CPC251 PROJECT

AZRI ZAMRUD BIN KIMIN (153507) , HAZIQ BIN HIZUL (152770) , MUHAMMAD ARMAND BIN MUHAMAD FAZLI (153857) , SYED MUHAMMAD HAIKAL BIN SYED HUSNI (153086)

The calls of frogs are a familiar noise from yards from backyards to the bush, but what is a frog call? From love songs to battle cries, frogs use vocal communication to find mates, fight over territories, and cry for help. Each frog species has a unique call, even though that call can differ place to place, just like human accents.

With the developments of predictive models, we can use multiple types of models to predict the species of each frogs or Anuran by its calls.

Aim : to build two effective predictive model that is able to accurately classify the Anuran species based on its calls

and compare its accuracy.

Dataset Description

This dataset was used in several classifications tasks related to the challenge of anuran or frogs species recognition through their calls. It is a multilabel dataset with three columns of labels. This dataset was created segmenting 60 audio recordings belonging to 4 different families, 8 genus , and 10 species in which each audio corresponds to one specimen (an individual frog/anuran) . The record ID is also included as an extra column . After segmentation, there are 7165 syllables which will become the instances for train and test the classifiers. Some species are from the campus of Federal University of Amazonas, Manaus, others from Mata Atlântica, Brazil, and one of them from Córdoba, Argentina. The attributes are acoustic features extracted from the syllables of anuran calls, including the family, the genus, and the species label. Mel-frequencies cepstral coefficient (MFCCs) are coefficients that collectively make up an Mel-frequencies cepstrum (MFC). Due to each syllables having different length , every row was normalized . There are in total 22 attributes of MFCCs.

	MFCCs_1	MFCCs_2	MFCCs_3	MFCCs_4	MFCCs_5	MFCCs_6	MFCCs_7	MFCCs_8	MFCCs_9	MFCCs_10	...
0	1.0	0.152936	-0.105586	0.200722	0.317201	0.260764	0.100945	-0.150063	-0.171128	0.124676	...
1	1.0	0.171534	-0.098975	0.268425	0.338672	0.268353	0.060835	-0.222475	-0.207693	0.170883	...
2	1.0	0.152317	-0.082973	0.287128	0.276014	0.189867	0.008714	-0.242234	-0.219153	0.232538	...
3	1.0	0.224392	0.118985	0.329432	0.372088	0.361005	0.015501	-0.194347	-0.098181	0.270375	...
4	1.0	0.087817	-0.068345	0.306967	0.330923	0.249144	0.006884	-0.265423	-0.172700	0.266434	...

Table 1 : Dataset of Anuran

MFCCs_14	MFCCs_15	MFCCs_16	MFCCs_17	MFCCs_18	MFCCs_19	MFCCs_20	MFCCs_21	MFCCs_22	Species
0.082245	0.135752	-0.024017	-0.108351	-0.077623	-0.009568	0.057684	0.118680	0.014038	AdenomeraAndre
0.022786	0.163320	0.012022	-0.090974	-0.056510	-0.035303	0.020140	0.082263	0.029056	AdenomeraAndre
0.050791	0.207338	0.083536	-0.050691	-0.023590	-0.066722	-0.025083	0.099108	0.077162	AdenomeraAndre
-0.011567	0.100413	-0.050224	-0.136009	-0.177037	-0.130498	-0.054766	-0.018691	0.023954	AdenomeraAndre
0.037439	0.219153	0.062837	-0.048885	-0.053074	-0.088550	-0.031346	0.108610	0.079244	AdenomeraAndre

