In [ ]:
```
"""
Project Topic:
Python Machine Learning (Decision Tree) analysis on "penguins_size.csv" dataset.

Dataset:
Dataset was downloaded from "https://github.com/allisonhorst/palmerpenguins/blob/main/inst/extdata/penguins.csv"

License:
Access in the CC0 (Creative Commons Zero) public domain.

Goals:
This project is totally generated for my educational and portfolio purposes to present data analysis skills in:
•       Data preprocessing,
•       Data analysis
•       Decision Tree (supervised machine learning), modeling including Python, scikit-learn, a
•       Model evaluation
•       Model visualization
Acknowledgement:
I do not own the dataset and am not redistributing it. All rights to this dataset relate to the original
uploader on 'https://github.com/allisonhorst/palmerpenguins'.

"""
```

In [ ]:
```
# Decision Tree (supervised machine learning),
```

In [114…
```
import pandas as pd
df = pd.read_csv('penguins_size.csv')
#print(df)
```

In [116…
```
# find total count of NaN in each column.
print(df.isnull().sum())
```

```
species              0
island               0
culmen_length_mm     2
culmen_depth_mm      2
flipper_length_mm    2
body_mass_g          2
sex                 10
dtype: int64
```

In [120…
```
# Drop all NaN values
df = df.dropna()
print(df)

# Updated number of rows.
print(len(df))
```

```
    species     island  culmen_length_mm  culmen_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
4    Adelie  Torgersen              36.7             19.3              193.0
5    Adelie  Torgersen              39.3             20.6              190.0
..      ...        ...               ...              ...                ...
338  Gentoo     Biscoe              47.2             13.7              214.0
340  Gentoo     Biscoe              46.8             14.3              215.0
341  Gentoo     Biscoe              50.4             15.7              222.0
342  Gentoo     Biscoe              45.2             14.8              212.0
343  Gentoo     Biscoe              49.9             16.1              213.0

     body_mass_g     sex
0         3750.0    MALE
1         3800.0  FEMALE
2         3250.0  FEMALE
4         3450.0  FEMALE
5         3650.0    MALE
..           ...     ...
338       4925.0  FEMALE
340       4850.0  FEMALE
341       5750.0    MALE
342       5200.0  FEMALE
343       5400.0    MALE

[334 rows x 7 columns]
334
```

In [122…
```
# Check the updated information about dataset
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 334 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   species            334 non-null    object
 1   island             334 non-null    object
 2   culmen_length_mm   334 non-null    float64
 3   culmen_depth_mm    334 non-null    float64
 4   flipper_length_mm  334 non-null    float64
 5   body_mass_g        334 non-null    float64
 6   sex                334 non-null    object
dtypes: float64(4), object(3)
memory usage: 20.9+ KB
None
```

In [124…
```
# find total count of NaN in each column of the updated df.
print(df.isnull().sum())
```

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

Decision Tree

In [127…
```
# Change categorical variables in each column into one hot encoder.
pd.get_dummies(df)

print(pd.get_dummies(df))
```

```
     culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g  \
0                39.1             18.7              181.0       3750.0
1                39.5             17.4              186.0       3800.0
2                40.3             18.0              195.0       3250.0
4                36.7             19.3              193.0       3450.0
5                39.3             20.6              190.0       3650.0
..                ...              ...                ...          ...
338              47.2             13.7              214.0       4925.0
340              46.8             14.3              215.0       4850.0
341              50.4             15.7              222.0       5750.0
342              45.2             14.8              212.0       5200.0
343              49.9             16.1              213.0       5400.0

     species_Adelie  species_Chinstrap  species_Gentoo  island_Biscoe  \
0              True              False           False          False
1              True              False           False          False
2              True              False           False          False
4              True              False           False          False
5              True              False           False          False
..              ...                ...             ...            ...
338           False              False            True           True
340           False              False            True           True
341           False              False            True           True
342           False              False            True           True
343           False              False            True           True

     island_Dream  island_Torgersen  sex_.  sex_FEMALE  sex_MALE
0           False              True  False       False      True
1           False              True  False        True     False
2           False              True  False        True     False
4           False              True  False        True     False
5           False              True  False       False      True
..            ...               ...    ...         ...       ...
338         False             False  False        True     False
340         False             False  False        True     False
341         False             False  False       False      True
342         False             False  False        True     False
343         False             False  False       False      True

[334 rows x 13 columns]
```

