Machine Learning Project

Introduction to Machine Learning

ARM 210

Submitted by:
Name- Satyam Kumar Jha
Branch- AIML B2
Enrolment no- 20219051622

Submitted To:
Dr. Amit Choudhary
Assistant Professor
USAR, GGSIPU

University School of Automation and Robotics
East Campus, GGSIP University
Surajmal Vihar, New Delhi - 110092

PROJECT INFORMATION

Title of Project: ***From Ribbit to Recognition***
Student Name: ***Satyam Kumar Jha***
Enrollment Number: ***20219051622***
Email ID: satyam.k.jha19@gmail.com
Contact Number: +91-8595822518

From Ribbit to Recognition

From Ribbit to Recognition is any of the vertebrates of the order Anura, characterised by the absence of a tail and very long hind legs specialised for hopping: class Amphibia (amphibians), which is commonly known as the frogs and toads. Recently, recognition of anuran species through their calls has received a lot of attention because of its potential applicability in ecological studies.

However, most of the recorded anuran species are considered to be monotypic based on various research presented. Hence, the classification of numerous anuran species would be a challenge for researchers.

Aim: *To develop an interpretable and trustworthy predictive model that can classify various anuran species accurately and effectively.*

In this project, we are using the given dataset about the Anuran species, to create two predictive models, to test and predict based on the dataset.

Data Description

The Anuran Species dataset is the dataset assigned to me. Based on the two tables below, we can notice that the dataset contains 22 columns of features and a single column of the target. There is not a very unique name for all the features, they only differ in terms of the numbering, which represents different animals without clarifying what the actual animal is. The target is the 'Species', in which we are expected to get the outcome of different types of Anuran species (10 different species).

	MFCCs_ 1	MFCCs_ 2	MFCCs_ 3	MFCCs_ 4	MFCCs_ 5	MFCCs_ 6	MFCCs_ 7	MFCCs_ 8	MFCCs_ 9	MFCCs_10	MFCCs_11
0	1.0	0.152936	-0.105586	0.200722	0.317201	0.260764	0.100945	-0.150063	-0.171128	0.124676	0.188654
1	1.0	0.171534	-0.098975	0.268425	0.338672	0.268353	0.060835	-0.222475	-0.207693	0.170883	0.270958
2	1.0	0.152317	-0.082973	0.287128	0.276014	0.189867	0.008714	-0.242234	-0.219153	0.232538	0.266064
3	1.0	0.224392	0.118985	0.329432	0.372088	0.361005	0.015501	-0.194347	-0.098181	0.270375	0.267279
4	1.0	0.087817	-0.068345	0.306967	0.330923	0.249144	0.006884	-0.265423	-0.172700	0.266434	0.332695
5	1.0	0.099704	-0.033408	0.349895	0.344535	0.247569	0.022407	-0.213767	-0.127916	0.277353	0.309861
6	1.0	0.021676	-0.062075	0.318229	0.380439	0.179043	-0.041667	-0.252300	-0.167117	0.220027	0.260326
7	1.0	0.145130	-0.033660	0.284166	0.279537	0.175211	0.005791	-0.183329	-0.158483	0.192567	0.264184
8	1.0	0.271326	0.027777	0.375738	0.385432	0.272457	0.098192	-0.173730	-0.157857	0.207181	0.269932
9	1.0	0.120565	-0.107235	0.316555	0.364437	0.307757	0.025992	-0.294179	-0.223236	0.268435	0.367813

