

IRIS FLOWER CLASSIFICATION

Import necessary modules

```
In [3]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

read the CSV file using pandas The head method is used to display the first five rows

```
In [4]: import os
working_directory = os.getcwd()
print (working_directory)
```

/Users/ishu

```
In [5]: path = working_directory + '/Desktop/Iris.csv'
data = pd.read_csv (path)
data.head()
```

Out[5]:

	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

```
In [6]: X = data.drop('Species', axis=1).values
y = data['Species'].values
```

Now split the data into training data and testing data

```
In [7]: X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.02, random_state=0)

#Scale the training data such that the mean of each column becomes equal to zero, and the standard deviation of each column is one.

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```

```
In [8]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
```

Out[8]:

▼ DecisionTreeClassifier

DecisionTreeClassifier()

```
In [9]: X_test = scaler.transform(X_test)
y_pred = model.predict(X_test)
```

```
In [10]: # Measure the performance of Training data.
model.score(X_train, y_train)
```

Out[10]: 1.0

```
In [11]: model.score(X_test, y_test)
```

Out[11]: 0.3333333333333333

```
In [12]: # Build a Confusion matrix for actual label data and predicted label data.
confusion_matrix(y_test, y_pred)
```

Out[12]:

```
array([[0, 0, 1],
       [0, 0, 1],
       [0, 0, 1]])
```

Find all the evaluation metrics like accuracy,precision,recall and f1-score

```
In [13]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
Iris-setosa	0.00	0.00	0.00	1
Iris-versicolor	0.00	0.00	0.00	1
Iris-virginica	0.33	1.00	0.50	1
accuracy			0.33	3
macro avg	0.11	0.33	0.17	3
weighted avg	0.11	0.33	0.17	3

/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []: