Lab-9

Build k-Means algorithm to cluster a set of data stored in a .CSV file.

In [ ]:
```
from google.colab import files
uploaded=files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving iris.csv to iris.csv
```

To Do: Implementation – K-Means Clustering

Write Python code to implement the following. Consider dataset files as “iris.csv”

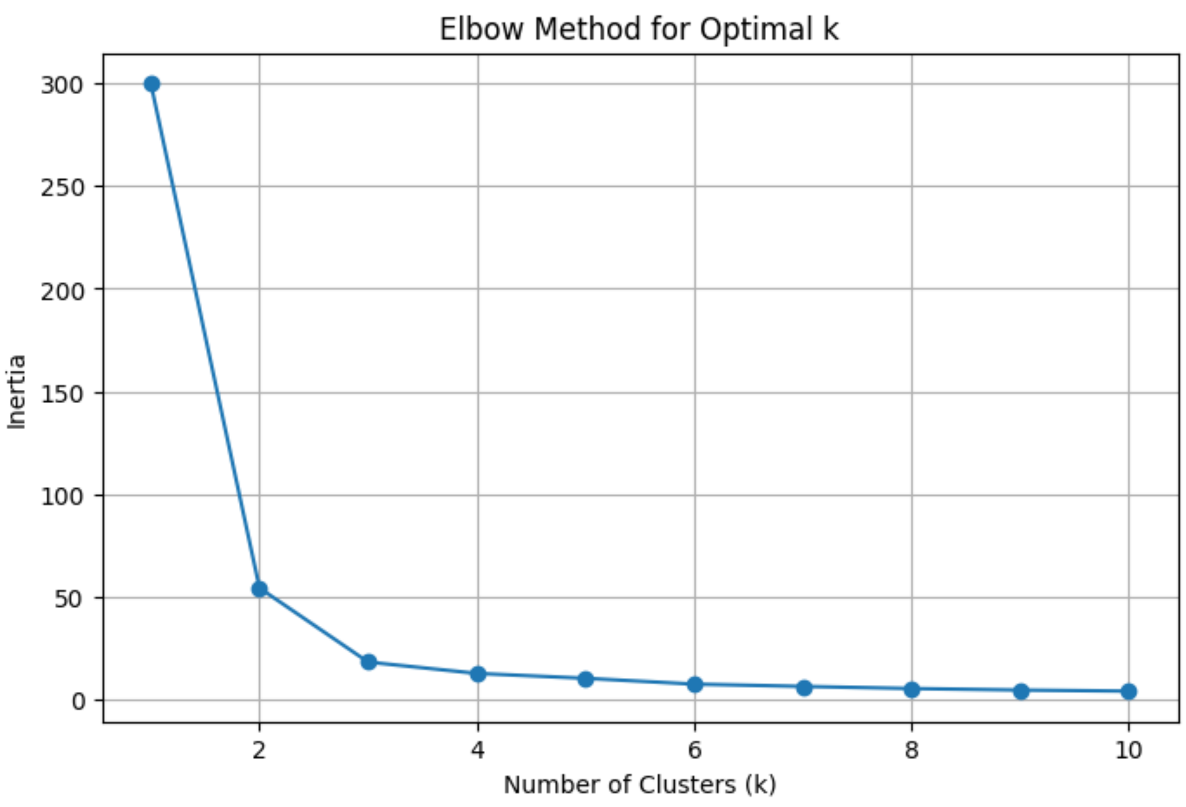
Build a K-Means Clustering algorithm to cluster IRIS flower dataset

Use iris flower dataset to form clusters of flowers using petal width and length features. Drop other two features for simplicity.

Figure out if any preprocessing such as scaling would help here

Draw elbow plot and from that figure out optimal value of k

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Load dataset
data = pd.read_csv("iris.csv")

# Select only petal length and width
X = data[["petal_length", "petal_width"]]

# Check if scaling helps (KMeans is sensitive to scale)
# Check if scaling helps (KMeans is sensitive to scale)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Elbow method: Try k from 1 to 10 and compute inertia
inertias = []
k_range = range(1, 11)
for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertias.append(kmeans.inertia_)

# Plot elbow graph
plt.figure(figsize=(8, 5))
plt.plot(k_range, inertias, marker='o')
plt.title("Elbow Method for Optimal k")
plt.xlabel("Number of Clusters (k)")
plt.ylabel("Inertia")
plt.grid(True)
plt.show()

#k=3 from the graph, k where it bends like an elbow
```


Elbow Method for Optimal k
Inertia
Number of Clusters (k)