Aadarsh Goyal ¶

MIS: 111915001

--

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv("penguins_size.csv")
df.head()
```

Out[2]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0

In [3]:

```
print(df.shape)
df.isna().sum()
```

```
(344, 7)
```

Out[3]:

```
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 [4]:

```
df = df.dropna()
df.shape
```

Out[4]:

```
(334, 7)
```

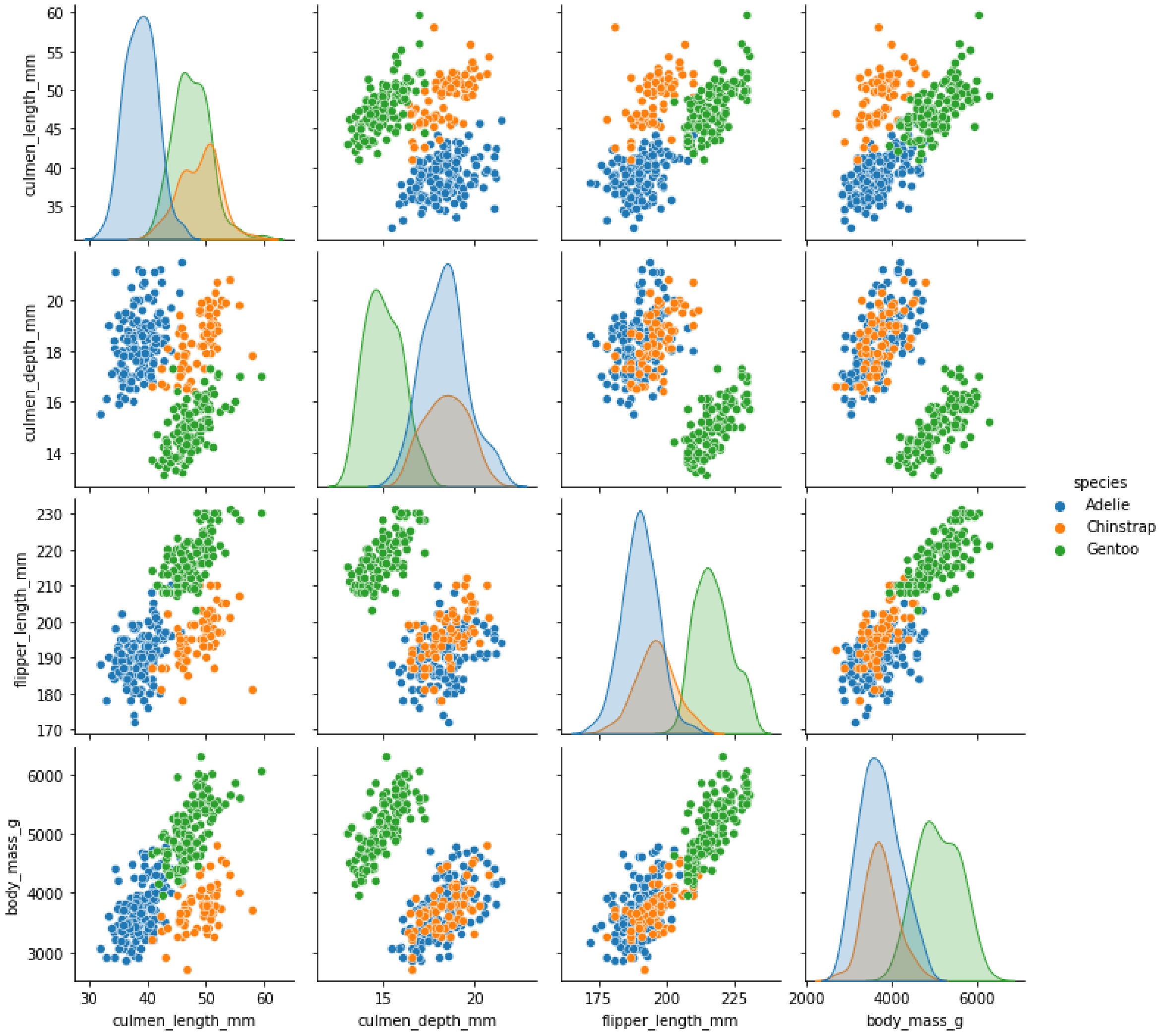
Data Visualization

In [5]:

```
sns.pairplot(data = df, hue = 'species')
```

Out[5]:

```
<seaborn.axisgrid.PairGrid at 0x26bffcb4d08>
```



In [6]:

```
X = pd.get_dummies(df.drop('species',axis=1),drop_first=True)
X = X.drop("sex_FEMALE", axis = 1)
y = df['species']
```