In [129…
```
# drop species column
X = pd.get_dummies(df.drop('species', axis = 1), drop_first = True)
print(X)

# extract the species column.
y = df['species']
print(y)
```

```
     culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g  \
0                39.1             18.7              181.0       3750.0
1                39.5             17.4              186.0       3800.0
2                40.3             18.0              195.0       3250.0
4                36.7             19.3              193.0       3450.0
5                39.3             20.6              190.0       3650.0
..                ...              ...                ...          ...
338              47.2             13.7              214.0       4925.0
340              46.8             14.3              215.0       4850.0
341              50.4             15.7              222.0       5750.0
342              45.2             14.8              212.0       5200.0
343              49.9             16.1              213.0       5400.0

     island_Dream  island_Torgersen  sex_FEMALE  sex_MALE
0           False              True       False      True
1           False              True        True     False
2           False              True        True     False
4           False              True        True     False
5           False              True       False      True
..            ...               ...         ...       ...
338         False             False        True     False
340         False             False        True     False
341         False             False       False      True
342         False             False        True     False
343         False             False       False      True

[334 rows x 8 columns]
0      Adelie
1      Adelie
2      Adelie
4      Adelie
5      Adelie
        ...
338    Gentoo
340    Gentoo
341    Gentoo
342    Gentoo
343    Gentoo
Name: species, Length: 334, dtype: object
```

In [131…
```
# load train_test_split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 101)
```

In [133…
```
# vizualization using decisiontreeclassifier

# Load DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier


# Initialize the  Decision Tree Classifier (model)
model = DecisionTreeClassifier()

# fit the model with the dataset (X_train, y_train).
model.fit(X_train, y_train)
```

Out[133…
```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [135…
```
# Use model for prediction on the X_test

prediction_X_test = model.predict(X_test)
print(prediction_X_test)
```

```
['Chinstrap' 'Gentoo' 'Adelie' 'Chinstrap' 'Gentoo' 'Chinstrap' 'Adelie'
 'Gentoo' 'Chinstrap' 'Gentoo' 'Adelie' 'Adelie' 'Adelie' 'Gentoo'
 'Gentoo' 'Adelie' 'Gentoo' 'Adelie' 'Adelie' 'Adelie' 'Gentoo'
 'Chinstrap' 'Adelie' 'Adelie' 'Adelie' 'Adelie' 'Chinstrap' 'Gentoo'
 'Adelie' 'Chinstrap' 'Gentoo' 'Chinstrap' 'Gentoo' 'Adelie' 'Adelie'
 'Chinstrap' 'Adelie' 'Gentoo' 'Chinstrap' 'Gentoo' 'Adelie' 'Adelie'
 'Gentoo' 'Adelie' 'Adelie' 'Chinstrap' 'Chinstrap' 'Chinstrap'
 'Chinstrap' 'Chinstrap' 'Adelie' 'Adelie' 'Gentoo' 'Gentoo' 'Adelie'
 'Adelie' 'Chinstrap' 'Chinstrap' 'Gentoo' 'Adelie' 'Chinstrap' 'Gentoo'
 'Adelie' 'Adelie' 'Chinstrap' 'Gentoo' 'Chinstrap' 'Chinstrap' 'Gentoo'
 'Gentoo' 'Gentoo' 'Gentoo' 'Gentoo' 'Gentoo' 'Gentoo' 'Gentoo' 'Gentoo'
 'Gentoo' 'Adelie' 'Gentoo' 'Adelie' 'Adelie' 'Gentoo' 'Adelie']
```

In [137…
```
# Load classification_report
from sklearn.metrics import classification_report

# Show the classification report
print(classification_report(y_test, prediction_X_test))     # It gives classification repor. # Overall accuracy is about 96%.
```

```
              precision    recall  f1-score   support

      Adelie       0.97      0.94      0.95        33
   Chinstrap       0.90      0.95      0.93        20
      Gentoo       1.00      1.00      1.00        31

