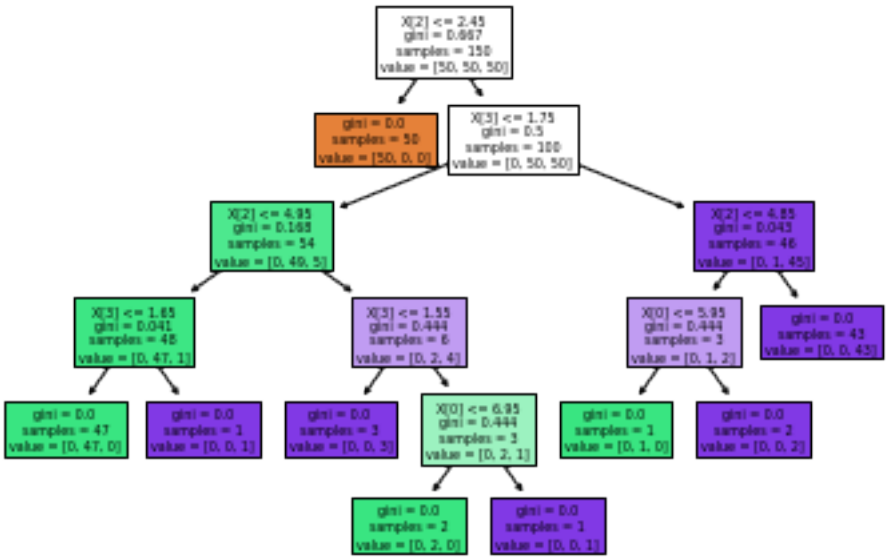
dtc_irisdata

January 27, 2022

[ ]:
```
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

df= sns.load_dataset("iris")
df.head(1)
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
X= df.iloc[:,:-1]
y=df.iloc[:, -1:] # conisder only last column
```

0.0.1 Saving High Resolution plots

[ ]:
```
model = DecisionTreeClassifier()
model.fit(X,y)
plt.title("Decision Model trained model of Iris")

model.predict(X)
plot_tree(model, filled=True)
# save in tiff and jpeg
plt.savefig('tiff_compressed.tiff',dpi=600, format='tiff',
facecolor='white', edgecolor='none',
pil_kwargs={"compression": "tiff_lzw"})

plt.show()
```



0.0.2 Running a Saved Model

[ ]:
```
import joblib

joblib.dump(model,"iris.joblib")

# How to run save stored model (Assignment)
saved_model=joblib.load('iris.joblib')

Final_predictions=saved_model.predict(X)
#Final_predictions
```

0.0.3 80-20

[ ]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
#model.fit(X,y)
#Accuracy test
score = model.score(X_test,y_test)
print("The accuracy score with 80-20 (X_test) and (y_test) is ",score)

# checking predicted values with input test data
predicted_values = model.predict(X_test)
print("The predicted values from 20% of test input is",predicted_values,"\n")

superscore = accuracy_score(y_test,predicted_values)
```

```
print("The accuracy score of model when compared with twenty percent original␣
 ↪test values is",superscore)

#Checking unknown 5 Values
unknownvalues=[[5.2,3.6,1.8,0.1],[5.2,3.5,1.3,1.1],[5.2,3.6,1.4,0.1],[5.5,3.6,1.
 ↪66,0.1],[5.4,3.6,1.8,0.1]]
unknownvalues = model.predict(unknownvalues)
print("\n The prediction of 5 unknown values is ",unknownvalues)
```

```
The accuracy score with 80-20 (X_test) and (y_test) is  1.0
The predicted values from 20% of test input is ['virginica' 'versicolor'
'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'versicolor' 'versicolor' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'versicolor' 'versicolor' 'setosa' 'versicolor' 'versicolor'
 'setosa' 'setosa' 'virginica' 'versicolor' 'setosa' 'setosa' 'virginica'
 'setosa' 'setosa' 'versicolor' 'versicolor' 'setosa']

The accuracy score of model when compared with twenty percent original test
values is 1.0

 The prediction of 5 unknown values is  ['setosa' 'setosa' 'setosa' 'setosa'
'setosa']

C:\Users\dell7450\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:450: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

0.0.4 90-10

[ ]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.1,␣
 ↪random_state=0)
#model.fit(X,y)
#Accuracy test
score = model.score(X_test,y_test)
print("The accuracy score with 90-10 (X_test) and (y_test) is ",score)

# checking predicted values with input test data
predicted_values = model.predict(X_test)
print("The predicted values from 10% of test input is",predicted_values,"\n")

superscore = accuracy_score(y_test,predicted_values)
print("The accuracy score of model when compared with ten percent original test␣
 ↪values is",superscore)

#Checking unknown 5 Values
unknownvalues=[[5.2,3.6,1.8,0.1],[5.2,3.5,1.3,1.1],[5.2,3.6,1.4,0.1],[5.5,3.6,1.
 ↪66,0.1],[5.4,3.6,1.8,0.1]]
```

```
unknownvalues = model.predict(unknownvalues)
print("\n The prediction of 5 unknown values is ",unknownvalues)
```

```
The accuracy score with 90-10 (X_test) and (y_test) is  1.0
The predicted values from 10% of test input is ['virginica' 'versicolor'
'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'versicolor' 'versicolor' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'versicolor' 'versicolor']

The accuracy score of model when compared with ten percent original test values
is 1.0

 The prediction of 5 unknown values is  ['setosa' 'setosa' 'setosa' 'setosa'
'setosa']

C:\Users\dell7450\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:450: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

0.0.5 70-30

[ ]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=0)
#model.fit(X,y)
#Accuracy test
score = model.score(X_test,y_test)
print("The accuracy score with 70-30 (X_test) and (y_test) is ",score)

# checking predicted values with input test data
predicted_values = model.predict(X_test)
print("The predicted values from 30% of test input is",predicted_values,"\n")

superscore = accuracy_score(y_test,predicted_values)
print("The accuracy score of model when compared with thirty percent original␣
 ↪test values is",superscore)

#Checking unknown 5 Values
unknownvalues=[[5.2,3.6,1.8,0.1],[5.2,3.5,1.3,1.1],[5.2,3.6,1.4,0.1],[5.5,3.6,1.
 ↪66,0.1],[5.4,3.6,1.8,0.1]]
unknownvalues = model.predict(unknownvalues)
print("\n The prediction of 5 unknown values is ",unknownvalues)
```

```
The accuracy score with 70-30 (X_test) and (y_test) is  1.0
The predicted values from 30% of test input is ['virginica' 'versicolor'
'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'versicolor' 'versicolor' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'versicolor' 'versicolor' 'setosa' 'versicolor' 'versicolor'
 'setosa' 'setosa' 'virginica' 'versicolor' 'setosa' 'setosa' 'virginica'
```

```
 'setosa' 'setosa' 'versicolor' 'versicolor' 'setosa' 'virginica'
 'versicolor' 'setosa' 'virginica' 'virginica' 'versicolor' 'setosa'
 'versicolor' 'versicolor' 'versicolor' 'virginica' 'setosa' 'virginica'
 'setosa' 'setosa']

The accuracy score of model when compared with thirty percent original test
values is 1.0

 The prediction of 5 unknown values is  ['setosa' 'setosa' 'setosa' 'setosa'
'setosa']

C:\Users\dell7450\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:450: UserWarning: X does not have valid feature names,
but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```