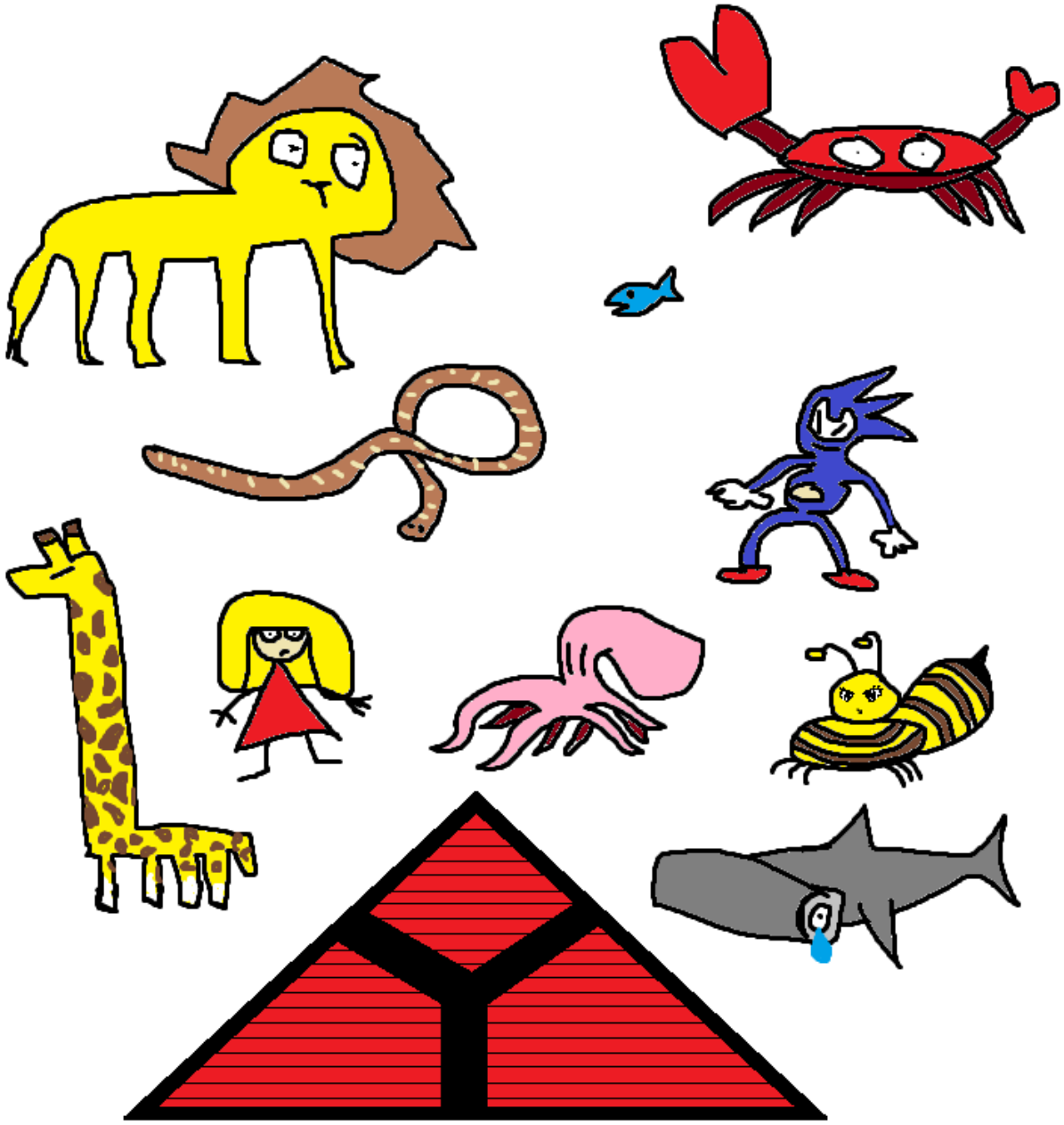

DTE-2602 assignment 1: Classify Animals by Traits

author: Illya Reseland

date: 2024.8.26

1 INTRODUCTION

The assignment was to classify animals based on a list of 16 traits. The traits varied from number of legs, hair, milk and more. The goal is to create an Ai capable of classifying what category it falls based on a provided list of traits. No restrictions for what tools or methods could and could not be used and as a result the number of possible solutions quite vast. The conditions the solution must satisfy are that when put through a provided unit test, it scores more than 5 out of 24. The assignment itself is included in the GitHub repository along with all other files.

2 THEORY

The problem of classification is a common one within data science and as such both a plethora of tools and resources exist regarding the subject. The tool chosen for the assignment was scikit. A popular, easy to use and powerful library for python. The classifier chosen was "RandomForestClassifier" as it has been shown to work well when it comes to problems of classification. On top of that it had the best performance of all algorithm that were tested when picking one to use for the assignment. The scikit cheat sheet was used to determine what algorithms to try.

3 METHOD

In the assignment multiple files were provided as a starting point. It was quickly decided that starting from scratch would be faster and lead to a better product. The exceptions being the data set and the unit test. both being lightly modified to work better with the other code. I would highly recommend looking through the GitHub repository and reading the code for yourself. The application is spread over multiple .py files to make it as maintainable and robust as possible. Starting with "data.py" from where the data set is exported as a dictionary. The reason for saving and loading the file is that it makes the application more modular and easier to maintain as the training data can more easily be manipulated, updated, or swapped out. Next the data set gets loaded in by "createSkynet.py". "createSkynet.py" is used to both set up the Ai model and fit it to the training data. After the model has been created it is exported as a file that can be loaded up and used at any time. "classifier.py" can then be used to load the fitted model, give it an input and have it return an output. Lastly there is the unit test. It tests the application. Not much needs to be said about it.

4 RESULT

When running the unit test, 22 of 24 come out as correct. The result is a simple machine learning application that works well. It should be easy to modify if necessary, because of the modular nature of both the application itself and the and scikit, which is the library the application is built on top of. The application checks every box required by the assignment and a few more.

5 DISCUSSION

Personally, I am quite happy with the result. There are things that could have been done better like using a grid search to further fine tune the hyper-parameters. A larger data set could also help to increase the accuracy. But even without these improvements the accuracy and quality it has pleases me.

6 CONCLUSION

The assignment was solved in a way that checks all the boxes. The product being robust, but also easily malleable and modifiable. There is clear room for improvement. But even as it is currently it still performs remarkably well. At its core it uses scikit which is most of the reason it works as well as it does. For more information on the assignment or for the files for the project itself, click here -> GitHub repository for assignment.