    accuracy                           0.96        84
   macro avg       0.96      0.96      0.96        84
weighted avg       0.97      0.96      0.96        84
```

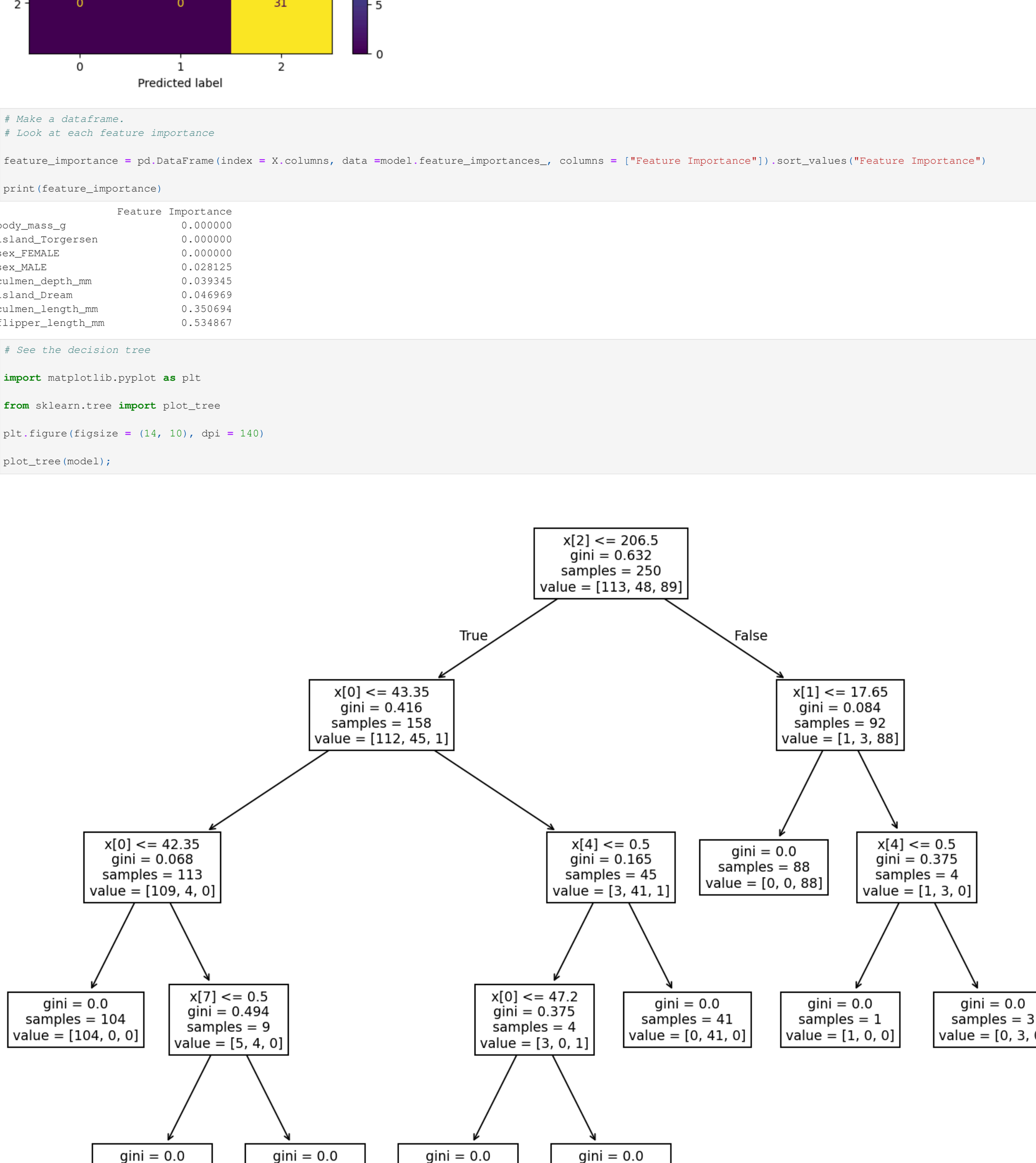
In [139…
```
# visualize the classification model's result
# plot confusion matrix

# Load confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Find confusion matrix
confusion_matrix = confusion_matrix(y_test, prediction_X_test)

# Show plot
Confusion_Matrix_Display = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix)
Confusion_Matrix_Display.plot()
```

Out[139… `<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x21516033a10>`

In [141…
```
# Make a dataframe.
# Look at each feature importance

feature_importance = pd.DataFrame(index = X.columns, data =model.feature_importances_, columns = ["Feature Importance"]).sort_values("Feature Importance")

