Citrus Fruit Classification

Sai Sree Doddapuneni - CS5300

May 5, 2023

Contents

1 Introduction

Although it seems very obvious to a human to distinguish between oranges and grapefruit, there are still some mistakes in manual observation. This dataset creates a larger dataset with a wide range of values that are "oranges" and "grapefruit" and uses the color, weight, and diameter of an "average" orange and grapefruit.

2 Dataset

The **"Citrus Dataset"** Downloaded the citrus dataset from kaggale website which is a good repositiory of datasets. The dataset contains columns diameter,weight,red,green and blue:

- **Diameter**: This aspect represents the diameter of the fruit.
- **Weight**: This feature represents the weight of the fruit.
- **red**: This indicates the colour of the fruit.
- **Green**:This indicates the colour of the fruit.
- **Blue**: This indicates the colour of the fruit.

The dataset contains 10001 data samples. These samples have been manually verified by annotators, and each row contains 12 continuous variables as well as a class label at the end, with two possible values - 0 (negative) and 1 (positive). The input features pertain to certain fields.

- Diameter, Width, Red, Green, Blue.

2.1 Visualization of the distribution of each input features

The histogram plot before normalization of each info highlights indicating their most extreme and least value as well as how they are distributed can be found in the pictures given underneath.

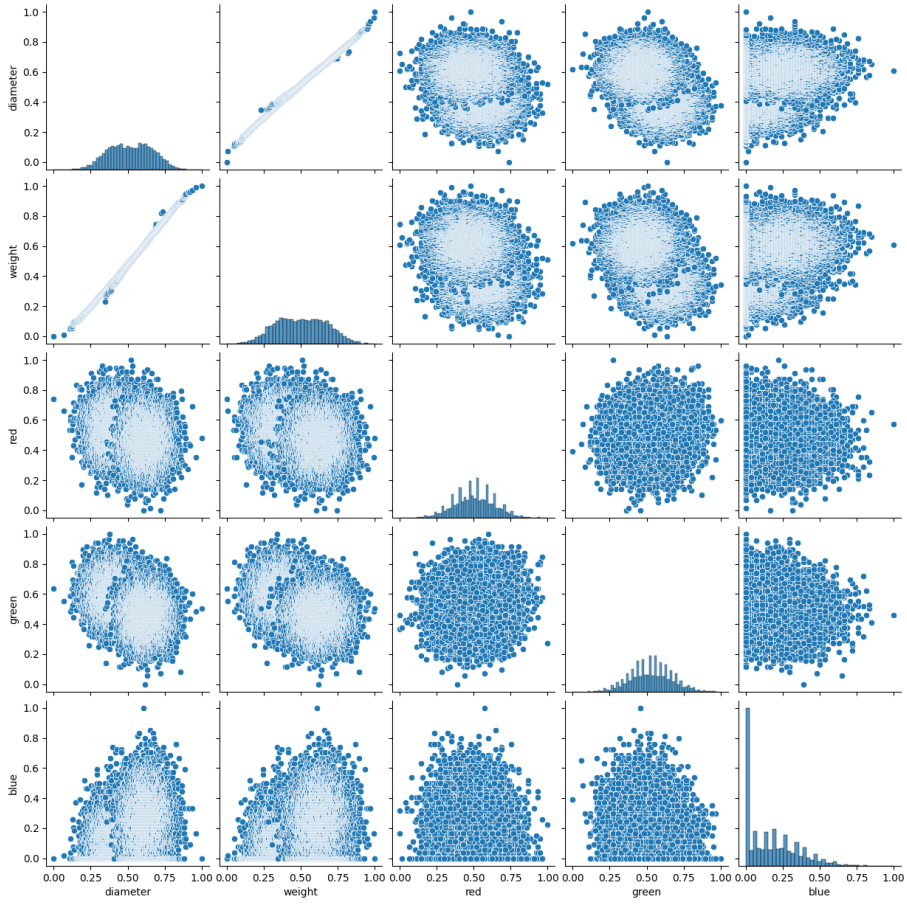

Figure 1: Input Data Distribution Histograms - Before Normalization

2.2 Distribution of the output labels

Notice that the data is imbalanced and may need to be resampled (such as oversampling, undersampling, or generate syn- thetic samples).