Table 1: Dataset sample 1

	MFCCs_12	MFCCs_13	MFCCs_14	MFCCs_15	MFCCs_16	MFCCs_17	MFCCs_18	MFCCs_19	MFCCs_20	MFCCs_21	MFCCs_22	Species
0	-0.075622	-0.156436	0.082245	0.135752	-0.024017	-0.108351	-0.077623	-0.009568	0.057684	0.118680	0.014038	AdenomeraAndre
1	-0.095004	-0.254341	0.022786	0.163320	0.012022	-0.090974	-0.056510	-0.035303	0.020140	0.082263	0.029056	AdenomeraAndre
2	-0.072827	-0.237384	0.050791	0.207338	0.083536	-0.050691	-0.023590	-0.066722	-0.025083	0.099108	0.077162	AdenomeraAndre
3	-0.162258	-0.317084	-0.011567	0.100413	-0.050224	-0.136009	-0.177037	-0.130498	-0.054766	-0.018691	0.023954	AdenomeraAndre
4	-0.100749	-0.298524	0.037439	0.219153	0.062837	-0.048885	-0.053074	-0.088550	-0.031346	0.108610	0.079244	AdenomeraAndre
5	-0.134528	-0.295123	0.012486	0.180641	0.055242	-0.080487	-0.130089	-0.171478	-0.071569	0.077643	0.064903	AdenomeraAndre
6	-0.100379	-0.236428	0.027070	0.216923	0.064853	-0.046620	-0.055146	-0.085972	-0.009127	0.065630	0.044040	AdenomeraAndre
7	-0.063748	-0.250981	-0.009015	0.184266	0.075654	-0.055978	-0.048219	-0.056637	-0.022419	0.070085	0.021419	AdenomeraAndre
8	-0.122893	-0.282427	-0.044984	0.064425	-0.032167	-0.120723	-0.112607	-0.156933	-0.118527	-0.002471	0.002304	AdenomeraAndre
9	-0.091062	-0.328433	0.042678	0.236484	0.053436	-0.051073	-0.052568	-0.111338	-0.040014	0.090204	0.088025	AdenomeraAndre

Table 2: Dataset sample 2

No	Targets
1	AdenomeraAndre
2	AdenomeraHylaedactylus
3	Ameeregatrivittata
4	HylaMinuta
5	HypsiboasCinerascens
6	HypsiboasCordobae
7	LeptodactylusFuscus
8	OsteocephalusOophagus
9	Rhinellagranulosa
10	ScinaxRuber

Data Analysis

Feature selection is a crucial step in building effective machine learning models, especially when you're dealing with a lot of data. Imagine you're training a model to classify different types of music, like rock, pop, and classical. You could feed it all sorts of information, like tempo, volume, lyrics, and even information about the artist. But that might be overwhelming for the model!

Feature selection helps me choose the most informative data for my model to focus on. It's like picking the key ingredients in a recipe instead of throwing everything in the kitchen sink at it.

In this project, Iam dealing with a dataset of frog calls described by MFCC features. These features are basically measurements that capture different aspects of the calls. I want to identify which MFCC features are most helpful in telling different frog calls apart.

Here's where ANOVA comes in. ANOVA is a statistical test that helps me see which features have the biggest influence on the target variable, which in this case is the type of frog call. It basically runs a competition to see which MFCC feature is best at separating the calls into different frog groups. The feature that wins is the most important one for building your model!

Once I've ranked the features using ANOVA, I can use a tool called SelectKBest to pick a specific number (k) of the most important features. This is like choosing the top 3 most influential ingredients for any recipe.

Finally, I can visualise the results using a bar graph. The higher the bar for a feature, the more it contributed to distinguishing between the frog calls in the ANOVA test. This helps you understand how important each feature is for building an accurate model.