print(feature_importance)
```

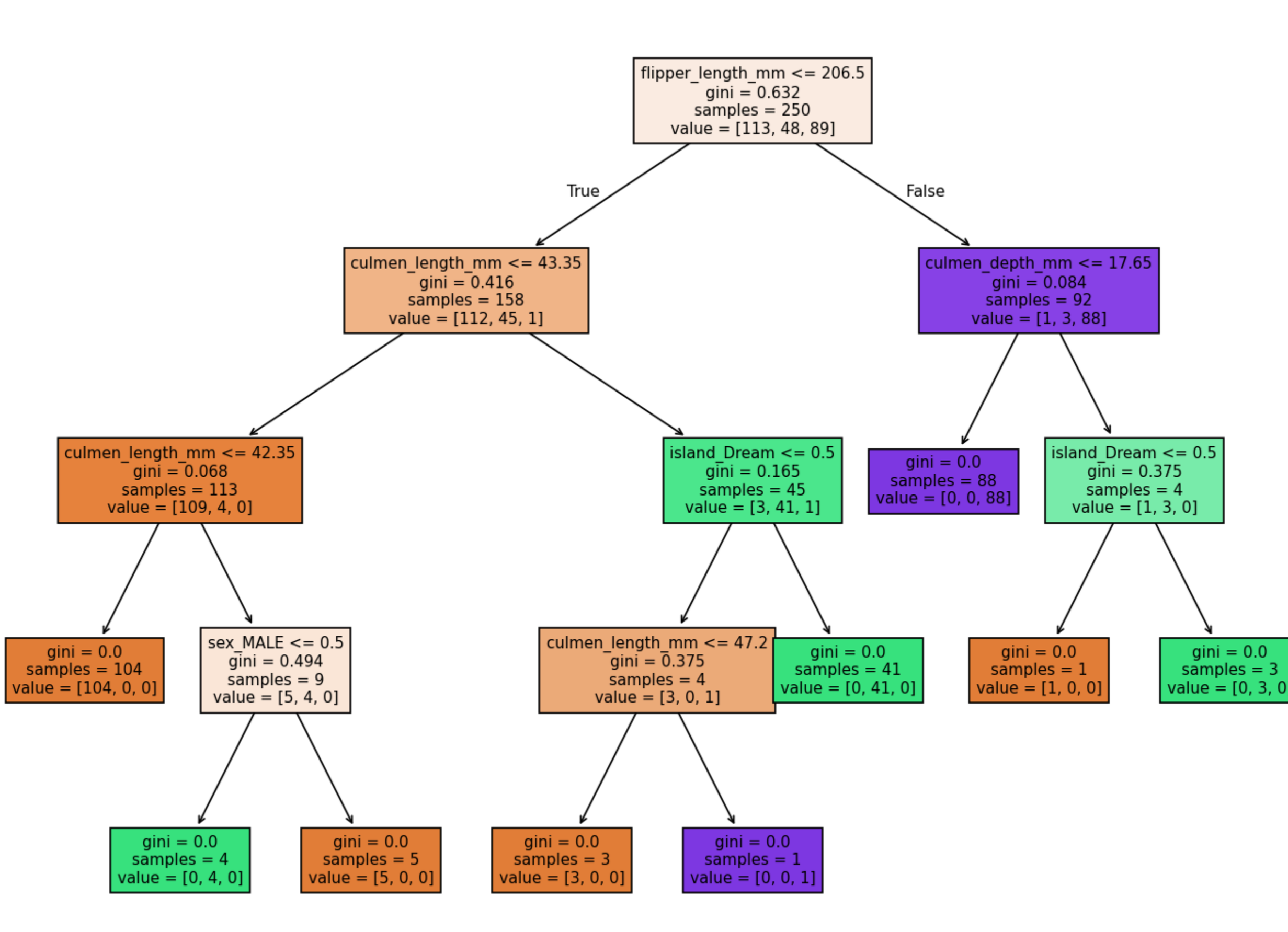
```
                   Feature Importance
body_mass_g                  0.000000
island_Torgersen             0.000000
sex_FEMALE                   0.000000
sex_MALE                     0.028125
culmen_depth_mm              0.039345
island_Dream                 0.046969
culmen_length_mm             0.350694
flipper_length_mm            0.534867
```

In [143…
```
# See the decision tree

import matplotlib.pyplot as plt

from sklearn.tree import plot_tree

plt.figure(figsize = (14, 10), dpi = 140)

