Data Analytics I| Assignment 4

Author: Prateek Pandey
ID: 2022201035

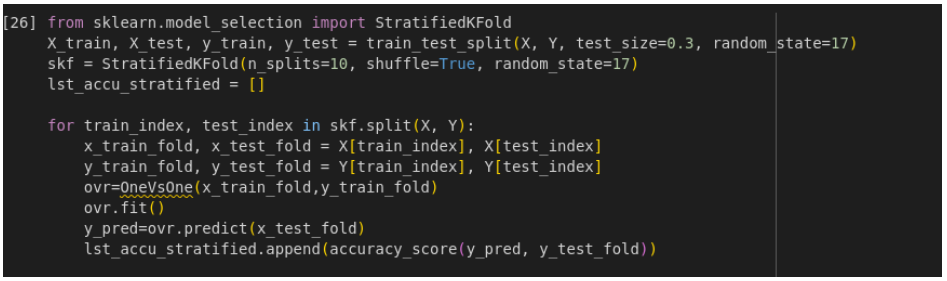
October 30, 2023

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

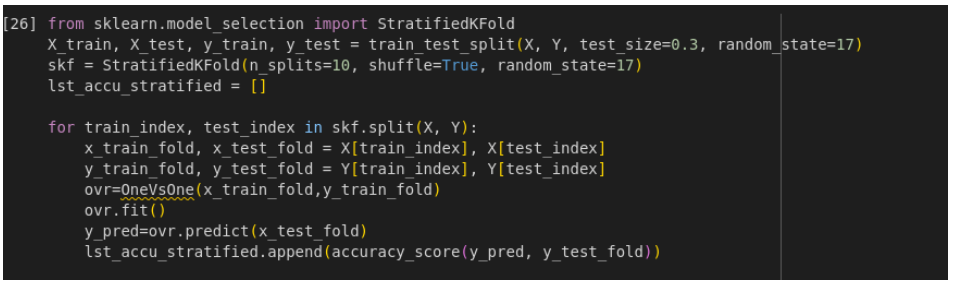
HYDERABAD

Contents

- **There are very less samples in the above dataset , How do you deal with that?**
  - The dataset I'm working with contains a limited number of samples. In order to preprocess this data, I transformed textual labels into a one-hot encoding format. This conversion resulted in a sparse matrix with numerous columns, reflecting the categorical values as binary features.
  - Additionally, I standardized the numeric columns to follow a standard normal distribution. This scaling process ensures that all numerical features contribute uniformly to the analysis, preparing them for further processing.
  - Due to the high dimensionality caused by the sparsity from one-hot encoding, I employed Principal Component Analysis (PCA). PCA allowed me to address both the sparsity and the high number of dimensions. By reducing the dimensionality, I was able to obtain a denser representation of the data, condensing it down to 7 columns, while still retaining crucial information.
  - Thus only relevant data remained
  - While Evaluating Model we used Stratified K-Fold Cross Validation, It maximizes the use of the available data by iteratively using it for both training and validation, reducing the variance in model performance evaluation and providing a more reliable estimate of how the model will perform on unseen data.

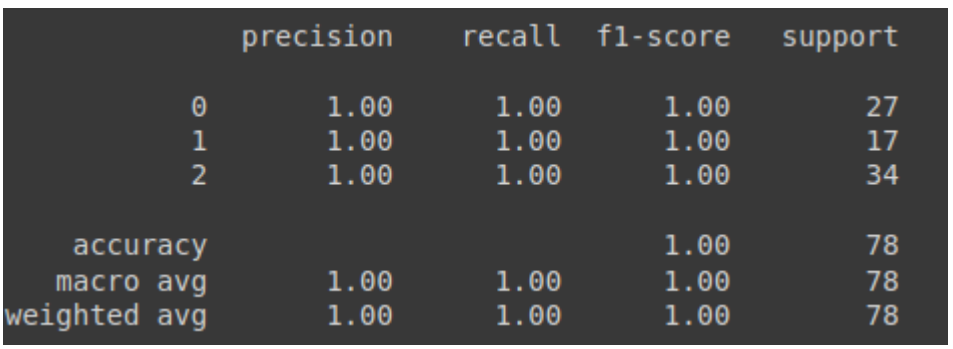


```
[26] from sklearn.model_selection import StratifiedKFold
     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=17)
     skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=17)
     lst_accu_stratified = []

     for train_index, test_index in skf.split(X, Y):
         x_train_fold, x_test_fold = X[train_index], X[test_index]
         y_train_fold, y_test_fold = Y[train_index], Y[test_index]
         ovr=OneVsOne(x_train_fold,y_train_fold)
         ovr.fit()
         y_pred=ovr.predict(x_test_fold)
         lst_accu_stratified.append(accuracy_score(y_pred, y_test_fold))
```