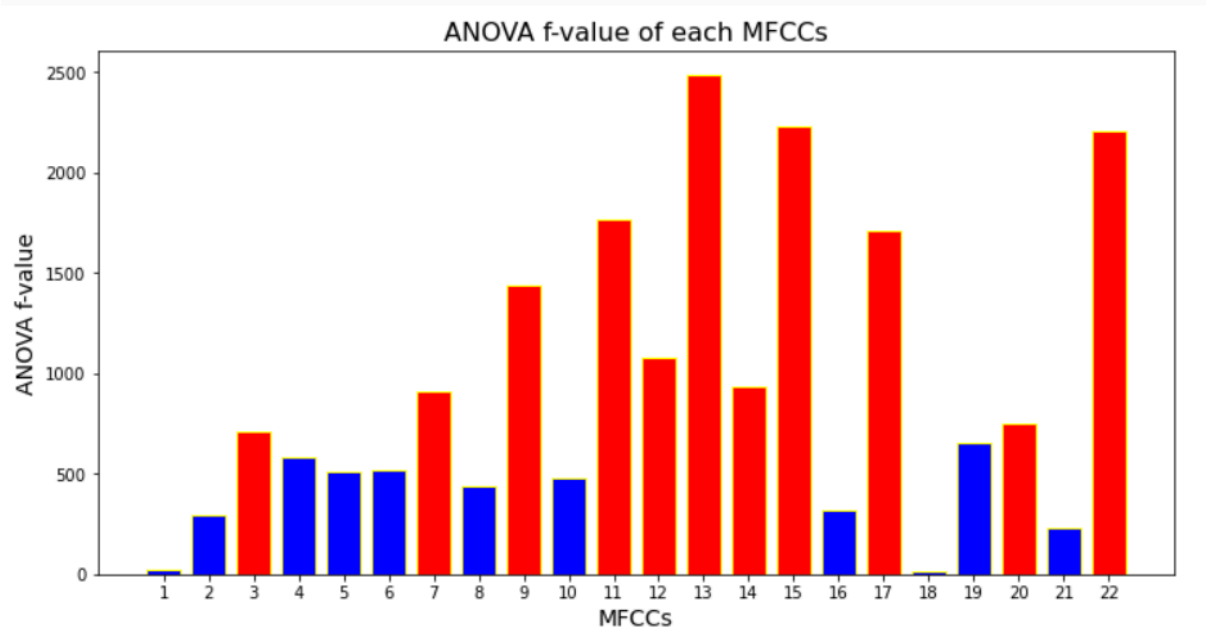

Figure 1: ANVOA f-value against MFCCs

A high f-ANOVA value indicates a high variation between sample means relative to the variation within samples to reject the null hypothesis. Therefore, the top 11 features with the highest ANOVA f-value out of the 22 features will be selected. The reason for selecting 11 features out of 22 features is to reduce the computational cost and modelling time as much as possible to create a highly efficient predictive model implementation. Also, the probability of an overflow problem is reduced to avoid misleading data. Thus, it can improve the performance of the predictive model by training the model faster using only 11 significant features.

Feature	Type	Value/Statistics
MFCCs_3	Continuous numerical	Range: -0.436028 – 1.0 Mean: 0.311224 Std: 0.263527
MFCCs_7	Continuous numerical	Range: -0.538982 – 1.0 Mean: -0.001397 Std: 0.171404
MFCCs_9	Continuous numerical	Range: -0.587313 – 0.738033 Mean: 0.128213 Std: 0.179008
MFCCs_11	Continuous numerical	Range: -0.901989 – 0.523033 Mean: -0.115682 Std: 0.186792
MFCCs_12	Continuous numerical	Range: -0.799441 – 0.690889 Mean: 0.043371 Std: 0.155983
MFCCs_13	Continuous numerical	Range: -0.644116 – 0.94571 Mean: 0.150945 Std: 0.206880
MFCCs_14	Continuous numerical	Range: -0.59038 – 0.575749 Mean: -0.039244 Std: 0.152515
MFCCs_15	Continuous numerical	Range: -0.717156 – 0.668924 Mean: -0.101748 Std: 0.187618
MFCCs_17	Continuous numerical	Range: -0.42148 – 0.681157 Mean: 0.088680 Std: 0.138055
MFCCs_20	Continuous numerical	Range: -0.361649 – 0.467831 Mean: -0.053244 Std: 0.094181
MFCCs_22	Continuous numerical	Range: -0.379304 – 0.432207 Mean: 0.087567 Std: 0.123442

Table 4: The selected features

Data modelling

The researchers built two different machines to predict what kind of frog made a sound based on the sound's measurements (MFCCs). Here's how they did it:

1. **Splitting the Data:** Imagine you have a bunch of frog call recordings. The researchers divided them into two groups: a training set (80%) and a test set (20%). The training set is like a study guide for the machines, and the test set is like a pop quiz to see how well they learned.