plot_tree(model);
```



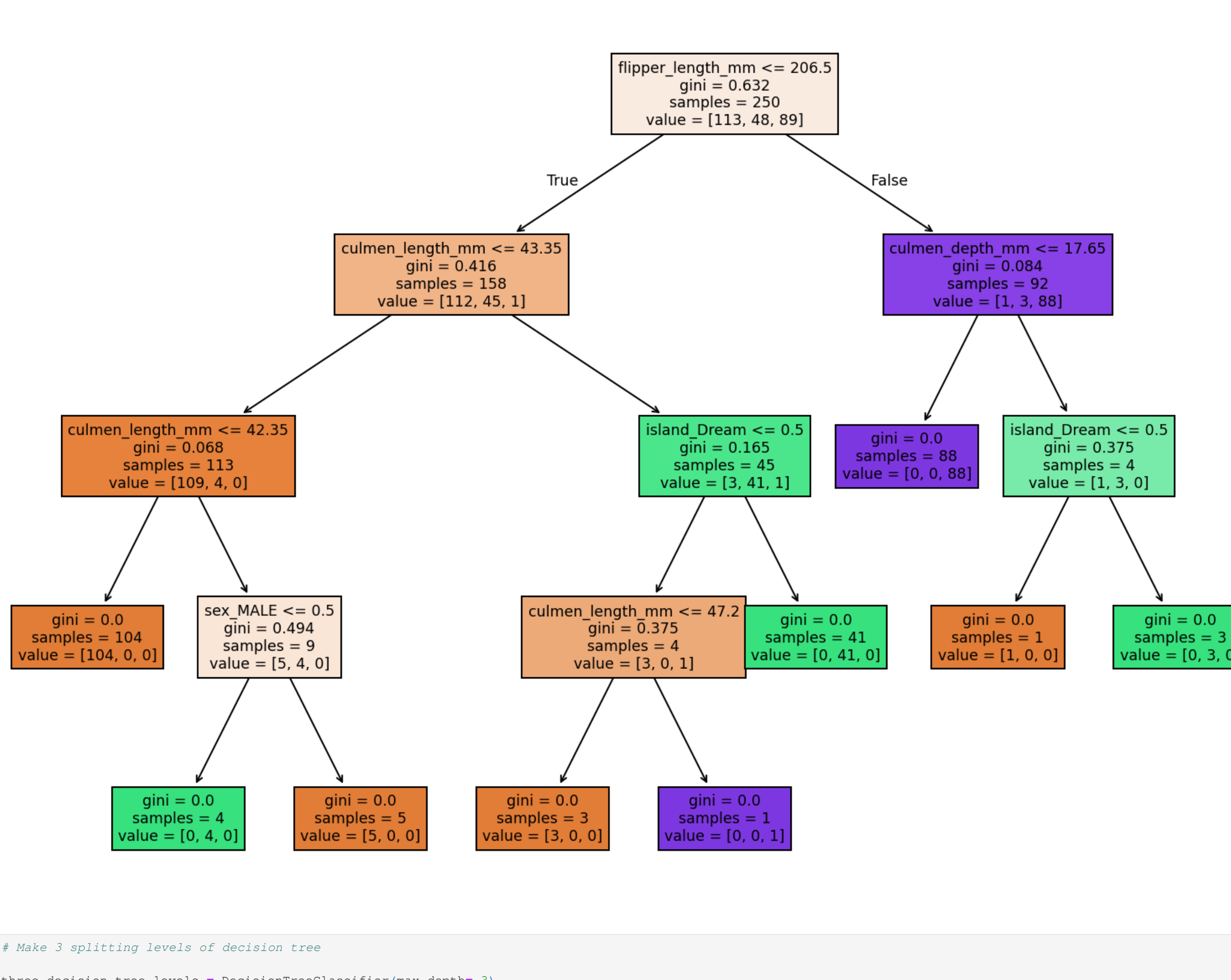
In [145…
```
# See the decision tree with features

plt.figure(figsize = (14, 10), dpi = 120)

plot_tree(model, feature_names=X.columns, filled = True, fontsize=9)
```

Out[145…
```
[Text(0.5714285714285714, 0.9, 'flipper_length_mm <= 206.5\ngini = 0.632\nsamples = 250\nvalue = [113, 48, 89]'),
 Text(0.35714285714285715, 0.7, 'culmen_length_mm <= 43.35\ngini = 0.416\nsamples = 158\nvalue = [112, 45, 1]'),
 Text(0.4642857142857143, 0.8, 'True  '),
 Text(0.14285714285714285, 0.5, 'culmen_length_mm <= 42.35\ngini = 0.068\nsamples = 113\nvalue = [109, 4, 0]'),
 Text(0.07142857142857142, 0.3, 'gini = 0.0\nsamples = 104\nvalue = [104, 0, 0]'),
 Text(0.21428571428571427, 0.3, 'sex_MALE <= 0.5\ngini = 0.494\nsamples = 9\nvalue = [5, 4, 0]'),
 Text(0.14285714285714285, 0.1, 'gini = 0.0\nsamples = 4\nvalue = [0, 4, 0]'),
 Text(0.2857142857142857, 0.1, 'gini = 0.0\nsamples = 5\nvalue = [5, 0, 0]'),
 Text(0.5714285714285714, 0.5, 'island_Dream <= 0.5\ngini = 0.165\nsamples = 45\nvalue = [3, 41, 1]'),
 Text(0.5, 0.3, 'culmen_length_mm <= 47.2\ngini = 0.375\nsamples = 4\nvalue = [3, 0, 1]'),
 Text(0.42857142857142855, 0.1, 'gini = 0.0\nsamples = 3\nvalue = [3, 0, 0]'),
 Text(0.5714285714285714, 0.1, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.6428571428571429, 0.3, 'gini = 0.0\nsamples = 41\nvalue = [0, 41, 0]'),
 Text(0.7857142857142857, 0.7, 'culmen_depth_mm <= 17.65\ngini = 0.084\nsamples = 92\nvalue = [1, 3, 88]'),
 Text(0.6785714285714286, 0.8, '  False'),
 Text(0.7142857142857143, 0.5, 'gini = 0.0\nsamples = 88\nvalue = [0, 0, 88]'),
 Text(0.8571428571428571, 0.5, 'island_Dream <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3, 0]'),
 Text(0.7857142857142857, 0.3, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),
 Text(0.9285714285714286, 0.3, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0]')]
```



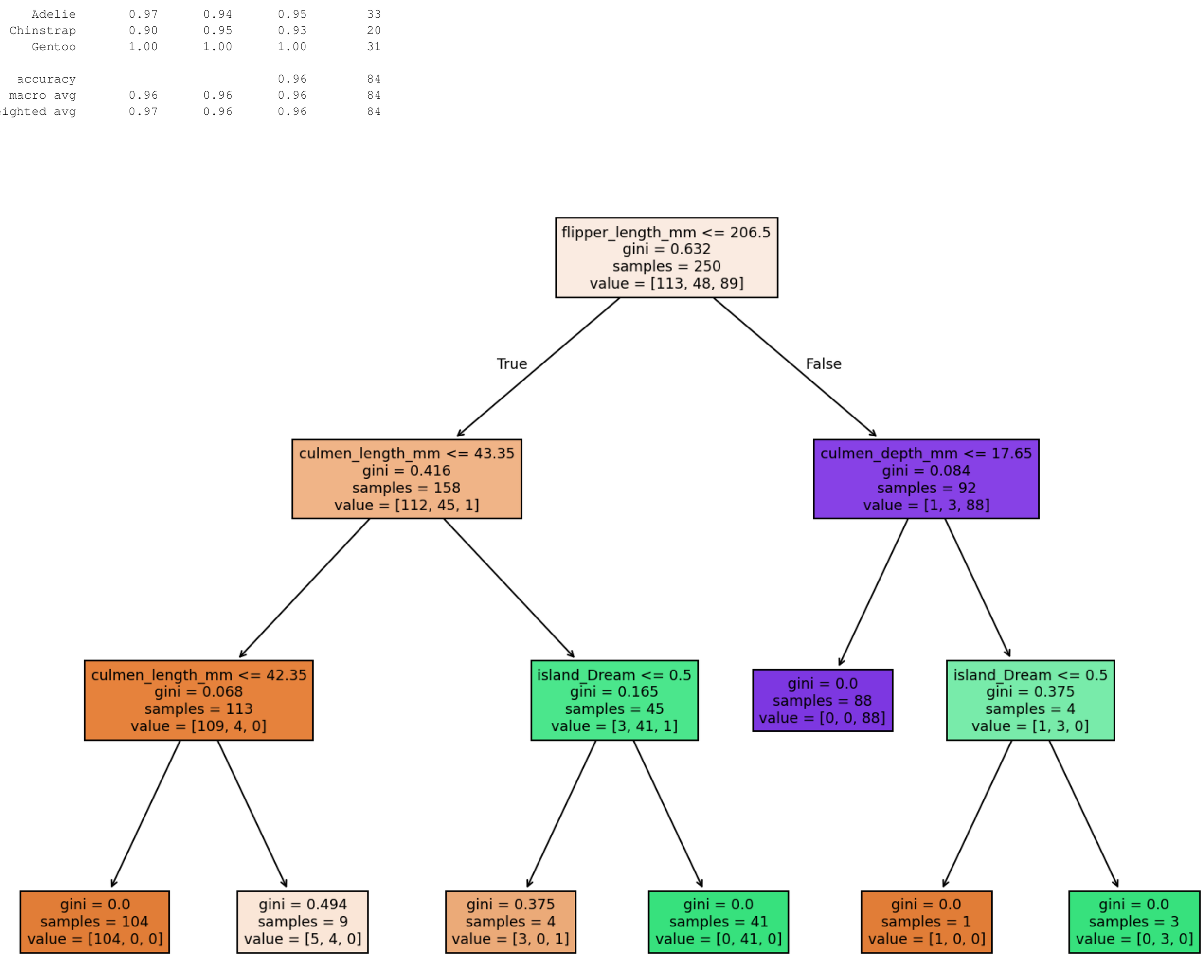
In [147…
```
# print classification report.
# plot decision tree

def tree_classification_report(model):
    model_prediction = model.predict(X_test)
    print(classification_report(y_test, model_prediction))
    plt.figure(figsize = (14, 10), dpi = 140)
    plot_tree(model, feature_names=X.columns, filled = True, fontsize=9);

tree_classification_report(model);
```

```
              precision    recall  f1-score   support

      Adelie       0.97      0.94      0.95        33
   Chinstrap       0.90      0.95      0.93        20
      Gentoo       1.00      1.00      1.00        31