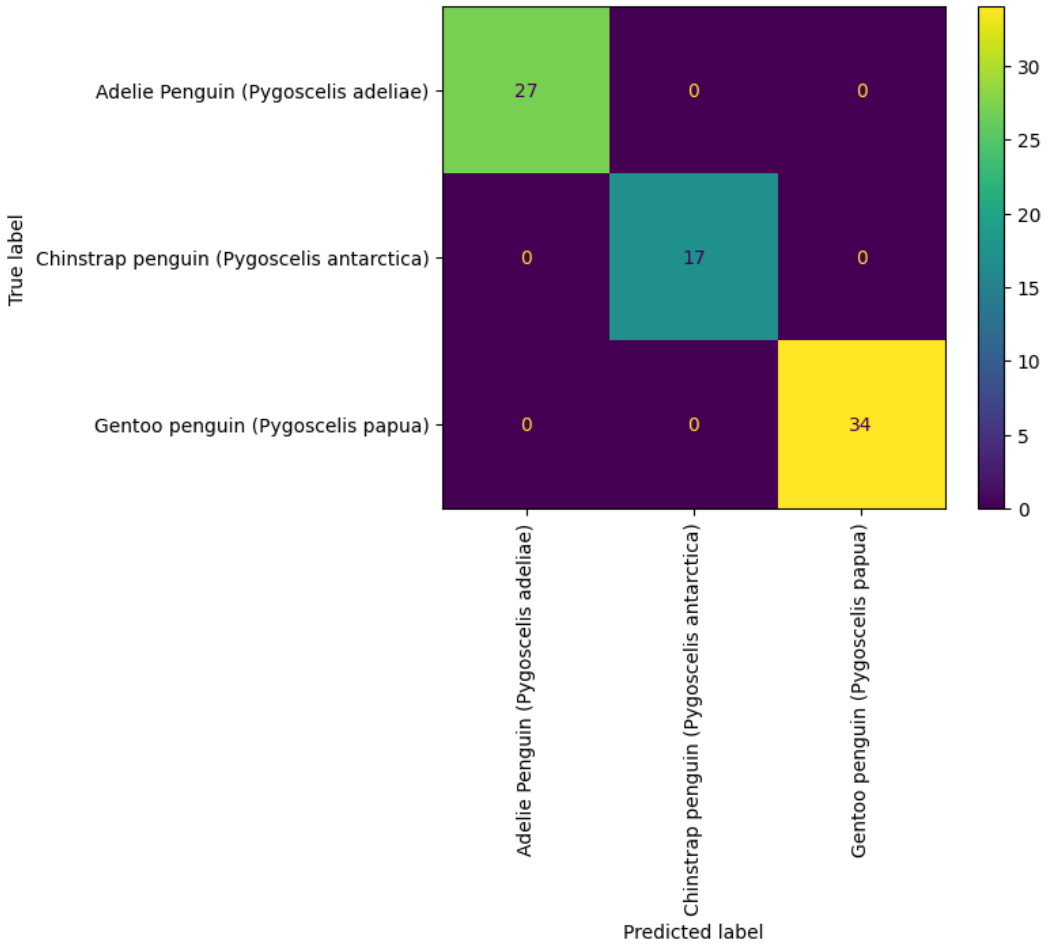
```
[26] from sklearn.model_selection import StratifiedKFold
     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=17)
     skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=17)
     lst_accu_stratified = []

     for train_index, test_index in skf.split(X, Y):
         x_train_fold, x_test_fold = X[train_index], X[test_index]
         y_train_fold, y_test_fold = Y[train_index], Y[test_index]
         ovr=OneVsOne(x_train_fold,y_train_fold)
         ovr.fit()
         y_pred=ovr.predict(x_test_fold)
         lst_accu_stratified.append(accuracy_score(y_pred, y_test_fold))
```