Algorithm	Value/Statistics
Decision Tree	Criteria: Gini Max Depth: 14 Min Samples in Leaf: 5 Min Samples to Split: 12
Support Vector Machine	C: 10 Kernel: Radial Basis Function Gamma: Scale

Table 5: Parameters of the predictive models

2. Training the Machines: There are two main machines (models) used here: Decision Tree and Support Vector Machine (SVM).

- **Decision Tree:** This machine works by asking a series of yes/no questions about the sound's measurements to figure out the frog species. The researchers tried different tree depths (how many questions) to see which worked best.

They picked the depth that gave the most accurate answers on the training set (study guide).

- **Support Vector Machine:** This machine is like a fancy line drawer. It tries to draw a line that best separates the different frog calls in the training set. The researchers tested two different line shapes (kernels) and how strictly the line should be drawn (regularisation). They used a special tool (GridSearchCV) to find the best combination of these settings for the test set.
- **K-Nearest Neighbours:** Imagine you have a group of friends, each with distinct characteristics. K Nearest Neighbors (KNN) is like finding the closest friends to you based on those characteristics. Instead of drawing a line, KNN looks at the 'neighbourhood' of data points around each sample and classifies it based on the majority class of its nearest neighbours. The 'k' in KNN represents how many neighbours to consider. If k=3, for example, KNN will look at the three closest neighbours to decide the class of the sample. Just like SVM, KNN can use different distance metrics (like Euclidean distance) and can be tuned to perform better using techniques like cross-validation.

3. Testing the Machines: Once trained, they tested both machines on the unseen test set (pop quiz). They looked at how well they did using different scores:

- **Accuracy:** How many frog calls did they classify correctly overall?
- **Recall:** For each frog type, how many calls did they correctly identify as that type?
- **Precision:** Out of the calls they said were a certain frog type, how many were actually that type?
- **F1-Score:** A mix of recall and precision to give a single score.
- **Confusion Matrix:** This shows how often the machines mixed up different frog calls.

By comparing these scores, I could see which machine, whether it was Decision Tree, SVM or K-Nearest Neighbors, performed better at predicting the frog species from the sound recordings.

.

Results of classification:

```
Accuracy: 0.940236275191105

Confusion matrix:
 [[128   0   4   3   1   1   0   1   0   1]
 [  0 691   0   4   0   1   0   0   0   0]
 [  4   2  86   2   0   1   0   0   0   0]
 [  2   5   0  56   0   1   2   0   0   2]
 [  1   0   0   1  91   3   0   2   0   0]
 [  1   2   1   1   3 211   2   1   0   4]
 [  2   0   0   0   2   2  46   1   0   0]
 [  4   1   0   0   3   2   0   9   0   1]
 [  0   1   0   0   0   0   0   0  11   1]
 [  1   1   0   0   0   5   0   0   0  24]]

Classification report:
                       precision    recall  f1-score   support

        AdenomeraAndre       0.90      0.92      0.91       139
AdenomeraHylaedactylus       0.98      0.99      0.99       696
    Ameeregatrivittata       0.95      0.91      0.92        95
           HylaMinuta       0.84      0.82      0.83        68
  HypsiboasCinerascens       0.91      0.93      0.92        98
     HypsiboasCordobae       0.93      0.93      0.93       226
   LeptodactylusFuscus       0.92      0.87      0.89        53
 OsteocephalusOophagus       0.64      0.45      0.53        20
     Rhinellagranulosa       1.00      0.85      0.92        13
           ScinaxRuber       0.73      0.77      0.75        31

              accuracy                           0.94      1439
             macro avg       0.88      0.84      0.86      1439
          weighted avg       0.94      0.94      0.94      1439
```