    accuracy                           0.96        84
   macro avg       0.96      0.96      0.96        84
weighted avg       0.97      0.96      0.96        84
```



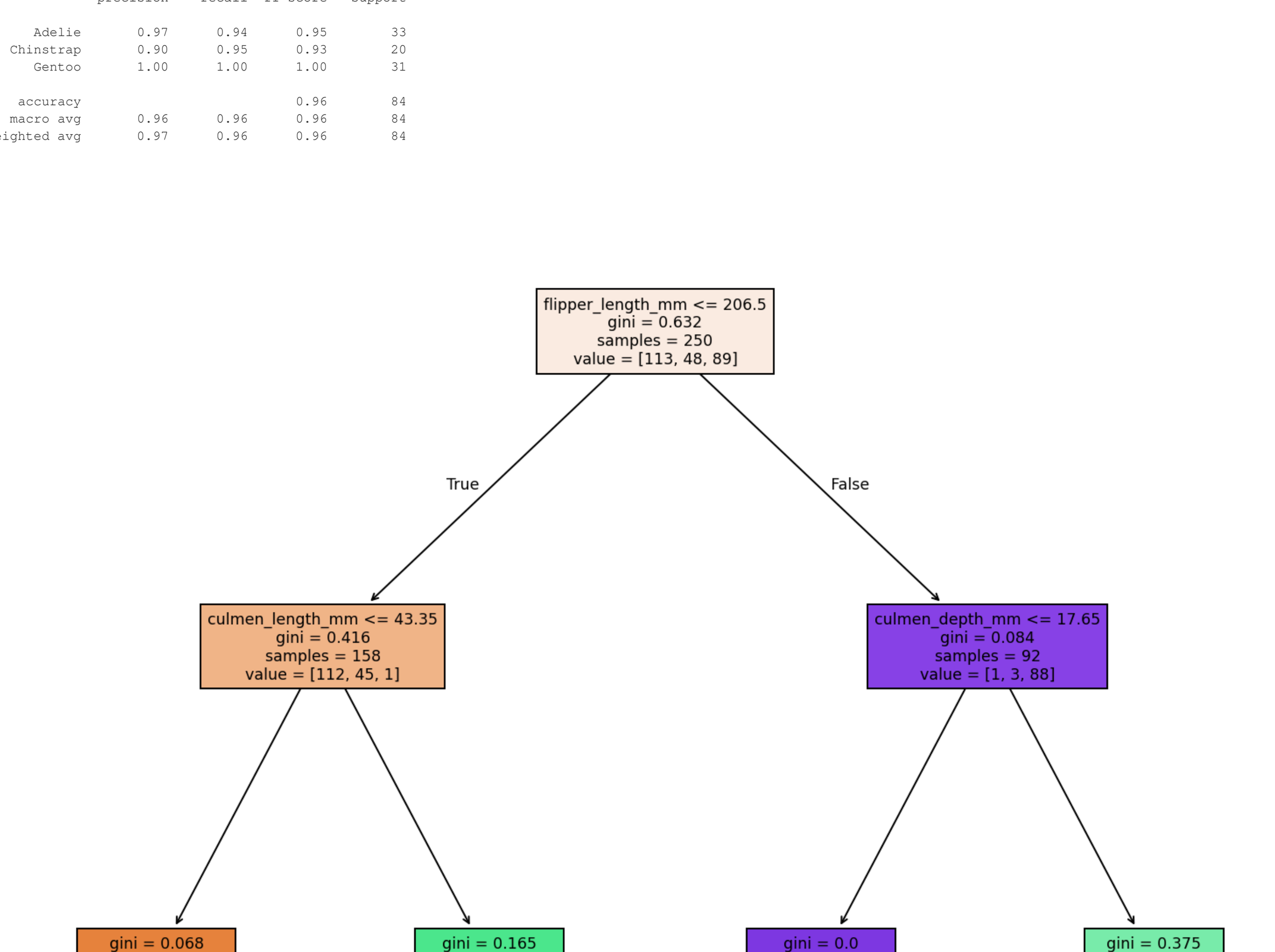
In [149…
```
# Make 3 splitting levels of decision tree

three_decision_tree_levels = DecisionTreeClassifier(max_depth= 3)

# fit model
three_decision_tree_levels.fit(X_train, y_train)

# fit report
tree_classification_report(three_decision_tree_levels)
```

```
              precision    recall  f1-score   support

      Adelie       0.97      0.94      0.95        33
   Chinstrap       0.90      0.95      0.93        20
      Gentoo       1.00      1.00      1.00        31