- **Plot the confusion matrix,and include the precision, recall, f1-score metrics in the report .**
  - **One Vs All**
    - **Precision Recall Accuracy**



	precision	recall	f1-score	support
0	1.00	1.00	1.00	27
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	34
accuracy			1.00	78
macro avg	1.00	1.00	1.00	78
weighted avg	1.00	1.00	1.00	78

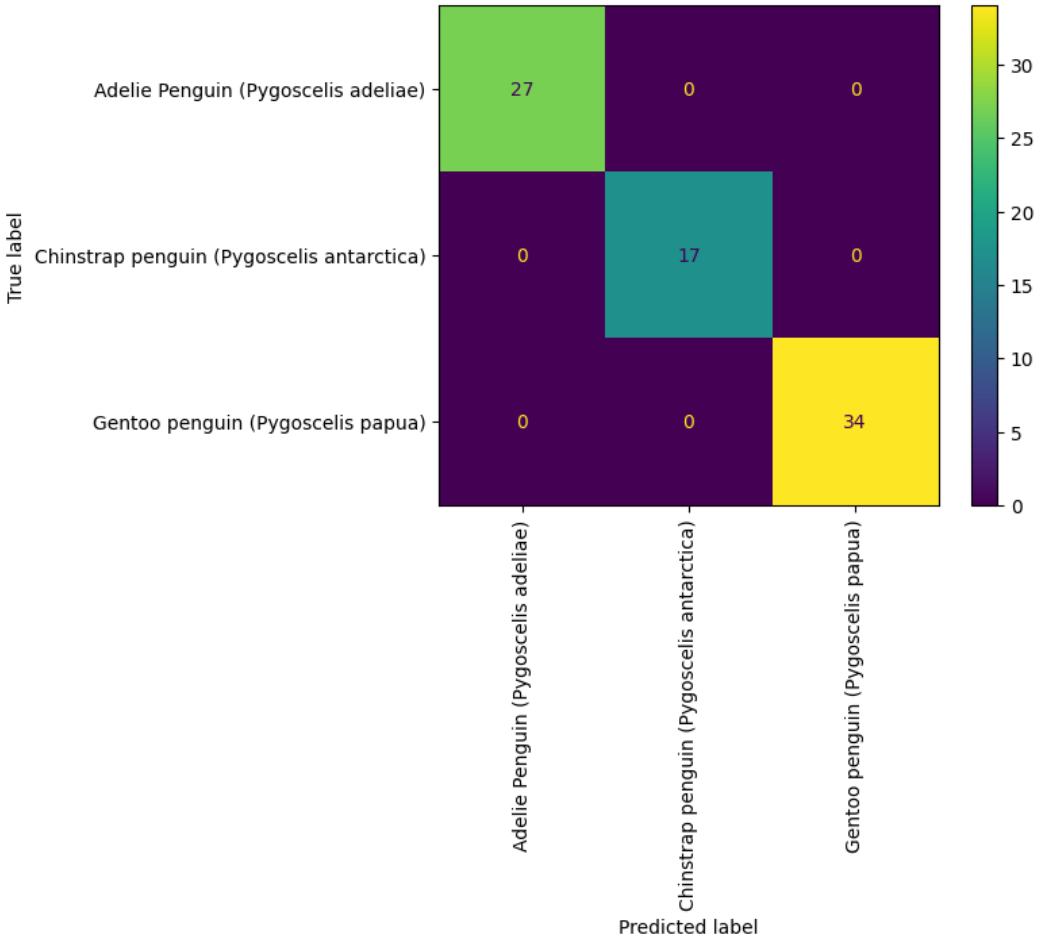
- **Confusion Matrix**



- **One vs One**
  - **Precision Recall Accuracy**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	27
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	34
accuracy			1.00	78
macro avg	1.00	1.00	1.00	78
weighted avg	1.00	1.00	1.00	78

