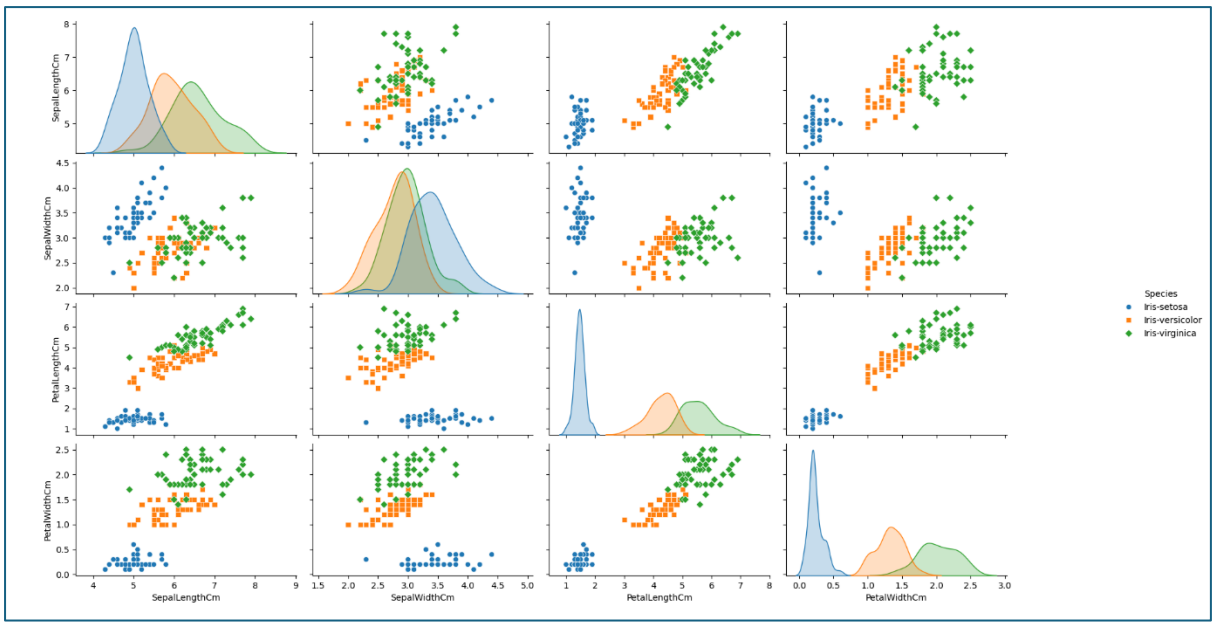
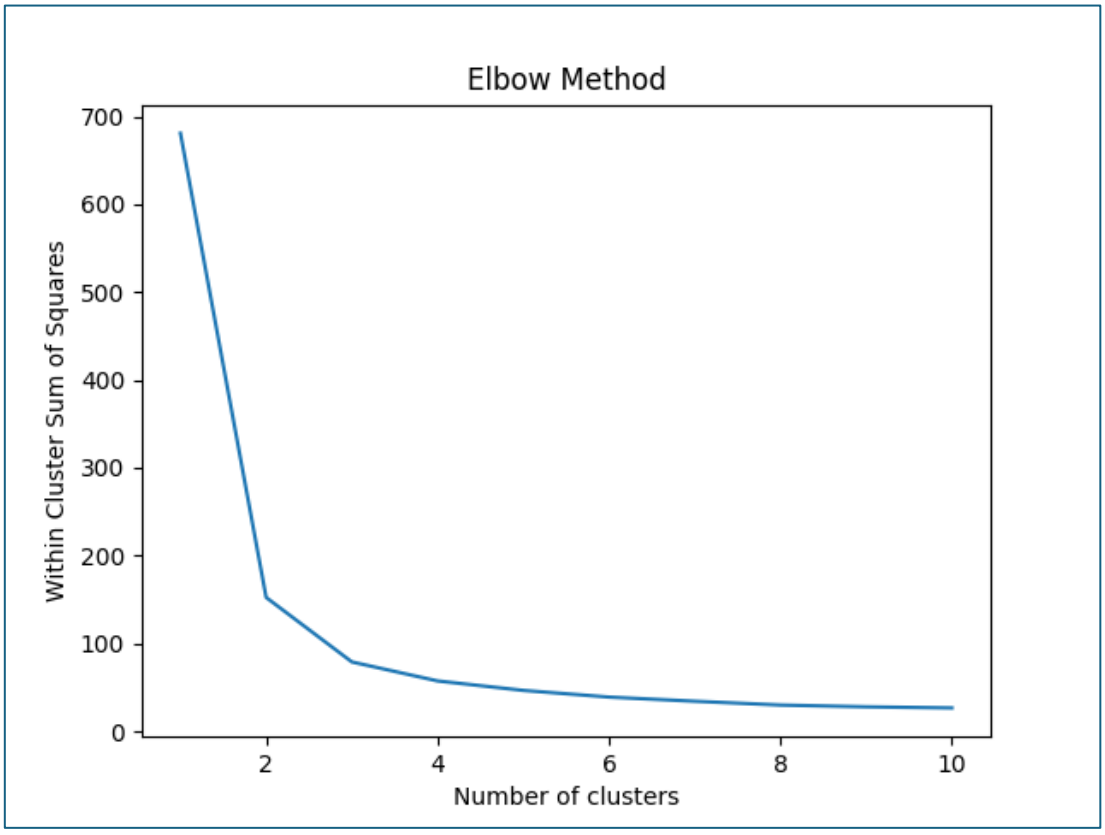
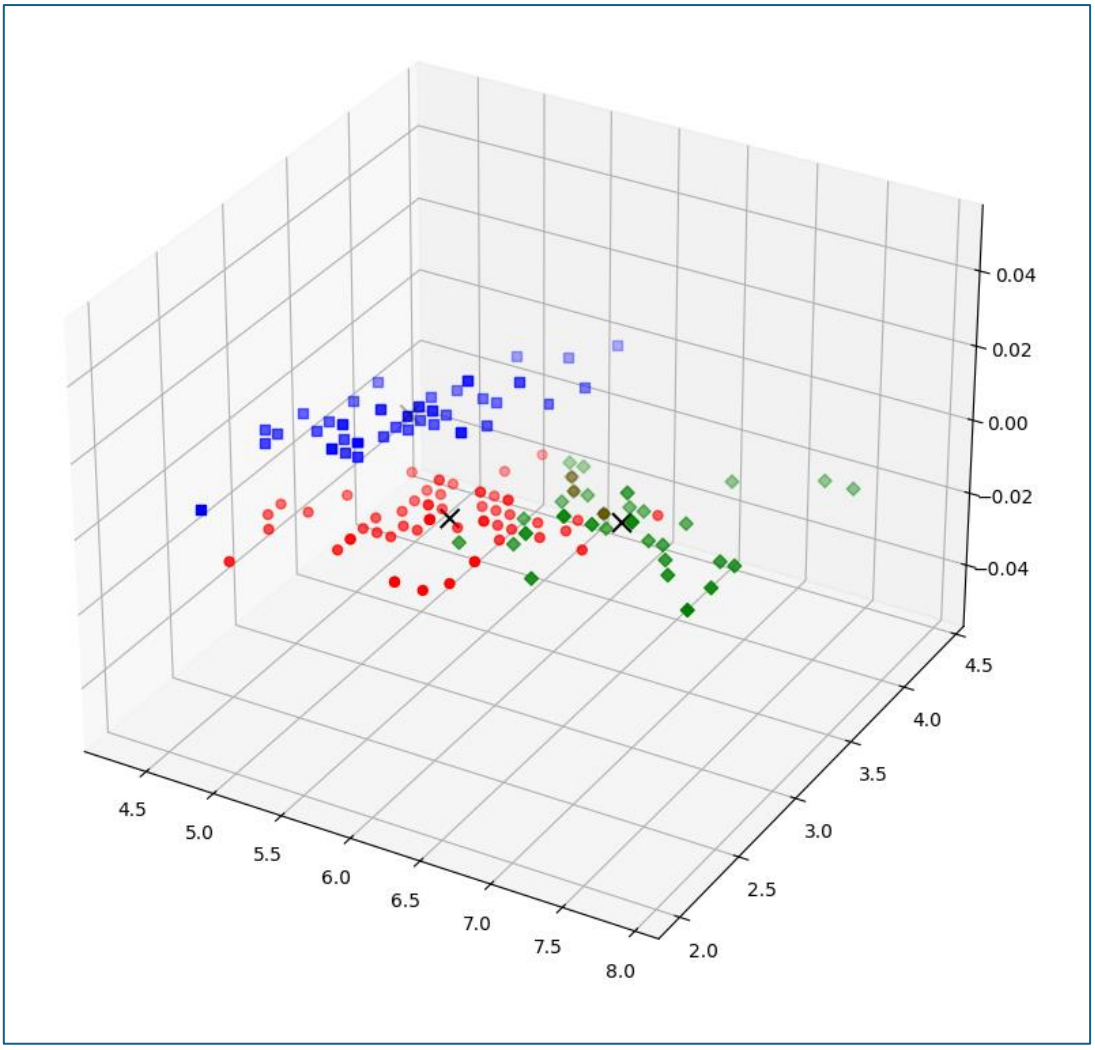

SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
Species
Iris-setosa
Iris-versicolor
Iris-virginica


Elbow Method
Within Cluster Sum of Squares
Number of clusters
0
100
200
300
400
500
600
700
2
4
6
8
10

```
import pathlib as pl
import os
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import cross_val_score
script_dir = pl.Path(__file__).parent.absolute()
os.chdir(script_dir)

df = pd.read_csv('iris.csv')

feature_columns = ['SepalLengthCm', 'SepalWidthCm',
'PetalLengthCm','PetalWidthCm']
X = df[feature_columns].values
y = df['Species'].values

encoder = LabelEncoder()
y = encoder.fit_transform(y)

sns.pairplot(df.drop("Id", axis=1), hue='Species', height=2, markers=['o', 's',
'D'])

plt.show()

wcss = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,
random_state=773)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('Within Cluster Sum of Squares')

plt.show()

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

_3means = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10,
random_state=773)
y_kmeans = _3means.fit_predict(X)

plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s=25, c='red', marker='o',
label='Iris-setosa')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s=25, c='blue', marker='s',
label='Iris-versicolour')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s=25, c='green',
marker='D', label='Iris-virginica')

plt.scatter(_3means.cluster_centers_[:, 0], _3means.cluster_centers_[:, 1],
s=100, c='black', marker='x', label='Centroids')

plt.show()
```


0.04
0.02
0.00
−0.02
−0.04
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0