968346e+00, -1.05276654e+00,  1.78583195e+00,
         1.44883158e+00],
       [ 1.89829664e-01, -1.97355361e+00,  7.05920842e-01,
         3.95774101e-01],
       [ 1.28034050e+00,  3.28414053e-01,  1.10378283e+00,
         1.44883158e+00],
       [-2.94841818e-01, -5.92373012e-01,  6.49083415e-01,
         1.05393502e+00],
       [ 2.24968346e+00, -5.92373012e-01,  1.67215710e+00,
         1.05393502e+00],
       [ 5.53333275e-01, -8.22569778e-01,  6.49083415e-01,
         7.90670654e-01],
       [ 1.03800476e+00,  5.58610819e-01,  1.10378283e+00,
         1.18556721e+00],
       [ 1.64384411e+00,  3.28414053e-01,  1.27429511e+00,
         7.90670654e-01],
       [ 4.32165405e-01, -5.92373012e-01,  5.92245988e-01,
         7.90670654e-01],
       [ 3.10997534e-01, -1.31979479e-01,  6.49083415e-01,
         7.90670654e-01],
       [ 6.74501145e-01, -5.92373012e-01,  1.04694540e+00,
         1.18556721e+00],
       [ 1.64384411e+00, -1.31979479e-01,  1.16062026e+00,
         5.27406285e-01],
       [ 1.88617985e+00, -5.92373012e-01,  1.33113254e+00,
         9.22302838e-01],
       [ 2.49201920e+00,  1.70959465e+00,  1.50164482e+00,
         1.05393502e+00],
       [ 6.74501145e-01, -5.92373012e-01,  1.04694540e+00,
         1.31719939e+00],
       [ 5.53333275e-01, -5.92373012e-01,  7.62758269e-01,
         3.95774101e-01],
       [ 3.10997534e-01, -1.05276654e+00,  1.04694540e+00,
         2.64141916e-01],
       [ 2.24968346e+00, -1.31979479e-01,  1.33113254e+00,
         1.44883158e+00],
       [ 5.53333275e-01,  7.88807586e-01,  1.04694540e+00,
         1.58046376e+00],
       [ 6.74501145e-01,  9.82172869e-02,  9.90107977e-01,
         7.90670654e-01],
       [ 1.89829664e-01, -1.31979479e-01,  5.92245988e-01,
         7.90670654e-01],
       [ 1.28034050e+00,  9.82172869e-02,  9.33270550e-01,
         1.18556721e+00],
       [ 1.03800476e+00,  9.82172869e-02,  1.04694540e+00,
         1.58046376e+00],
       [ 1.28034050e+00,  9.82172869e-02,  7.62758269e-01,
         1.44883158e+00],
       [-5.25060772e-02, -8.22569778e-01,  7.62758269e-01,
         9.22302838e-01],
       [ 1.15917263e+00,  3.28414053e-01,  1.21745768e+00,
         1.44883158e+00],
       [ 1.03800476e+00,  5.58610819e-01,  1.10378283e+00,
         1.71209594e+00],
       [ 1.03800476e+00, -1.31979479e-01,  8.19595696e-01,
         1.44883158e+00],
       [ 5.53333275e-01, -1.28296331e+00,  7.05920842e-01,
         9.22302838e-01],
       [ 7.95669016e-01, -1.31979479e-01,  8.19595696e-01,
         1.05393502e+00],
       [ 4.32165405e-01,  7.88807586e-01,  9.33270550e-01,
         1.44883158e+00],
       [ 6.86617933e-02, -1.31979479e-01,  7.62758269e-01,
         7.90670654e-01]])

aggclust=AgglomerativeClustering(n_clusters=3,linkage='ward')
df['cluster']=aggclust.fit_predict(scaled_features)

df[['target','cluster']]
```

```
{"summary":"{\n  \"name\": \"df[['target','cluster']]\",\n  \"rows\":
150,\n  \"fields\": [\n    {\n      \"column\": \"target\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          0,\n
1,\n          2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"cluster\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 2,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          1,\n
2,\n          0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
# Perform hierarchical clustering
Z = linkage(scaled_features, method='ward')

# Plot the dendrogram

dendrogram(Z,leaf_rotation=90)
plt.title('Dendrogram of Agglomerative Hierarchical Clustering')
plt.xlabel("Observation Units")
plt.ylabel('Distance')
plt.show()
```



We looked at the hierarchy of clusters with the dendogram, we see that we can choose the optimum number of clusters ideally 3.