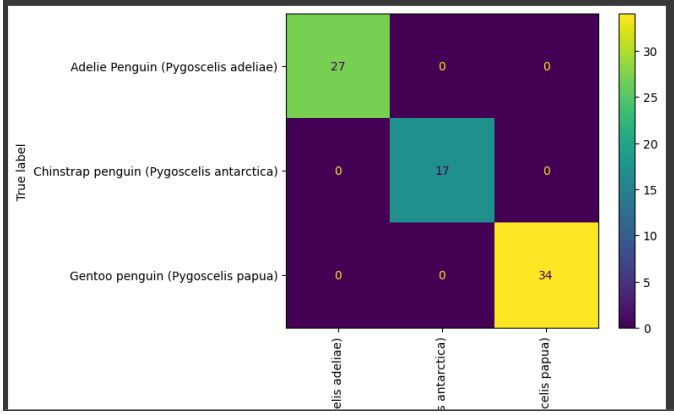
  - **Confusion Matrix**



- **Compare the results obtained for one-vs-one and one-vs-all(which according to you performs better for the above dataset)**
  - **One Vs One**

```
ovo=OneVsOne(X_train,y_train)
ovo.fit()
y_pred=ovo.predict(X_test)
print(classification_report(y_pred,y_test))
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

disp = ConfusionMatrixDisplay.from_predictions(encoder.inverse_transform(y_pred),encoder.inverse_transform(y_test),display_labels=encoder.classes_,xticks_rotation="vertical")

plt.show()
```

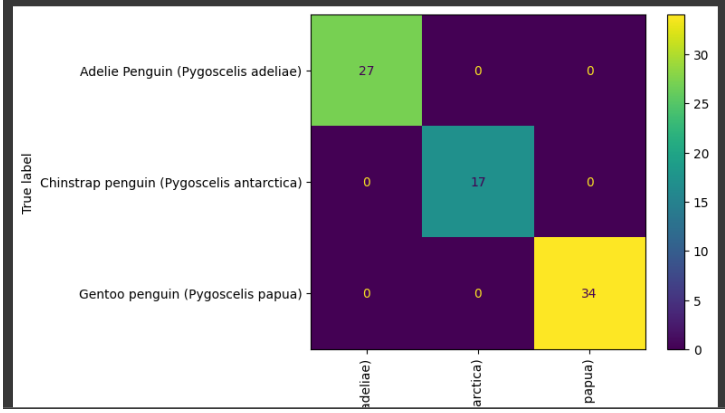
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00        17
           2       1.00      1.00      1.00        34

    accuracy                           1.00        78
   macro avg       1.00      1.00      1.00        78
weighted avg       1.00      1.00      1.00        78
```



  - **One Vs Rest**

```
from sklearn.model_selection import StratifiedKFold
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=17)
ovr=OneVsRest(X_train,y_train)
ovr.fit()
y_pred=ovr.predict(X_test)
print(classification_report(y_pred,y_test))
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

disp = ConfusionMatrixDisplay.from_predictions(encoder.inverse_transform(y_pred),encoder.inverse_transform(y_test),display_labels=encoder.classes_,xticks_rotation="ve

plt.show()
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00        17
           2       1.00      1.00      1.00        34

