

USL_KNN_NB

March 2, 2025

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

```
[2]: url = r"D:\Unsupervised Machine Learning Lab (USLL)\.Lab\Iris.csv"
df = pd.read_csv(url)
df.head()
```

```
[2]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
[3]: df.drop('Id', axis=1, inplace=True)
df.head()
```

```
[3]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

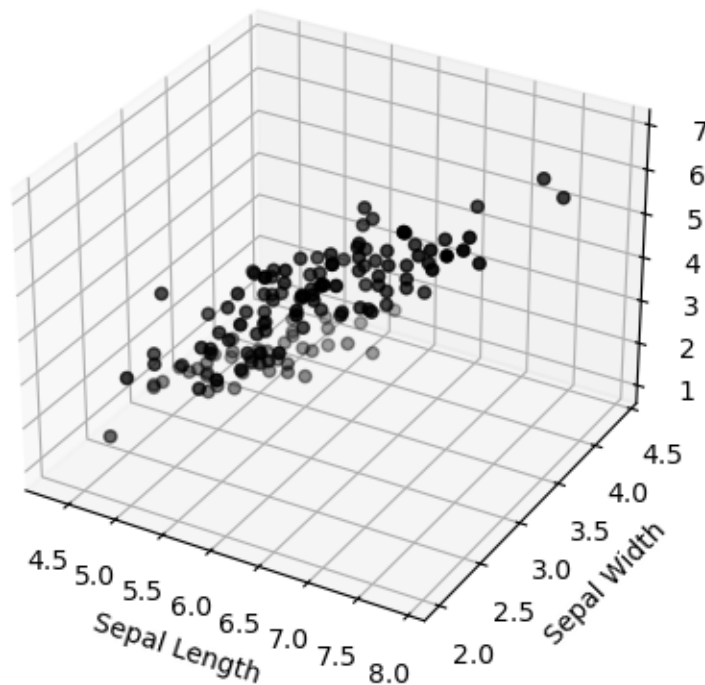
```
[4]: X = df.iloc[:, 0]
y = df.iloc[:, 1]
z = df.iloc[:, 2]
```

```
[5]: df1 = pd.DataFrame(df.iloc[:, [0, 1, 2]].values)
df1.head()
```

```
[5]:      0      1      2
      0  5.1  3.5  1.4
      1  4.9  3.0  1.4
      2  4.7  3.2  1.3
      3  4.6  3.1  1.5
      4  5.0  3.6  1.4
```

```
[6]: from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X, y, z, c='black')
ax.set_xlabel('Sepal Length')
ax.set_ylabel('Sepal Width')
ax.set_zlabel('Petal Length')
plt.show()
```



```
[ ]: kmeans = KMeans(n_clusters=3)
```

```
[8]: # Fit the KMeans model
      kmeans.fit(df1)

