ECE 326 - Programming Languages - Fall 2022

Mini-Project

SUBMISSION: December 07, 2022, 11:59pm

Mathura Shivakaran
1006879184

Question 1: Provide a brief description of Tensorflow. Compare pros and cons with scikit package.

Tensorflow is an open-source library which is mainly for numerical computations. Tensorflow is a great tool for Machine Learning and Artificial Intelligence. Tensorflow particularly focuses on deep learning and developing deep neural networks.

Pros:

- ➔ Tensorflow is a much better option for deep learning and neural networks as Scikit does not focus on that.
- ➔ Tensorflow's specialization in deep neural networks makes it easy to compare between different complex neural networks.
- ➔ Scikit implements some basic neural networks, but does not support more complex neural networks similar to the way Tensorflow does.
- ➔ In Tensorflow, the developer must implement all the fine details of their models, while Scikit helps with that with existing machine learning algorithms. Having the developer implement everything causes an increase in performance and accuracy.

Cons:

- ➔ Scikit provides many frameworks to work with while Tensorflow is limited to neural networks.
- ➔ Scikits multiple provided frameworks make it easy to compare between distinct machine learning models.
- ➔ In Scikit, you can compare a variety of machine learning models and algorithms, which is limited in Tensorflow.
- ➔ In Scikit, the developer does not need to implement all the final details of their models, which can make development faster and easier.

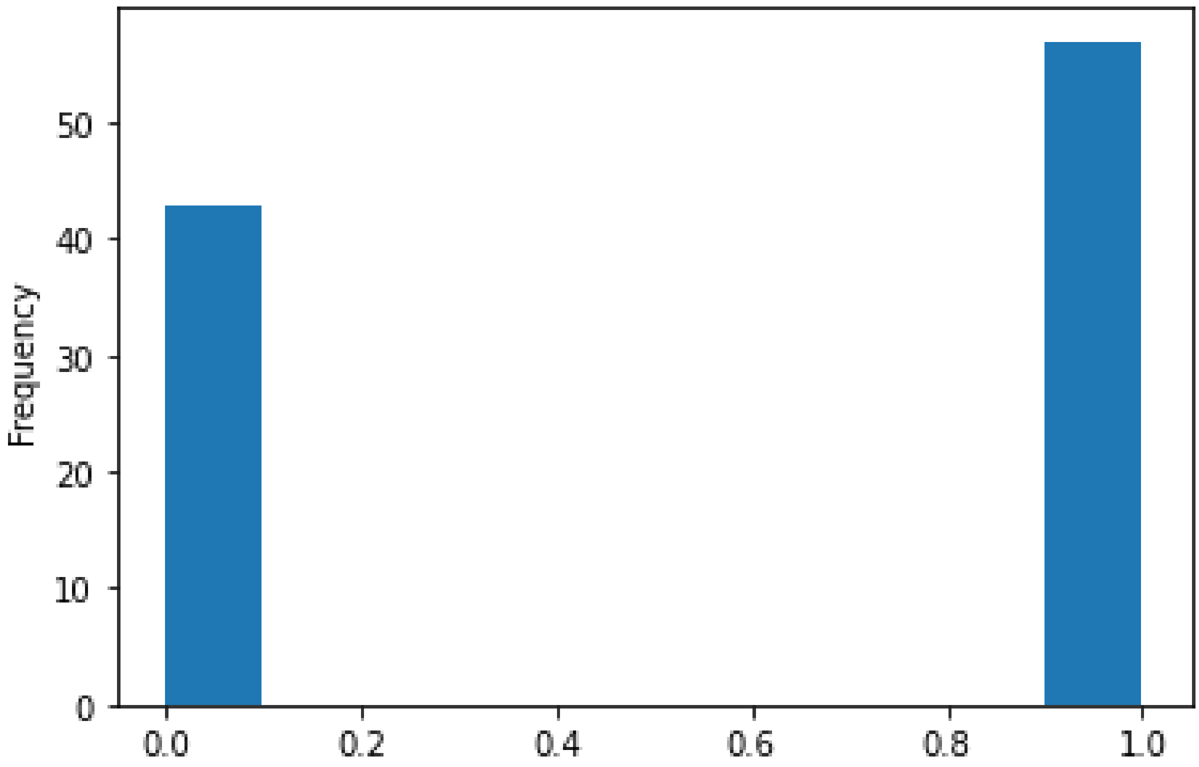
Question 2: Choose a dataset, provide a reason why you chose that dataset.

The dataset that I have chosen from the available datasets is the cats_vs_dogs dataset. I think this is a good dataset to start working with for an introduction to machine learning, hence I selected it. It is also something that the machine can easily improve upon when it comes to improving the next prediction and identification. This dataset contains simple building blocks and simple pieces of data, thus making it easier for the developer (me) who is new to machine learning to gain a stronger understanding of what occurs in Tensorflow and how it manipulates this dataset to improve its prediction on whether a piece of data in the dataset is either a cat or a dog. Along with this, only having two divisions in the dataset again creates an easier foundation to understand the building blocks behind machine learning in Tensorflow.

Question 3: What anomalies (if any) did you filter out, and why?

In this dataset, I was unable to filter any anomalies, so it is safe to make the assumption that no anomalies were found in the dataset. To test for anomalies, when printing out the dataset, I put in a condition. I stated that if the label is neither a 0 or a 1, it should print out “Anomaly found.”. However, upon running the print statement on the iterated dataset, it never once printed the output, indicating that there were no anomalies in this data set. It is also mentioned in the documentation for this dataset that all corrupted images were already effectively dropped from the dataset. In the case anomalies existed: a way to find out if an anomaly exists is by checking the labels of all the images. In this dataset, a label of 0 indicates a cat and a label of 1 indicates a dog. If the label is of any other integer value other than 0 and 1, that image can be considered an anomaly as it neither signifies a cat or a dog. Then, when writing in the dataset as a dataframe and then as a CSV file, I would ensure that the anomalies would not be written in.

Question 4: Draw a histogram for the important parameters.



The x-axis has two values, (0) and (1). The (0) represents the cats in the dataset, while the (1) represents the dogs in the dataset. The y-axis represents the frequency of each value, so it shows how many cats and dogs are found in the dataset. The dataset provided by Tensorflow was cut down to the first 100 pieces of data, for simplification and program running purposes.

Question 5: Do you think Tensorflow can be used to implement a spell-checker? Explain.

I believe that Tensorflow can most definitely be used to create a smart spell-checker. Often, spelling mistakes are made due to common reasons. One reason would just be the overall pronunciation of the word. Often, when a spelling error is caused due to one spelling the word based off of pronunciation, many people will make the same or similar spelling errors, as the word is always pronounced similarly. I believe that with Tensorflow's deep neural networks, the machine can learn what the correct spelling is after seeing many people put in similar spelling errors for the same word. Rather than having to cross reference the whole dictionary each time, through this machine learning in Tensorflow, the program will provide the correct spelling faster. Another reason spelling errors occur is due to grammatical rules which some people do not remember. Again, if enough people make the same spelling error for the same word, using Tensorflow and machine learning, the program will quickly learn to predict from the spelling error itself what the correct word is, rather than having to cross reference the whole dictionary and do ascii value checks, etc. With Tensorflow Machine Learning using deep neural networks and having the ability to make predictions from previous actions, I believe tensorflow can most definitely be used to create an efficient and fast spell-checker.