    accuracy                           0.96        84
   macro avg       0.96      0.96      0.96        84
weighted avg       0.97      0.96      0.96        84
```



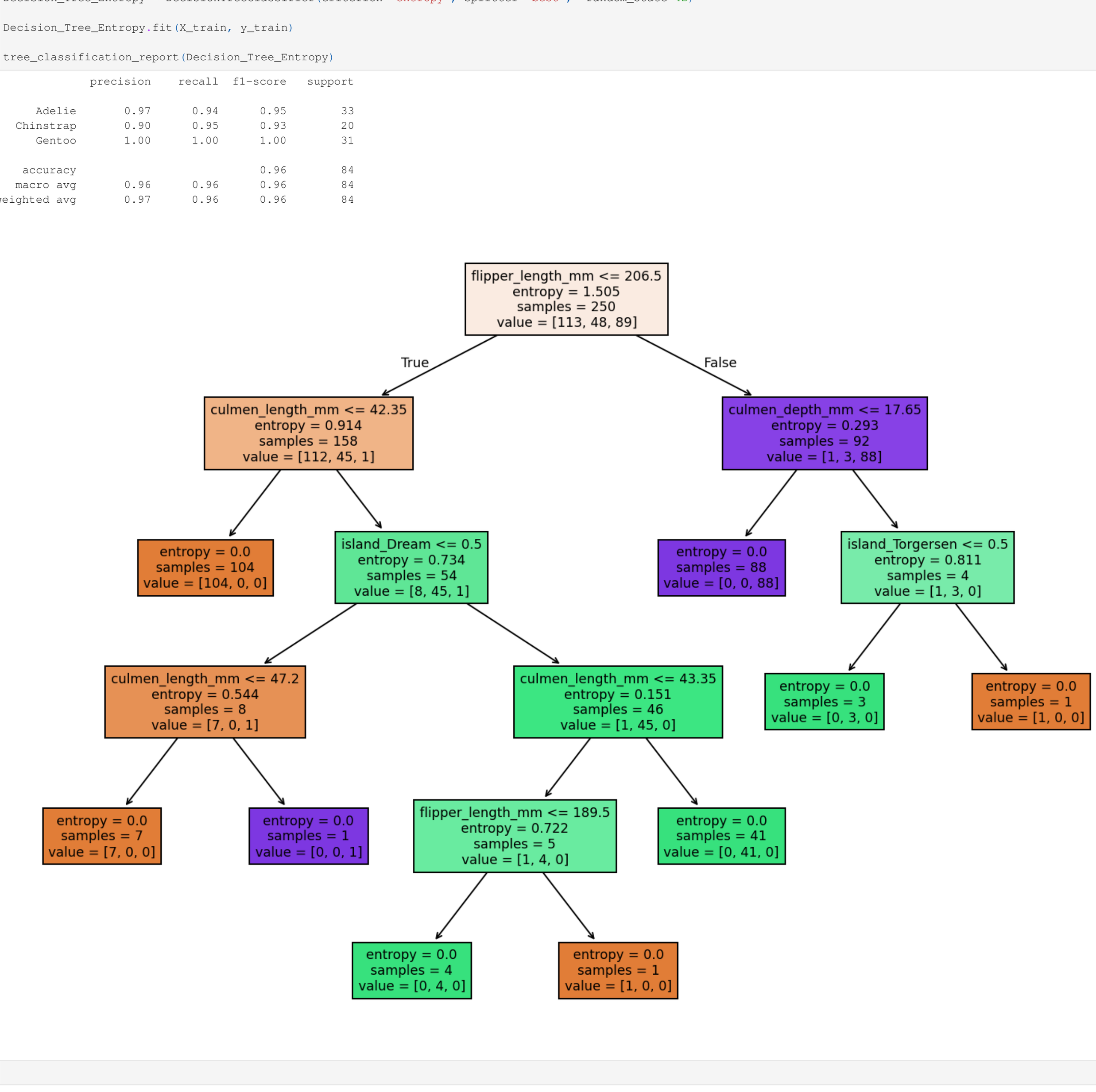
In [151…
```
# See maximum leaf tree = 4
maximum_leaf_tree_node = DecisionTreeClassifier(max_leaf_nodes=4)
maximum_leaf_tree_node.fit(X_train, y_train)
tree_classification_report(maximum_leaf_tree_node)
```

```
              precision    recall  f1-score   support

      Adelie       0.97      0.94      0.95        33
   Chinstrap       0.90      0.95      0.93        20
      Gentoo       1.00      1.00      1.00        31

    accuracy                           0.96        84
   macro avg       0.96      0.96      0.96        84
weighted avg       0.97      0.96      0.96        84
```



In [153…
```
# Find some information about entropy in decision tree

Decision_Tree_Entropy = DecisionTreeClassifier(criterion="entropy", splitter="best",  random_state=42)

Decision_Tree_Entropy.fit(X_train, y_train)

tree_classification_report(Decision_Tree_Entropy)
```

```
              precision    recall  f1-score   support

      Adelie       0.97      0.94      0.95        33
   Chinstrap       0.90      0.95      0.93        20
      Gentoo       1.00      1.00      1.00        31

    accuracy                           0.96        84
   macro avg       0.96      0.96      0.96        84
weighted avg       0.97      0.96      0.96        84
```



In [ ]: