Prediction using Decision Tree Algorithm

Create the Decision Tree classifier and visualize it graphically.

The purpose is if we feed any new data to this classifier, it would be able to predict the right class accordingly.

Shital More

```
In [ ]:
In [1]: import os #Standard imports
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
In [2]: | from sklearn.datasets import load_iris
In [3]: | df=pd.read_csv("C:\\Users\\Shri Krupa\\Downloads\\Iris.csv")
In [4]: from sklearn import datasets
In [5]: data=datasets.load iris()
In [6]: data.keys()
Out[6]: dict keys(['data', 'target', 'target names', 'DESCR', 'feature names', 'filenam
        e'])
In [7]: data["data"][:5]
Out[7]: array([[5.1, 3.5, 1.4, 0.2],
               [4.9, 3., 1.4, 0.2],
               [4.7, 3.2, 1.3, 0.2],
               [4.6, 3.1, 1.5, 0.2],
               [5., 3.6, 1.4, 0.2]])
```

```
In [8]: data["target"]
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
           In [9]: data["target names"]
Out[9]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')</pre>
In [10]: | df=pd.DataFrame(data["data"],columns=data["feature names"])
In [11]: | df["target"]=data["target"]
In [12]: | df.head()
Out[12]:
        sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) target
      0
               5.1
                                   1.4
                         3.5
                                            0.2
                                                 0
       1
               4.9
                         3.0
                                   1.4
                                            0.2
                                                 0
      2
               4.7
                         3.2
                                  1.3
                                            0.2
                                                 0
       3
               4.6
                         3.1
                                   1.5
                                            0.2
                                                 0
               5.0
                         3.6
                                   1.4
                                            0.2
                                                 0
      from sklearn.model selection import train test split
In [13]:
      x=df.drop(columns=['target'])
      y=df['target']
      x_train, x_test, y_train , y_test=train_test_split(x,y,test_size=0.30)
```

Decision tree

Print metric to get performance

```
In [16]: print("Accuracy:", model.score(x_test,y_test)*100)
```

Accuracy: 95.55555555556

conclusion

A decision tree is one of most frequently and widely used supervised machine learning algorithms that can perform both regression and classification tasks. The intuition behind the decision tree algorithm is simple, yet also very powerful.

