assignment 15

March 8, 2022

1 Decision tree predicting the categorical column of species in Penguins dataset

[1]:
```
import pandas as pd
import seaborn as sns
```

[2]:
```
penguins = sns.load_dataset("penguins")
penguins.head()
```

[2]:
```
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0  Adelie  Torgersen            39.1           18.7              181.0
1  Adelie  Torgersen            39.5           17.4              186.0
2  Adelie  Torgersen            40.3           18.0              195.0
3  Adelie  Torgersen             NaN            NaN                NaN
4  Adelie  Torgersen            36.7           19.3              193.0

   body_mass_g     sex
0       3750.0    Male
1       3800.0  Female
2       3250.0  Female
3          NaN     NaN
4       3450.0  Female
```

[3]:
```
from sklearn.model_selection import train_test_split
```

[4]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import graphviz
```

[5]:
```
penguins.isna().sum()
```

[5]:
```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
```

```
sex                  11
dtype: int64
```

[6]:
```
penguins_model = penguins.dropna()
```

[7]:
```
penguins_train, penguins_test = train_test_split(penguins_model, test_size=0.3,␣
↪stratify=penguins_model['species'], random_state=42)
```

[8]:
```
features= ['bill_length_mm','body_mass_g']
dt_classification = DecisionTreeClassifier(max_depth = 2) # Increase max_depth␣
↪to see effect in the plot
dt_classification.fit(penguins_train[features], penguins_train['species'])
```

[8]:
```
DecisionTreeClassifier(max_depth=2)
```

[9]:
```
def calculate_accuracy(predictions, actuals):
    if(len(predictions) != len(actuals)):
        raise Exception("The amount of predictions did not equal the amount of␣
↪actuals")

    return (predictions == actuals).sum() / len(actuals)
```

[10]:
```
predictionsOnTrainset = dt_classification.predict(penguins_train[features])
predictionsOnTestset = dt_classification.predict(penguins_test[features])

accuracyTrain = calculate_accuracy(predictionsOnTrainset, penguins_train.
↪species)
accuracyTest = calculate_accuracy(predictionsOnTestset, penguins_test.species)

print("Accuracy on training set " + str(accuracyTrain))
print("Accuracy on test set " + str(accuracyTest))
```

```
Accuracy on training set 0.9356223175965666
Accuracy on test set 0.91
```

The accurracy are slightly different and that was expected, as the training data is almost always higher than the test set.

[11]:
```
from sklearn import tree
import graphviz

def plot_tree_classification(model, features, class_names):
    # Generate plot data
    dot_data = tree.export_graphviz(model, out_file=None,
                                    feature_names=features,
                                    class_names=class_names,
                                    filled=True, rounded=True,
                                    special_characters=True)
```

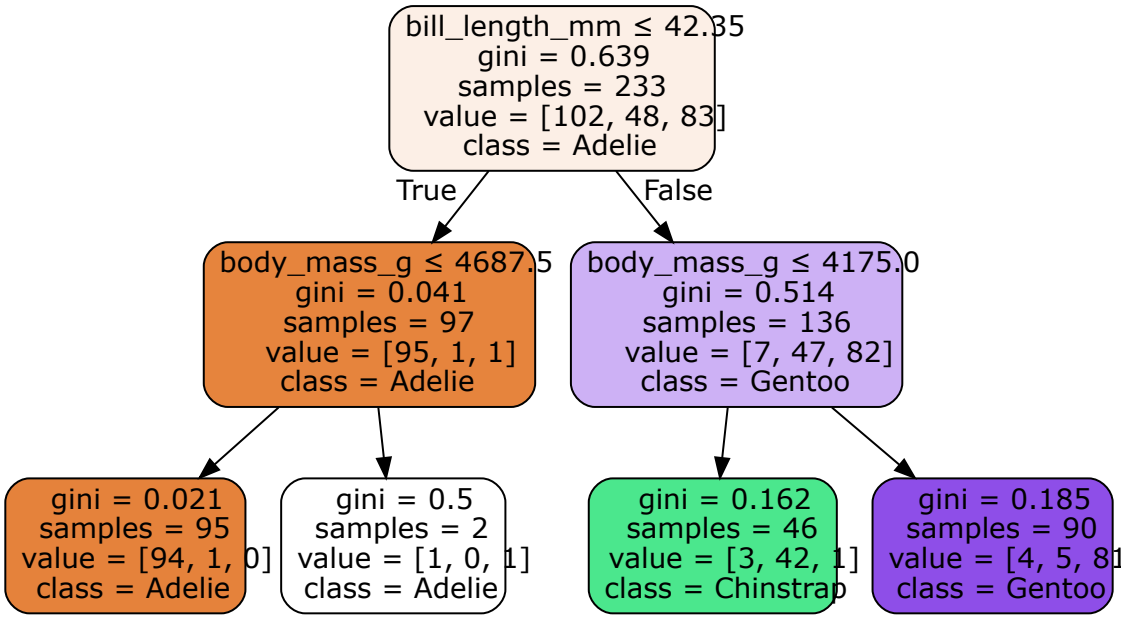
```
    # Turn into graph using graphviz
    graph = graphviz.Source(dot_data)