      # Predict the clusters
```

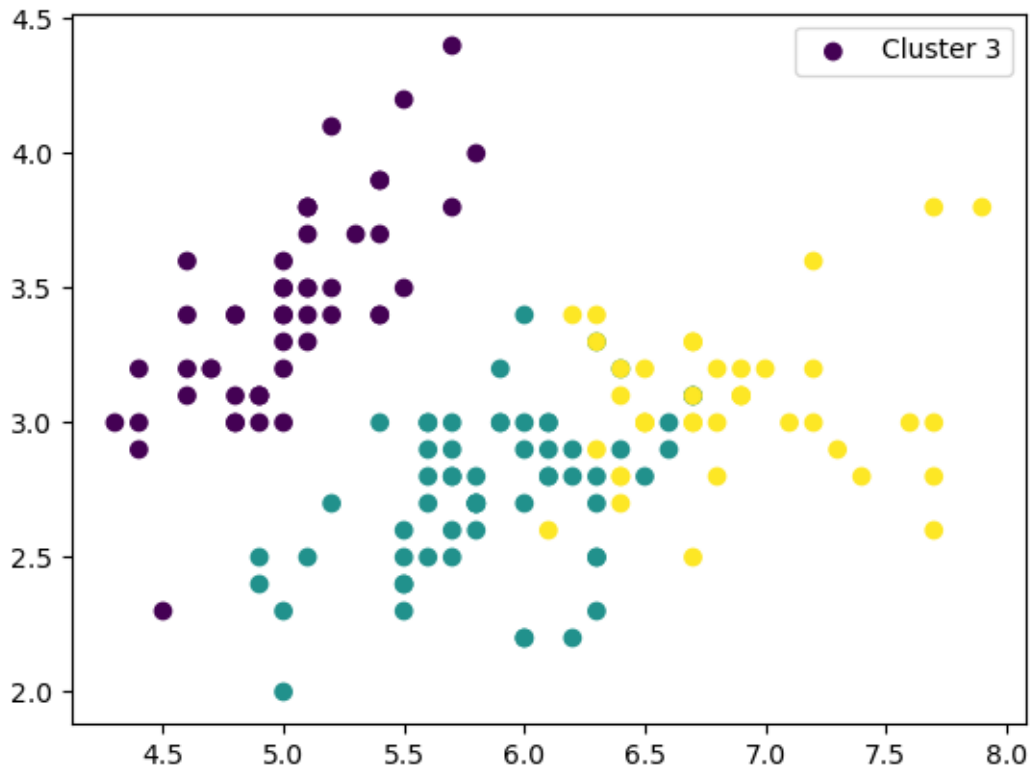
```
predictions = kmeans.predict(df1)
```

```
# Display the predictions  
print(predictions)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2  
2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2  
2 1]
```

```
c:\Users\Neil\anaconda3\Lib\site-  
packages\joblib\externals\loky\backend\context.py:136: UserWarning: Could not  
find the number of physical cores for the following reason:  
[WinError 2] The system cannot find the file specified  
Returning the number of logical cores instead. You can silence this warning by  
setting LOKY_MAX_CPU_COUNT to the number of cores you want to use.  
warnings.warn(  
File "c:\Users\Neil\anaconda3\Lib\site-  
packages\joblib\externals\loky\backend\context.py", line 257, in  
_count_physical_cores  
cpu_info = subprocess.run(  
~~~~~  
File "c:\Users\Neil\anaconda3\Lib\subprocess.py", line 548, in run  
with Popen(*popenargs, **kwargs) as process:  
~~~~~  
File "c:\Users\Neil\anaconda3\Lib\subprocess.py", line 1026, in __init__  
self._execute_child(args, executable, preexec_fn, close_fds,  
File "c:\Users\Neil\anaconda3\Lib\subprocess.py", line 1538, in _execute_child  
hp, ht, pid, tid = _winapi.CreateProcess(executable, args,  
~~~~~  
c:\Users\Neil\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1429:  
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when  
there are less chunks than available threads. You can avoid it by setting the  
environment variable OMP_NUM_THREADS=1.  
warnings.warn(  
~~~~~
```

```
[23]: plt.scatter(df1[0], df1[1], c=predictions, cmap='viridis')  
plt.legend({'Cluster 1'})  
plt.legend({'Cluster 2'})  
plt.legend({'Cluster 3'})  
plt.show()
```



```
[ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris
from mpl_toolkits.mplot3d import Axes3D

# Load the iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)

# Select three features for 3D visualization
features = ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)']
data = df[features]

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(data.iloc[:, 0], data.iloc[:, 1], data.iloc[:, 2], cmap='viridis',
           marker='o', s=50)
ax.set_xlabel(features[0])
ax.set_ylabel(features[1])
```

```

ax.set_zlabel(features[2])
ax.set_title('K-Means Clustering on Iris Dataset (3D)')
ax.legend()

# Perform K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(data)
labels = kmeans.labels_
centroids = kmeans.cluster_centers_

# Plot the clusters in 3D
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(data.iloc[:, 0], data.iloc[:, 1], data.iloc[:, 2], c=labels,
           cmap='viridis', marker='o', s=50)

# Plot centroids
ax.scatter(centroids[:, 0], centroids[:, 1], centroids[:, 2], c='red',
           marker='X', s=200, label='Centroids')

# Labels and title
ax.set_xlabel(features[0])
ax.set_ylabel(features[1])
ax.set_zlabel(features[2])
ax.set_title('K-Means Clustering on Iris Dataset (3D)')
ax.legend()

plt.show()

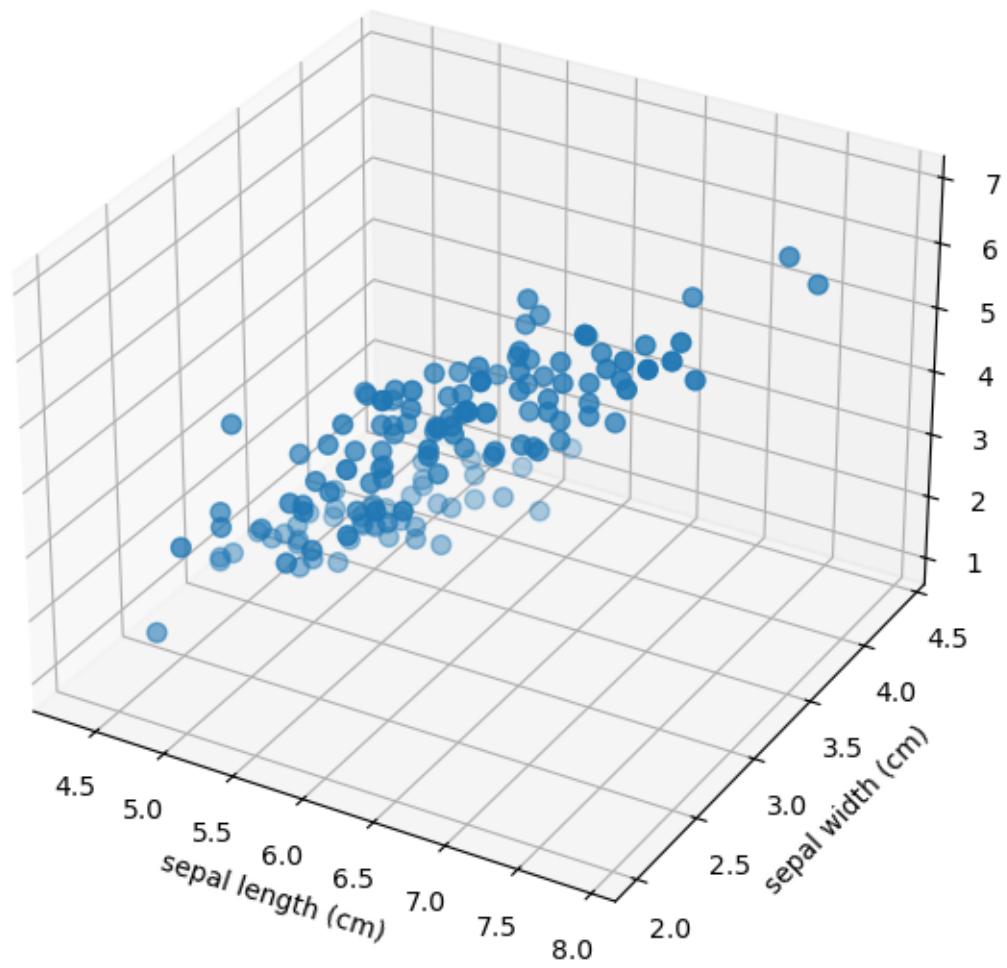
```

C:\Users\Neil\AppData\Local\Temp\ipykernel_42600\2503888613.py:18: UserWarning:
No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
ax.scatter(data.iloc[:, 0], data.iloc[:, 1], data.iloc[:, 2], cmap='viridis',
marker='o', s=50)

C:\Users\Neil\AppData\Local\Temp\ipykernel_42600\2503888613.py:23: UserWarning:
No artists with labels found to put in legend. Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
ax.legend()

c:\Users\Neil\anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1429:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
warnings.warn(

K-Means Clustering on Iris Dataset (3D)



K-Means Clustering on Iris Dataset (3D)

