**LETS GROW MORE (VIRTUAL INTERNSHIP MAY-2023) **

Name- Arushi

Data Science Intern

Task 2- Prediction using Decision Tree Algorithm

Importing important libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Reading data

```
data=pd.read_csv('/content/Iris.csv',index_col=0)
```

Getting information about the data

```
data.head()
```

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

```
#Checking null values
data.isnull().sum()
```

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
```

✓ 0s completed at 9:06 AM

```
#Checking number of rows and columns in dataset
data.shape
```

```
(150, 5)
```

```
#Overall stats of the dataset
data.describe()
```

	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

Here Target variable is Species

```
#Checking the unique values in target variable
data.Species.unique()
```

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

```
#Splitting data into training and testing
from sklearn import tree
from sklearn.model_selection import train_test_split
X=data[['SepalLengthCm','SepalWidthCm' ,    'PetalLengthCm' ,   'PetalWidthCm']]
Y=data.Species.values
```

```
X
```

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	5.1	3.5	1.4	0.2

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

150 rows × 4 columns

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

150 rows × 4 columns

Y

```
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-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)
```

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25,random_state=0)
```

Creating the model

```
model=tree.DecisionTreeClassifier(criterion='entropy')
```

```
#Training our classifier
model.fit(X_train,Y_train)
```

```
▾          DecisionTreeClassifier
```

```
DecisionTreeClassifier(criterion='entropy')
```

```
#Prediction
Y_pred=model.predict(X_test)
```

```
model.score(X_test,Y_test)
```

```
0.9736842105263158
```

Evaluation

```
from sklearn.metrics import accuracy_score,classification_report
accuracy_score(Y_test,Y_pred)
```

```
0.9736842105263158
```

```
#Evaluation Summary
print(classification_report(Y_test,Y_pred))
```

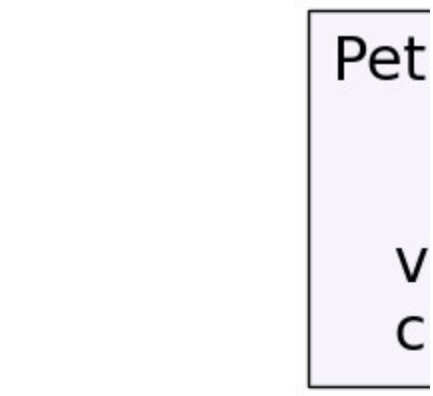
```
                    precision    recall  f1-score   support

        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
```

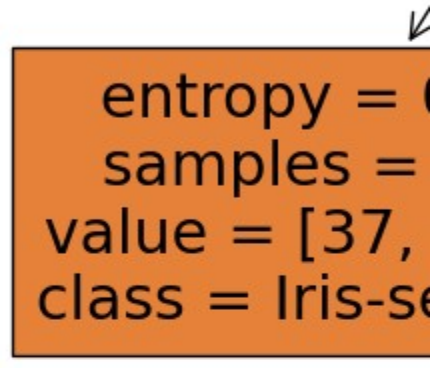
Visualization of Decision Tree

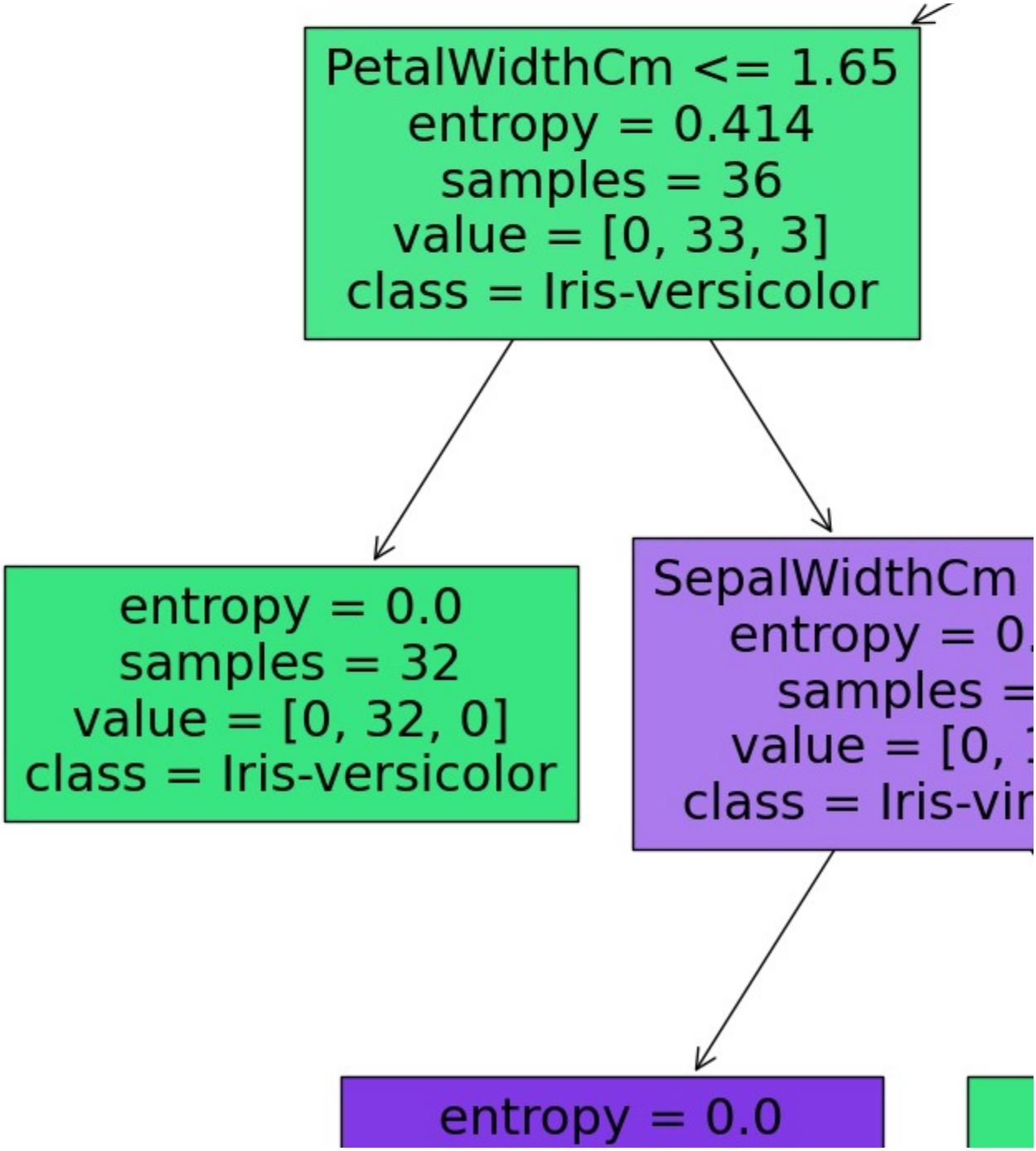
```
cn=data["Species"].unique().tolist()
fig = plt.figure(figsize=(25,20))
tree.plot_tree(model, feature_names=data.columns[:-1],class_names=cn,filled=True)
```

```
[Text(0.4, 0.9, 'PetalLengthCm <= 2.35\nentropy = 1.581\nsamples = 112\nvalue = [37,
34, 41]\nclass = Iris-virginica'),
 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-virginica'),
 Text(0.2, 0.5, 'PetalWidthCm <= 1.65\nentropy = 0.414\nsamples = 36\nvalue = [0,
33, 3]\nclass = Iris-versicolor'),
 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-virginica'),
 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, 'PetalLengthCm <= 5.05\nentropy = 0.172\nsamples = 39\nvalue = [0,
1, 38]\nclass = Iris-virginica'),
 Text(0.7, 0.3, 'SepalLengthCm <= 6.5\nentropy = 0.811\nsamples = 4\nvalue = [0, 1,
3]\nclass = Iris-virginica'),
 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')]
```







Prediction

```
#Predicted Value
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
```

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

```
#Actual value
Y_test[:5]
```

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

THANKYOU