Table 1 : Dataset of Anuran

Data Analysis

SCATTER PLOT

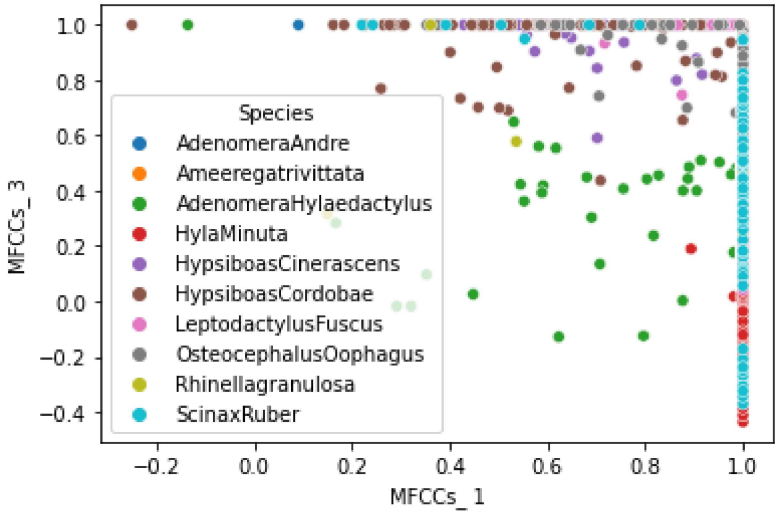


Figure 1 : MFCCs_1 against MFCCs_3

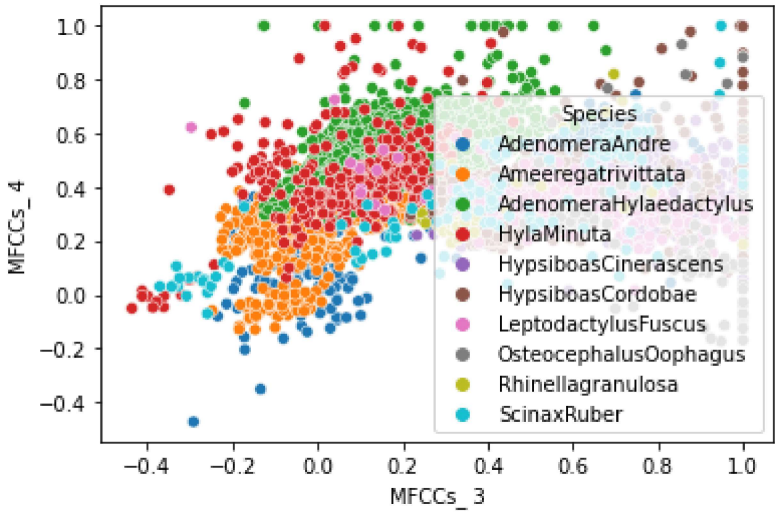


Figure 2 : MFCCs_3 against MFCCs_4

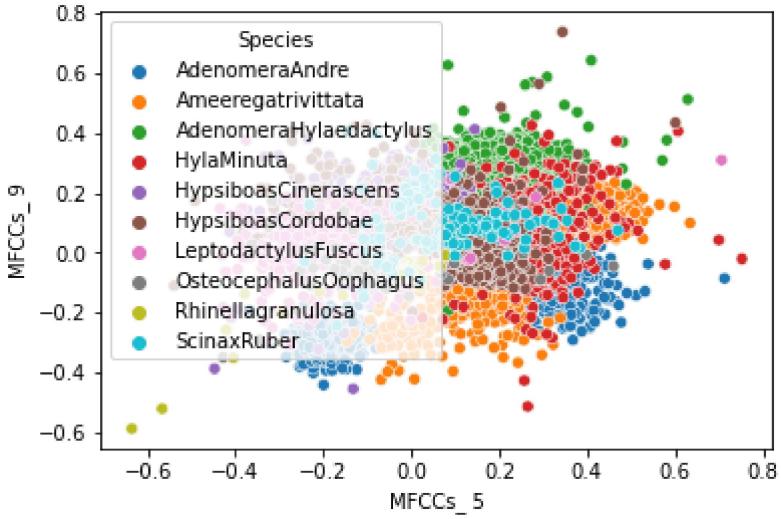


Figure 3 : MFCCs_5 against MFCCs_9

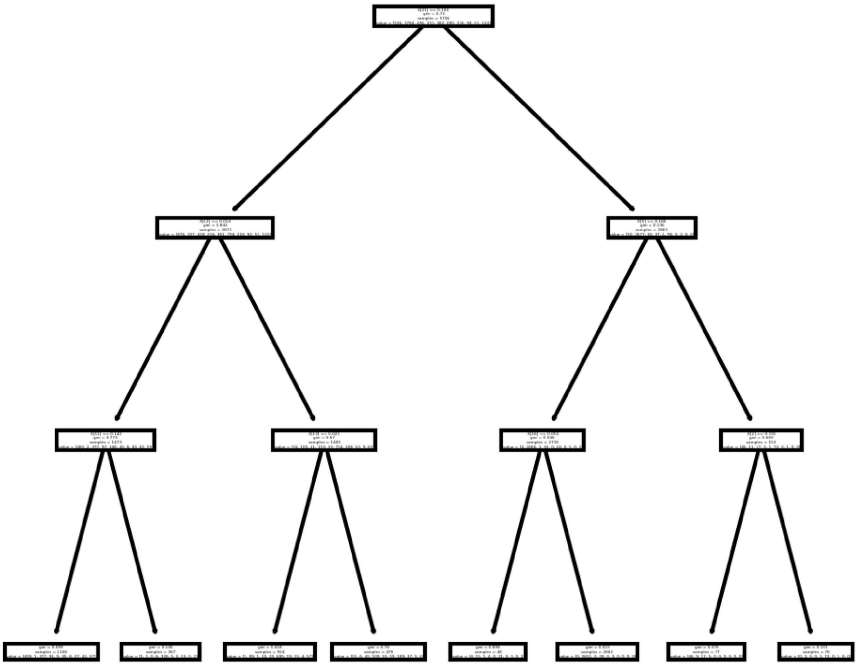


Figure 4 : Decision Tree Diagram of the Model

Data Modelling

Two predictive models are built using Decision Tree and K-Nearest Neighbor (KNN) algorithm.

Algorithm	Value/Statistics
Decision Tree	**Criteria** – Gini **Max Depth** – 3 **Min samples In leaf** - 1
K Nearest Neighbor (KNN)	**K** - 5

Table 2 : Parameters of the predictive models.

Result of the classification of each predictive models are given below.

DECISION TREE

	precision	recall	f1-score	support
AdenomeraAndre	0.92	0.88	0.90	139
AdenomeraHylaedactylus	0.99	0.99	0.99	696
Ameeregatrivittata	0.89	0.96	0.92	95
HylaMinuta	0.73	0.82	0.77	68
HypsiboasCinerascens	0.92	0.94	0.93	98
HypsiboasCordobae	0.92	0.92	0.92	226
LeptodactylusFuscus	0.87	0.87	0.87	53
OsteocephalusOophagus	0.71	0.50	0.59	20
Rhinellagranulosa	0.86	0.92	0.89	13
ScinaxRuber	0.93	0.84	0.88	31
accuracy			0.94	1439
macro avg	0.87	0.86	0.87	1439
weighted avg	0.94	0.94	0.94	1439

Figure 5: Results of classification using Decision Tree model.

```
0.9388464211257818
[[123   0   5   7   1   1   1   1   0   0]
 [  1 686   0   7   0   2   0   0   0   0]
 [  1   0  91   2   1   0   0   0   0   0]
 [  4   3   3  56   0   1   1   0   0   0]
 [  0   0   0   1  92   4   0   1   0   0]
 [  1   1   2   4   2 209   3   1   1   2]
 [  1   0   0   0   0   5  46   1   0   0]
 [  1   0   0   0   4   3   2  10   0   0]
 [  0   0   1   0   0   0   0   0  12   0]
 [  2   0   0   0   0   2   0   0   1  26]]
```

Figure 6: Confusion Matrix for Decision Tree model.

K-NEAREST NEIGHBOR

	precision	recall	f1-score	support
AdenomeraAndre	0.96	0.97	0.97	146
AdenomeraHylaedactylus	1.00	1.00	1.00	694
Ameeregatrivittata	0.99	0.99	0.99	106
HylaMinuta	0.96	0.91	0.94	57
HypsiboasCinerascens	0.94	0.99	0.96	90
HypsiboasCordobae	0.99	0.99	0.99	231
LeptodactylusFuscus	0.96	1.00	0.98	54
OsteocephalusOophagus	1.00	0.70	0.82	20
Rhinellagranulosa	1.00	0.88	0.94	17
ScinaxRuber	1.00	0.96	0.98	24
accuracy			0.98	1439
macro avg	0.98	0.94	0.96	1439
weighted avg	0.98	0.98	0.98	1439

Figure 7: Results of classification using K-Nearest Neighbor model.

```
[[114   4  20   3   1   2   2   0   0   0]
 [  3 680   3   6   0   1   1   0   0   0]
 [ 12   8  79   5   0   2   0   0   0   0]
 [  7  27   5  18   0   0   0   0   0   0]
 [  0   1   0   0  84   4   1   0   0   0]
 [  2   2   0   0  13 209   4   1   0   0]
 [  1   2   0   2   0   8  40   1   0   0]
 [  1   0   0   0   3   5   0  11   0   0]
 [  0   0   1   0   1   3   0   0  12   0]
 [  0   0   1   1   1   5   4   0   0  12]]
```

Figure 8: Confusion Matrix for K-Nearest Neighbor model.

Conclusion

- Based on the classification results, we can see that the average precision of the K-Nearest Neighbor model is higher than the precision of the Decision Tree model.
- K-Nearest Neighbor also has the greater average value of recall.
- Thus, it can be deduced that in this case. K-Nearest Neighbor predictive model might be the better algorithm to use.

Data Modelling

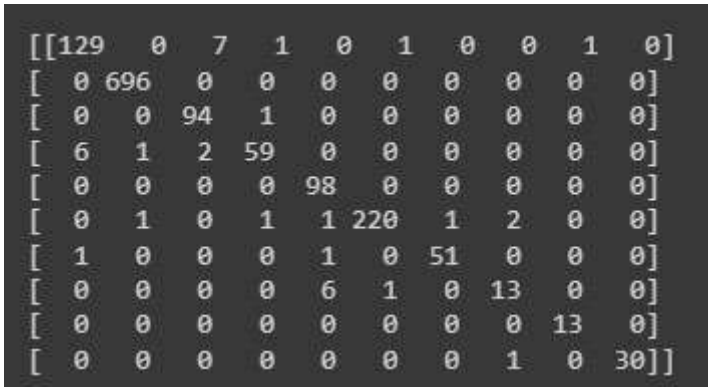
Two predictive models are built using Neural Network and Extreme Gradient Boosting (XGBoost)

Result of the classification of each predictive models are given below.

