Oasis Infobyte Internship

Intern Name -Akshay Anandkar

Task 1-IRIS FLOWER CLASSIFICATION

Problem Statement-Iris flower has three species; setosa, versicolor, and virginica, which differs according to their measurements. Now assume that you have the measurements of the iris flowers according to their species, and here your task is to train a machine learning model that can learn from the measurements of the iris species and classify them.

In [1]:
```
#import all require liabraries
import pandas as pd
import numpy as np
```

In [2]:
```
#now importing iris dataset
iris=pd.read_csv(r"D:\Data-Science-Internship\Iris.csv")
iris.head(10)
```

Out[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
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [3]:
```
#checking dataset
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [4]:
```
#Checking for null values
iris.isnull().sum()
```

Out[4]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [5]:
```
#Taking independent varaible
X=iris.drop(['Species'],axis=1)
X
```

Out[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

In [6]:
```
#dependent varaible
y=iris['Species']
```

In [7]:
```
#Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size=0.02,random_state=0)
```

In [8]:
```
print(X_train)
```

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
107  108            7.3           2.9            6.3           1.8
7      8            5.0           3.4            1.5           0.2
100  101            6.3           3.3            6.0           2.5
40    41            5.0           3.5            1.3           0.3
86    87            6.7           3.1            4.7           1.5
..   ...            ...           ...            ...           ...
9     10            4.9           3.1            1.5           0.1
103  104            6.3           2.9            5.6           1.8
67    68            5.8           2.7            4.1           1.0
117  118            7.7           3.8            6.7           2.2
47    48            4.6           3.2            1.4           0.2

[147 rows x 5 columns]
```

In [9]:
```
#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [10]:
```
#Applying decision tress classifier model
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train,y_train)
```

Out[10]:
```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [11]:
```
X_test=sc.transform(X_test)
```

```
C:\Users\Akshay\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but StandardScaler was fitted wi
th feature names
  warnings.warn(
```

In [12]:
```
#Gettig y_prediction
y_pred=classifier.predict(X_test)
y_pred
```

Out[12]:
```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

In [13]:
```
classifier.score(X_train,y_train)
```

Out[13]:
```
1.0
```

In [14]:
```
classifier.score(X_test,y_test)
```

Out[14]:
```
0.3333333333333333
```

In [15]:
```
#Applying Classification report
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

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

In [16]:
```
cr=classification_report(y_test,y_pred)
print(cr)
```

```
                 precision    recall  f1-score   support

    Iris-setosa       0.50      1.00      0.67         1
Iris-versicolor       0.00      0.00      0.00         1
 Iris-virginica       0.00      0.00      0.00         1

       accuracy                           0.33         3
      macro avg       0.17      0.33      0.22         3
   weighted avg       0.17      0.33      0.22         3
```

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

In [ ]:

In [ ]: