Task 1 - Iris Flower Classfication

Import necessary libraries

In [83]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score
```

Load the Iris dataset

In [84]:

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.dat
names = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class'
df = pd.read_csv(url, names=names)
df
```

Out[84]:

	sepal_length	sepal_width	petal_length	petal_width	class
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Split the dataset into features (X) and target (y)

In [85]:

```
x = df.drop('class', axis=1)
y = df['class']
x
```

Out[85]:

	sepal_length	sepal_width	petal_length	petal_width

In [85]:
```
x = df.drop('class', axis=1)
y = df['class']
x
```

Out[85]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

Split the data into training and testing sets

In [86]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, rando
```

In [87]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [88]:
```
k = 3
```

Train the classifier on the training data

In [89]:
```
knn_classifier = KNeighborsClassifier(n_neighbors=k)
knn_classifier.fit(X_train, y_train)
```

Out[89]:
```
▾        KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

Make predictions on the test data

In [90]:
```
y_pred = knn_classifier.predict(X_test)
```

In [91]:
```
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

In [92]:
```
print(f"Accuracy: {accuracy}")
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 1.0
Classification Report:
                  precision    recall  f1-score   support
```

In [92]:

```
print(f"Accuracy: {accuracy}")
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 1.0
Classification Report:
                  precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        19
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00        13

       accuracy                           1.00        45
      macro avg       1.00      1.00      1.00        45
   weighted avg       1.00      1.00      1.00        45
```

In [ ]:

```
```