

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

```
Dia=pd.read_csv('/content/archive (9).csv')
```

```
Dia.head()
```

```
{"summary":{"\n  \"name\": \"Dia\", \n  \"rows\": 768, \n  \"fields\": [\n    {\n      \"column\": \"Age\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 11, \n        \"min\": 21, \n        \"max\": 81, \n        \"num_unique_values\": 52, \n        \"samples\": [\n          60, \n          47, \n          72\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Pregnancies\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 3, \n        \"min\": 0, \n        \"max\": 17, \n        \"num_unique_values\": 17, \n        \"samples\": [\n          6, \n          1, \n          3\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Glucose\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 31, \n        \"min\": 0, \n        \"max\": 199, \n        \"num_unique_values\": 136, \n        \"samples\": [\n          151, \n          101, \n          112\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"BloodPressure\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 19, \n        \"min\": 0, \n        \"max\": 122, \n        \"num_unique_values\": 47, \n        \"samples\": [\n          86, \n          46, \n          85\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"SkinThickness\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 15, \n        \"min\": 0, \n        \"max\": 99, \n        \"num_unique_values\": 51, \n        \"samples\": [\n          7, \n          12, \n          48\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"Insulin\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 115, \n        \"min\": 0, \n        \"max\": 846, \n        \"num_unique_values\": 186, \n        \"samples\": [\n          52, \n          41, \n          183\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    {\n      \"column\": \"BMI\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 7.8841603203754405, \n        \"min\": 0.0, \n        \"max\": 67.1, \n        \"num_unique_values\": 248, \n        \"samples\": [\n          19.9, \n          31.0, \n          38.1\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    } \n  ] \n}
```

```

n    },\n    {\n        \"column\": \"DiabetesPedigreeFunction\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 254.431659112447, \n            \"min\": 0.1, \n            \"max\": 997.0, \n            \"num_unique_values\": 517, \n            \"samples\": [\n                1.731, \n                426.0, \n                138.0\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n        } \n    ]\n} \", \"type\": \"dataframe\", \"variable_name\": \"Dia\"}

```

```
Dia.isnull().sum()
```

```

Age          0
Pregnancies  0
Glucose      0
BloodPressure 0
SkinThickness 0
Insulin      0
BMI          0
DiabetesPedigreeFunction 0
dtype: int64

```

```
Dia.shape
```

```
(768, 8)
```

```
X=Dia.iloc[:,[2,5,6]].values
```

```
print(X)
```

```

[[148.    0.    33.6]
 [ 85.    0.    26.6]
 [183.    0.    23.3]
 ...
 [121.  112.    26.2]
 [126.    0.    30.1]
 [ 93.    0.    30.4]]

```

```
wcss=[]
```

```

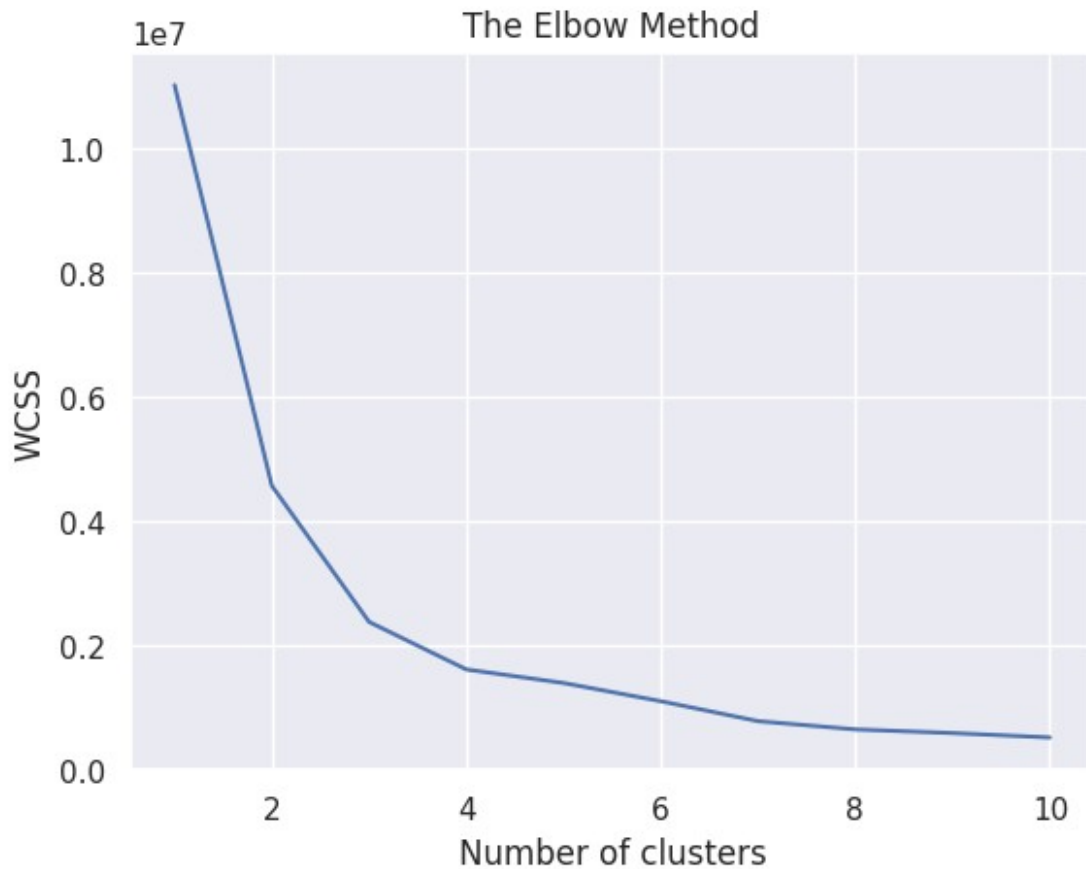
for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init='k-means++',random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

```

```

sns.set()
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

```



```
kmeans=KMeans(n_clusters=3,init='k-means++',random_state=42)
y_kmeans=kmeans.fit_predict(X)
Dia['Cluster']=y_kmeans
plt.figure(figsize=(7,7))
plt.scatter(X[y_kmeans==0,0],X[y_kmeans==0,2],s=50,c="green",label="high risk")
plt.scatter(X[y_kmeans==1,0],X[y_kmeans==1,2],s=50,c="red",label="medium risk")
plt.scatter(X[y_kmeans==2,0],X[y_kmeans==2,2],s=50,c="blue",label="less risk")
plt.scatter(kmeans.cluster_centers[:,0],kmeans.cluster_centers[:,2],s=100,c="yellow",label="Centroids")
plt.title("Glucose vs Insulin Clusters")
plt.xlabel("Glucose")
plt.ylabel("Insulin")
plt.legend()
plt.show()
```

