LETS GROW MORE - Virtual Internship 2023

Name :Shiva Dagdu Mehenge

Data Science Intern

Task 6- Prediction using Decision Tree Algorithm

Importing Libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
from sklearn import tree
from sklearn.model_selection import train_test_split
```

In [9]:
```
df=pd.read_csv('Iris.csv',index_col=0)
```

Undestanding the data

In [10]:
```
1 df.head()
```

Out[10]:

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

In [11]:
```
1 df.isnull().sum().any()
```

Out[11]:
```
False
```

In [12]:
```
1 df.shape
```

Out[12]:
```
(150, 5)
```

In [13]:
```
1 df.describe()
```

Out[13]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [14]:
```
#Checking the unique values in target variable
df.Species.unique()
```

Out[14]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [15]:
```
#splitting our data for training and testing
x=df[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']].values
```

In [16]:
```
y=df.Species.values
```

In [17]:
```
x
```

Out[17]:
```
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],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
```

In [18]:

```
1 y
```

Out[18]:
```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
```

```
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica'], dtype=object)
```

In [19]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [20]:
```
#creating the model
model=tree.DecisionTreeClassifier(criterion='entropy')
```

In [21]:
```
#training our classifier
model.fit(x_train,y_train)
```

Out[21]:
```
DecisionTreeClassifier(criterion='entropy')
```

In [22]:
```
#predicting using the  classifier
y_pred= model.predict(x_test)

```

In [23]:
```
model.score(x_test,y_test)
```

Out[23]:
```
0.9736842105263158
```

Evaluation

In [24]:
```
from sklearn.metrics import accuracy_score,classification_report
accuracy_score(y_test,y_pred)
```

Out[24]:
```
0.9736842105263158
```

In [25]:

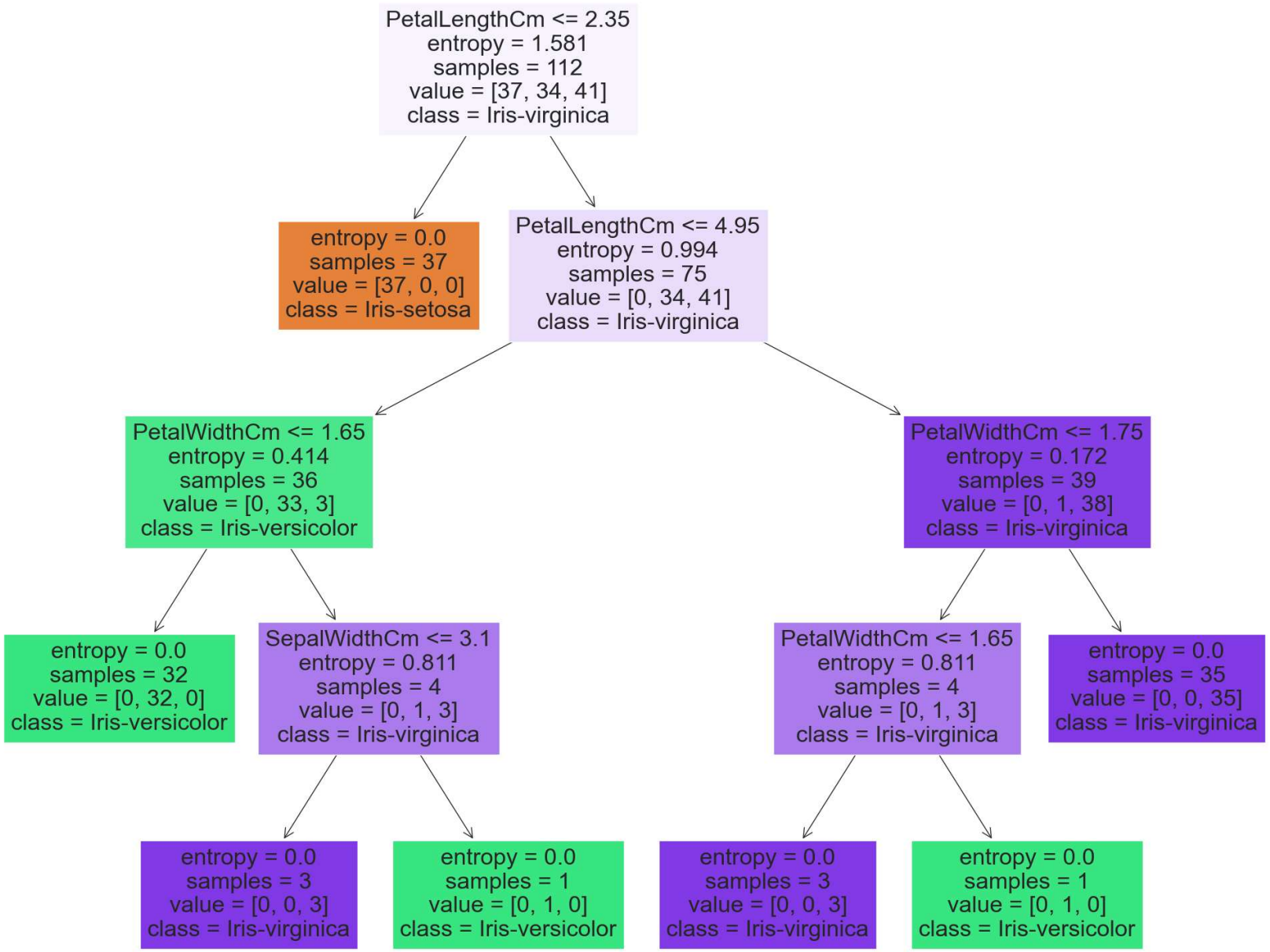
```
#Evaluation Summary
print(classification_report(y_test,y_pred))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        13
Iris-versicolor       1.00      0.94      0.97        16
 Iris-virginica       0.90      1.00      0.95         9

      accuracy                           0.97        38
     macro avg       0.97      0.98      0.97        38
  weighted avg       0.98      0.97      0.97        38
```

Decision Tree Visualization

In [26]:
```
cn = df["Species"].unique().tolist()
fig = plt.figure(figsize=(25,20))
tree.plot_tree(model, feature_names=df.columns[:-1],class_names=cn,filled=True)

```

Out[26]:
```
[Text(0.4, 0.9, 'PetalLengthCm <= 2.35\nentropy = 1.581\nsamples = 112\nvalue = [37, 34, 41]\nclass = Iris-v
irginica'),
 Text(0.3, 0.7, 'entropy = 0.0\nsamples = 37\nvalue = [37, 0, 0]\nclass = Iris-setosa'),
 Text(0.5, 0.7, 'PetalLengthCm <= 4.95\nentropy = 0.994\nsamples = 75\nvalue = [0, 34, 41]\nclass = Iris-vir
ginica'),
 Text(0.2, 0.5, 'PetalWidthCm <= 1.65\nentropy = 0.414\nsamples = 36\nvalue = [0, 33, 3]\nclass = Iris-versi
color'),
 Text(0.1, 0.3, 'entropy = 0.0\nsamples = 32\nvalue = [0, 32, 0]\nclass = Iris-versicolor'),
 Text(0.3, 0.3, 'SepalWidthCm <= 3.1\nentropy = 0.811\nsamples = 4\nvalue = [0, 1, 3]\nclass = Iris-virginic
a'),
 Text(0.2, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = Iris-virginica'),
 Text(0.4, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = Iris-versicolor'),
 Text(0.8, 0.5, 'PetalWidthCm <= 1.75\nentropy = 0.172\nsamples = 39\nvalue = [0, 1, 38]\nclass = Iris-virgi
nica'),
 Text(0.7, 0.3, 'PetalWidthCm <= 1.65\nentropy = 0.811\nsamples = 4\nvalue = [0, 1, 3]\nclass = Iris-virgini
ca'),
 Text(0.6, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = Iris-virginica'),
 Text(0.8, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = Iris-versicolor'),
 Text(0.9, 0.3, 'entropy = 0.0\nsamples = 35\nvalue = [0, 0, 35]\nclass = Iris-virginica')]
```


PetalLengthCm <= 2.35
entropy = 1.581
samples = 112
value = [37, 34, 41]
class = Iris-virginica
entropy = 0.0
samples = 37
value = [37, 0, 0]
class = Iris-setosa
PetalLengthCm <= 4.95
entropy = 0.994
samples = 75
value = [0, 34, 41]
class = Iris-virginica
PetalWidthCm <= 1.65
entropy = 0.414
samples = 36
value = [0, 33, 3]
class = Iris-versicolor
PetalWidthCm <= 1.75
entropy = 0.172
samples = 39
value = [0, 1, 38]
class = Iris-virginica
entropy = 0.0
samples = 32
value = [0, 32, 0]
class = Iris-versicolor
SepalWidthCm <= 3.1
entropy = 0.811
samples = 4
value = [0, 1, 3]
class = Iris-virginica
PetalWidthCm <= 1.65
entropy = 0.811
samples = 4
value = [0, 1, 3]
class = Iris-virginica
entropy = 0.0
samples = 35
value = [0, 0, 35]
class = Iris-virginica
entropy = 0.0
samples = 3
value = [0, 0, 3]
class = Iris-virginica
entropy = 0.0
samples = 1
value = [0, 1, 0]
class = Iris-versicolor
entropy = 0.0
samples = 3
value = [0, 0, 3]
class = Iris-virginica
entropy = 0.0
samples = 1
value = [0, 1, 0]
class = Iris-versicolor

Prediction

In [27]:
```
y_pred=model.predict([[5.8, 2.8, 5.1, 2.4],
                      [6. , 2.2, 4. , 1. ],
                      [5.5, 4.2, 1.4, 0.2],
                      [7.3, 2.9, 6.3, 1.8],
                      [5. , 3.4, 1.5, 0.2]])
y_pred #This is our predicted value
```

Out[27]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa'], dtype=object)
```

In [28]:
```
y_test[:5] #this is our actual values
```

Out[28]:
```
array(['Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa'], dtype=object)
```

Thank You !!

In [ ]:
```

```