

```

# Importing Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import load_iris

from sklearn.cluster import KMeans


# Data Loading and Conversion to DataFrame

D = load_iris()

X = pd.DataFrame(D.data, columns = ["sepal length", "sepal width", "petal length", "petal width"])

print(X)

X.shape

X.head()

X.tail()


# Setting a Random Seed

np.random.seed(201681)


#Initializes an empty list to store the Within-Cluster

#Sum of Squares (WSS) for different values of k

wss = []


# Looping over a range of cluster numbers to compute WSS

n = 10

for i in range(1, n+1):

    KM = KMeans(n_clusters = i, init = "k-means++", max_iter = 10000, random_state = 17)

    KM.fit(X)

    wss.append(KM.inertia_)

```

```
print(wss)
```

```
# Plotting the Elbow Method
```

```
number_clusters = range(1, n+1)
```

```
plt.figure(figsize = (6,4))
```

```
plt.plot(number_clusters, wss, color = "red", linestyle = "solid", linewidth = 2, markerfacecolor = "blue",  
markersize = 5)
```

```
plt.axvline(x = 3, color = "blue", ls = "--", linewidth = 2)
```

```
plt.xlabel("Number of clusters")
```

```
plt.ylabel("WSS")
```

```
plt.title("Elbow Method for Optimal value of k")
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Fitting KMeans with Optimal k
```

```
np.random.seed(201681)
```

```
K = KMeans(n_clusters = 3, init = "k-means++", max_iter = 10000, random_state = 17)
```

```
K.fit(X)
```

```
K.fit_predict(X)
```

```
# Accessing KMeans Attributes and Methods
```

```
K.cluster_centers_
```

```
K.labels_
```

```
K.n_features_in_
```

```
K.n_iter_
```

```
K.inertia_
```

```
K.predict([[2,5,3,1.5]])
```

```
X.insert(0, "Cluster", K.labels_, True)
```