NEURAL NETWORK

	precision	recall	f1-score	support
0	0.95	0.93	0.94	139
1	1.00	1.00	1.00	696
2	0.91	0.99	0.95	95
3	0.95	0.87	0.91	68
4	0.92	1.00	0.96	98
5	0.99	0.97	0.98	226
6	0.98	0.96	0.97	53
7	0.81	0.65	0.72	20
8	0.93	1.00	0.96	13
9	1.00	0.97	0.98	31
accuracy			0.97	1439
macro avg	0.94	0.93	0.94	1439
weighted avg	0.98	0.97	0.97	1439

Figure 9 : Classification for Neural Network machine learning algorithm



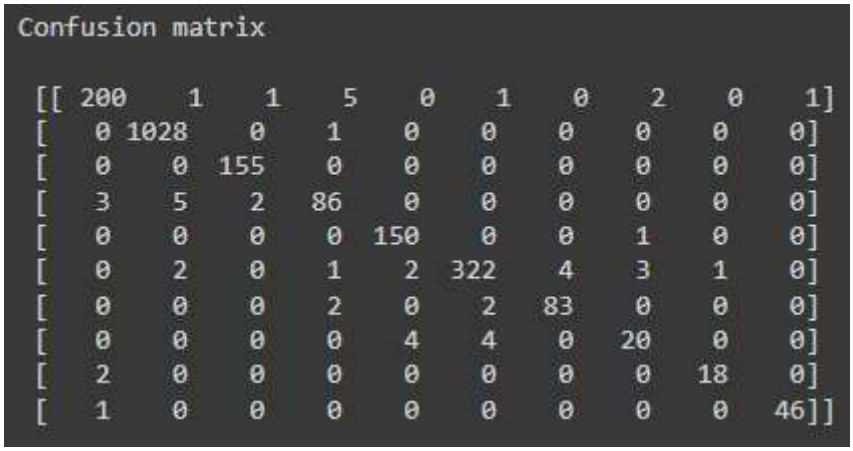
```
[[129   0   7   1   0   1   0   0   1   0]
 [  0 696   0   0   0   0   0   0   0   0]
 [  0   0  94   1   0   0   0   0   0   0]
 [  6   1   2  59   0   0   0   0   0   0]
 [  0   0   0   0  98   0   0   0   0   0]
 [  0   1   0   1   1 220   1   2   0   0]
 [  1   0   0   0   1   0  51   0   0   0]
 [  0   0   0   0   6   1   0  13   0   0]
 [  0   0   0   0   0   0   0   0  13   0]
 [  0   0   0   0   0   0   0   1   0  30]]
```

Figure 10 : Confusion Matrix for Neural Network machine learning algorithm

EXTREME GRADIENT BOOSTING

	precision	recall	f1-score	support
0	0.97	0.95	0.96	211
1	0.99	1.00	1.00	1029
2	0.98	1.00	0.99	155
3	0.91	0.90	0.90	96
4	0.96	0.99	0.98	151
5	0.98	0.96	0.97	335
6	0.95	0.95	0.95	87
7	0.77	0.71	0.74	28
8	0.95	0.90	0.92	20
9	0.98	0.98	0.98	47
accuracy			0.98	2159
macro avg	0.94	0.93	0.94	2159
weighted avg	0.98	0.98	0.98	2159

Figure 11 : Classification for XGBoost machine learning algorithm



```
Confusion matrix

 [[ 200    1    1    5    0    1    0    2    0    1]
 [   0 1028    0    1    0    0    0    0    0    0]
 [   0    0  155    0    0    0    0    0    0    0]
 [   3    5    2   86    0    0    0    0    0    0]
 [   0    0    0    0  150    0    0    1    0    0]
 [   0    2    0    1    2  322    4    3    1    0]
 [   0    0    0    2    0    2   83    0    0    0]
 [   0    0    0    0    4    4    0   20    0    0]
 [   2    0    0    0    0    0    0    0   18    0]
 [   1    0    0    0    0    0    0    0    0   46]]
```

Figure 12 : Confusion Matrix for XGBoost machine learning algorithm

Conclusion

- Based on the classification results, we can see that the average precision of the Extreme Gradient Boosting machine learning algorithm is more or less the same as the Neural Network Algorithm
- Extreme Gradient Boosting also has the greater average value of recall.
- Thus, it can be deduced that in this case, Extreme Gradient Boosting machine learning algorithm might be the same ML algorithm to use compared to Neural Network because of the overall more or less same score of average for accuracy, precision, recall and f1-Score.
- Hence, Both is good for ML algorithm