    # Write out a pdf
    graph.render("decision_tree")

    # Display in the notebook
    return graph
```

[12]:
```
plot_tree_classification(dt_classification, features, penguins_model.species.
↪unique())
```

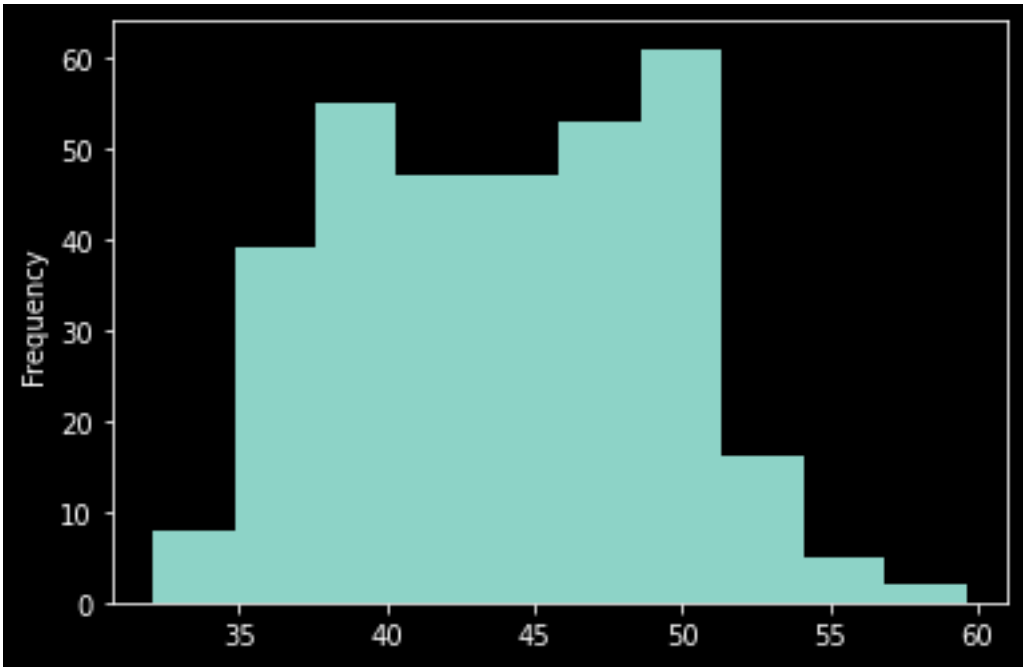
[12]:



Adelie penguins have small bills and are heavier Gentoo have larger bill_length and are heavier Chinstrap have larger bill_length and are lighter

[13]:
```
penguins_model['bill_length_mm'].plot(kind='hist')
```

[13]:
```
<AxesSubplot:ylabel='Frequency'>
```


Frequency
0
10
20
30
40
50
60
35
40
45
50
55
60