    accuracy                           1.00        78
   macro avg       1.00      1.00      1.00        78
weighted avg       1.00      1.00      1.00        78
```



- Since both the models are showing 1.0 accuracy at a split of 0.7-0.3 both the models are accurate

Random Forest

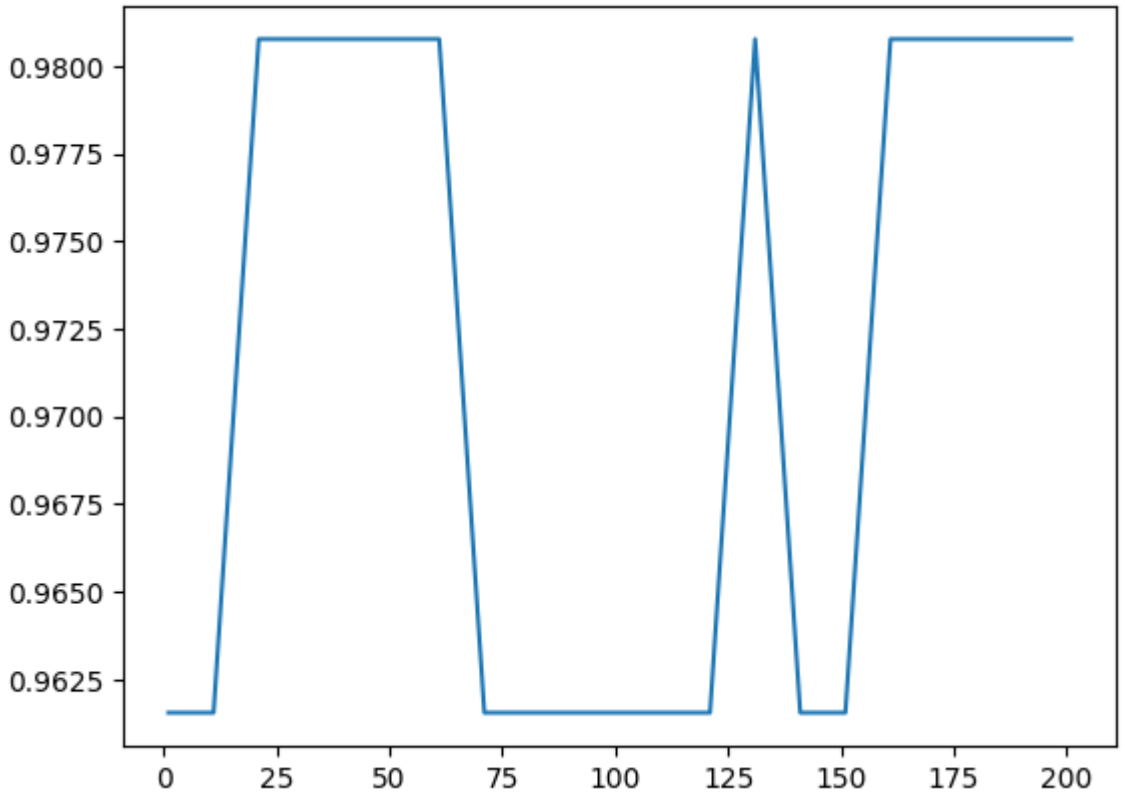
- Predicted vs Actual Labels

```
[{'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Chinstrap penguin (Pygoscelis antarctica)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Chinstrap penguin (Pygoscelis antarctica)',
  'predicted': 'Chinstrap penguin (Pygoscelis antarctica)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Chinstrap penguin (Pygoscelis antarctica)',
  'predicted': 'Chinstrap penguin (Pygoscelis antarctica)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Chinstrap penguin (Pygoscelis antarctica)',
  'predicted': 'Chinstrap penguin (Pygoscelis antarctica)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Adelie Penguin (Pygoscelis adeliae)',
  'predicted': 'Adelie Penguin (Pygoscelis adeliae)'},
 {'actual': 'Gentoo penguin (Pygoscelis papua)',
  'predicted': 'Gentoo penguin (Pygoscelis papua)'},
```

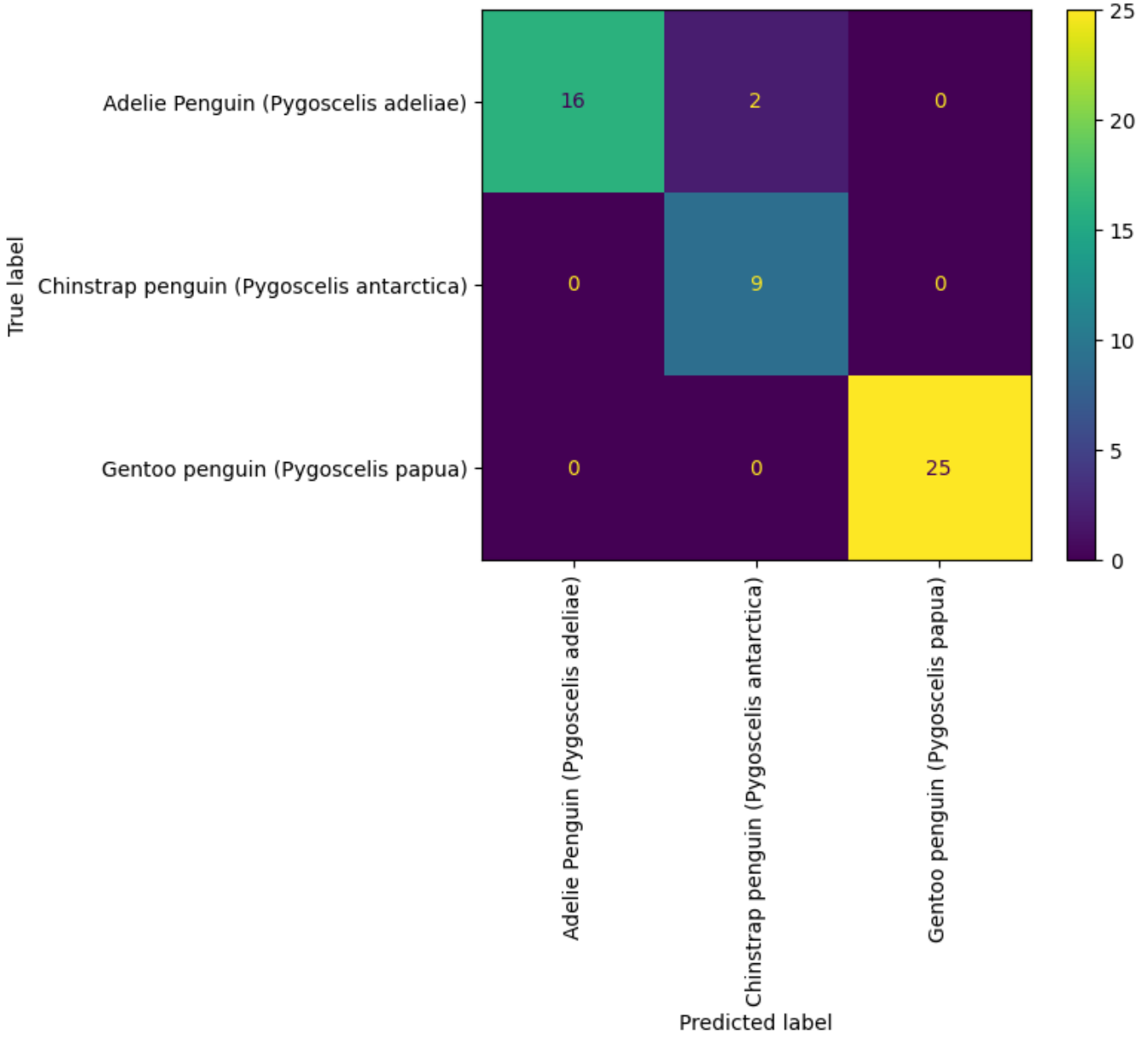

Accuracy Precision and Recall

	precision	recall	f1-score	support
Adelie Penguin (Pygoscelis adeliae)	1.00	0.89	0.94	18
Chinstrap penguin (Pygoscelis antarctica)	0.82	1.00	0.90	9
Gentoo penguin (Pygoscelis papua)	1.00	1.00	1.00	25
accuracy			0.96	52
macro avg	0.94	0.96	0.95	52
weighted avg	0.97	0.96	0.96	52

- Accuracy Score over no. of trees 1-201 at interval of 10



Confusion Matrix


Adelie Penguin (Pygoscelis adeliae)
Chinstrap penguin (Pygoscelis antarctica)
Gentoo penguin (Pygoscelis papua)
True label
16
2
0
0
9
0
0
0
25
Adelie Penguin (Pygoscelis adeliae)
Chinstrap penguin (Pygoscelis antarctica)
Gentoo penguin (Pygoscelis papua)
Predicted label
25
20
15
10
5
0