Results of classification of anuran species using Decision Tree

```
Accuracy: 0.9742876997915219

Confusion matrix:
 [[136   0   0   2   0   0   0   0   0   1]
 [  0 693   1   2   0   0   0   0   0   0]
 [  0   0  95   0   0   0   0   0   0   0]
 [  3   3   1  59   0   0   1   0   1   0]
 [  1   0   0   0  97   0   0   0   0   0]
 [  0   2   0   1   0 214   4   4   0   1]
 [  0   0   0   0   0   2  51   0   0   0]
 [  1   0   0   0   2   0   1  16   0   0]
 [  0   0   0   0   0   0   0   0  13   0]
 [  1   0   0   0   1   0   0   1   0  28]]

Classification report:
                       precision    recall  f1-score   support

        AdenomeraAndre       0.96      0.98      0.97       139
AdenomeraHylaedactylus       0.99      1.00      0.99       696
    Ameeregatrivittata       0.98      1.00      0.99        95
           HylaMinuta        0.92      0.87      0.89        68
  HypsiboasCinerascens       0.97      0.99      0.98        98
    HypsiboasCordobae        0.99      0.95      0.97       226
   LeptodactylusFuscus       0.89      0.96      0.93        53
 OsteocephalusOophagus       0.76      0.80      0.78        20
    Rhinellagranulosa        0.93      1.00      0.96        13
          ScinaxRuber        0.93      0.90      0.92        31

             accuracy                           0.97      1439
            macro avg       0.93      0.94      0.94      1439
         weighted avg       0.97      0.97      0.97      1439
```

Results of classification of anuran species using Support Vector Machine

```
**********************************************************************************************************

Accuracy: 0.9861014593467686

Confusion matrix:
[[137   0   0   0   0   1   1   0   0   0]
 [  0 696   0   0   0   0   0   0   0   0]
 [  0   0  95   0   0   0   0   0   0   0]
 [  1   0   0  67   0   0   0   0   0   0]
 [  0   0   0   0  96   1   0   1   0   0]
 [  0   2   0   1   2 217   2   2   0   0]
 [  0   0   0   0   1   0  52   0   0   0]
 [  0   0   0   0   3   1   1  15   0   0]
 [  0   0   0   0   0   0   0   0  13   0]
 [  0   0   0   0   0   0   0   0   0  31]]

Classification report:
                       precision    recall  f1-score   support

        AdenomeraAndre       0.99      0.99      0.99       139
AdenomeraHylaedactylus       1.00      1.00      1.00       696
    Ameeregatrivittata       1.00      1.00      1.00        95
           HylaMinuta       0.99      0.99      0.99        68
  HypsiboasCinerascens       0.94      0.98      0.96        98
     HypsiboasCordobae       0.99      0.96      0.97       226
   LeptodactylusFuscus       0.93      0.98      0.95        53
OsteocephalusOophagus       0.83      0.75      0.79        20
     Rhinellagranulosa       1.00      1.00      1.00        13
           ScinaxRuber       1.00      1.00      1.00        31

              accuracy                           0.99      1439
             macro avg       0.97      0.96      0.96      1439
          weighted avg       0.99      0.99      0.99      1439


**********************************************************************************************************
```

Results of classification of anuran species using K-Nearest Neighbors

After comparing the three models, namely Decision Tree, SVM and K-Nearest Neighbors, for identifying frog species from sound recordings. I looked at how many predictions each machine got right overall (accuracy) and how good they were at specific frog types (precision and recall). Based on those scores, it seems like KNN is the better choice.

Here's why KNN did a better job:

- **More Bullseyes:** Imagine a table where correct guesses land on the diagonal. KNN had more "bullseyes" (correct guesses) on that table than the SVM of the Decision Tree (higher accuracy, 99% vs 97.43% and 94.02%).
- **More Specific:** When KNN said a sound belonged to a certain frog, it was usually right (higher precision, like pointing at a sound and saying the exact frog type). The SVM and Decision Tree were less specific (lower precision).
- **Finding More Frogs:** Imagine there's a pond with a specific type of frog. KNN found more of those specific frogs in the recordings (higher recall). The Decision Tree and SVM missed a few.
- **Overall Champion:** Combining these (precision and recall), KNN did better overall (higher F1-score).

So, if you want to build a machine to identify frog calls, KNNseems like the winner based on this test! This means that KNNmight be better for future datasets of frog sounds as well