Test and Train splitting

In [7]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Random Forest Classification

In [8]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [9]:

```
model = RandomForestClassifier(n_estimators=10,max_features='auto',random_state=0)
model.fit(X_train,y_train)
```

Out[9]:

```
RandomForestClassifier(n_estimators=10, random_state=0)
```

In [10]:

```
prediction = model.predict(X_test)
```

Evaluate the model

In [11]:

```
from sklearn.metrics import confusion_matrix,classification_report,plot_confusion_matrix,ac

confusion_matrix(y_test,prediction)
```

Out[11]:

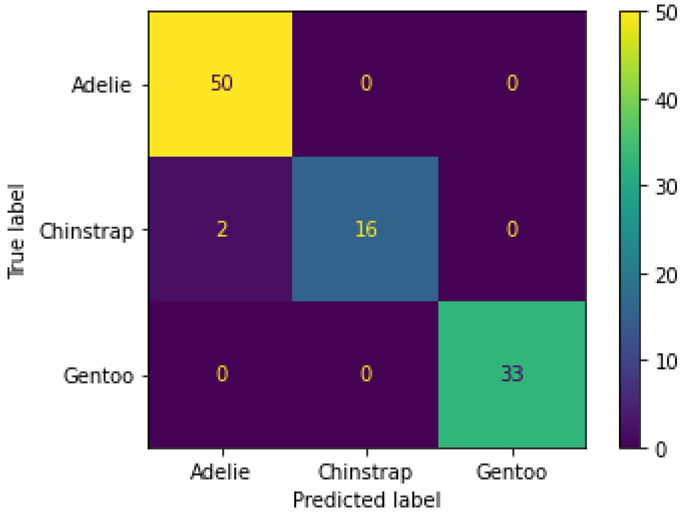
```
array([[50,  0,  0],
       [ 2, 16,  0],
       [ 0,  0, 33]], dtype=int64)
```

In [12]:

```
plot_confusion_matrix(model,X_test,y_test)
```

Out[12]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x26b82ce8
ec8>
```

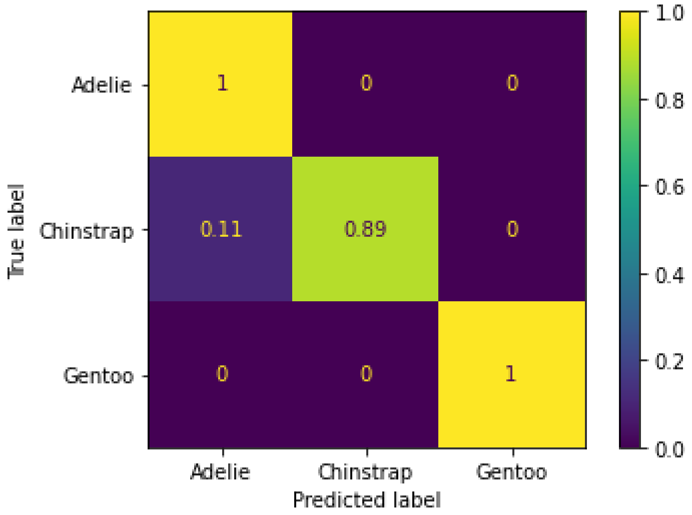


In [13]:

```
plot_confusion_matrix(model,X_test,y_test,normalize='true')
```

Out[13]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x26b82cf0
dc8>
```



In [14]:

```
print(classification_report(y_test,prediction))
```

```
              precision    recall  f1-score   support

      Adelie       0.96      1.00      0.98        50
   Chinstrap       1.00      0.89      0.94        18
      Gentoo       1.00      1.00      1.00        33

    accuracy                           0.98       101
   macro avg       0.99      0.96      0.97       101
weighted avg       0.98      0.98      0.98       101
```

In [ ]: