Prediction using Decision Tree Algorithm

A decision tree is a supervised learning algorithm that can be used for both classification and regression problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

The decision tree algorithm works by recursively splitting the data into subsets based on the most significant feature at each node of the tree. The splitting continues until all of the subsets are pure, or until a certain stopping criterion is met.

Import Libraries

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
```

Reading Data

```
data = pd.read_csv("/content/Iris.csv")
```

```
data
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa

```
data.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
```

149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
X = data[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']].values
X[0:5]
```

```
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]])
```

```
Y = data['Species']
Y[0:5]
```

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Species, dtype: object
```

```
from sklearn.model_selection import train_test_split
```

```
X_trainset, X_testset, Y_trainset, Y_testset = train_test_splittrain_X, test_X, train_y, test
```

```
SpeciesTree = DecisionTreeClassifier(criterion = 'entropy', max_depth = 4)
```

```
SpeciesTree
```

```
▾ DecisionTreeClassifier
```

```
SpeciesTree.fit(X_trainset, Y_trainset)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=4)
```

Prediction

```
predTree = SpeciesTree.predict(X_testset)
```

```
print(predTree [0:5])
print(Y_testset[0:5])
```

```
['Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa']
114     Iris-virginica
62     Iris-versicolor
33         Iris-setosa
107     Iris-virginica
7          Iris-setosa
Name: Species, dtype: object
```

Evaluation

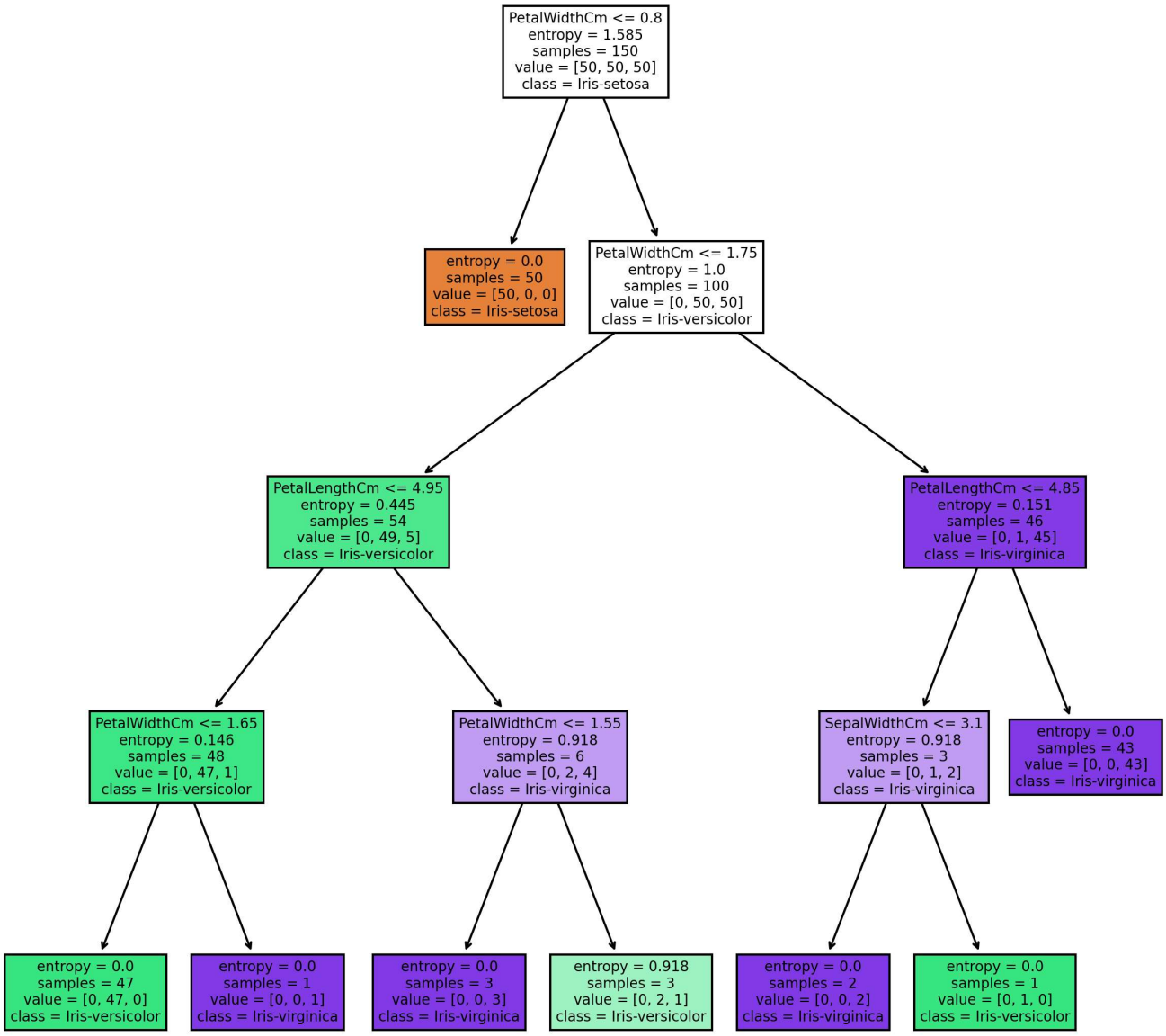
```
from sklearn import metrics
import matplotlib.pyplot as plt
```

```
print("DecisionTrees's Accuracy: ",metrics.accuracy_score(Y_testset, predTree))
```

```
DecisionTrees's Accuracy:  0.9777777777777777
```

Visualization

```
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
fn = data.columns[1:5]
cn = data["Species"].unique().tolist()
SpeciesTree.fit(X, Y)
fig, axes = plt.subplots(nrows = 1,ncols = 1, figsize = (10,10), dpi = 300)
tree.plot_tree(SpeciesTree,feature_names = fn, class_names = cn, filled = True);
```



Prediction species for set of values

```
X_new = [[6.3,3.0,1.3,0.2]]

predTree = SpeciesTree.predict(X_new)

predTree

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

✓ 0s completed at 7:46 PM