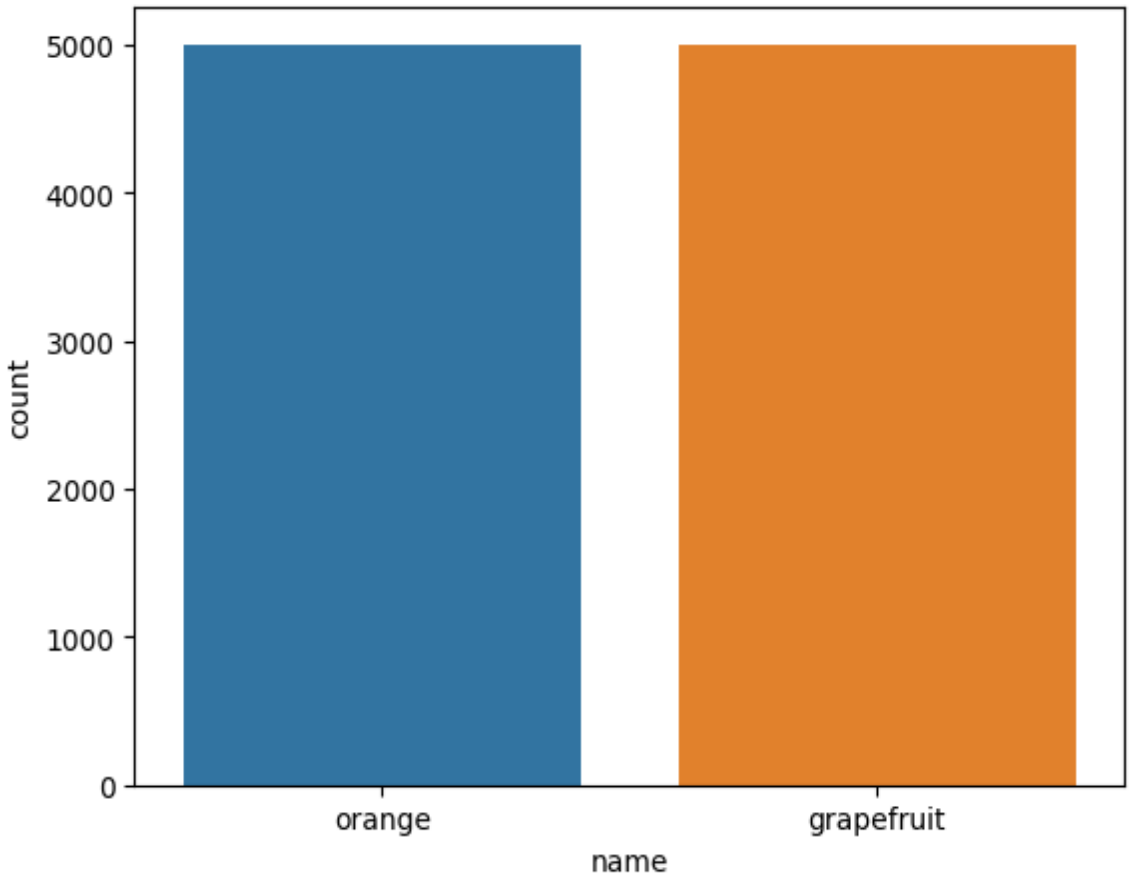

Figure 2: Output Data Distribution - Before Normalization

3 Data Processing

3.1 Data Splitting

The data was taken randomly and the dataset was split into training and validation, where 70% of the dataset was allocated for training and 30% was allocated for validation or testing.

3.2 Normalization of data

Prior to data mining, data pretreatment is crucial to resolving the uneven distribution of the data. In order to accomplish this, normalization techniques are utilized to strengthen training and increase the numerical stability of the optimization problem. All values should fall between 0 and 1, and outliers should be discernible in the normalized data, thanks to normalization. Both of the available normalizing methods—each with distinct side effects—can be employed at the moment.

Mean Normalization Formula

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Z-Score Normalization

$$X_{normalized} = \frac{X - X_{mean}}{X_{standard_deviation}}$$

4 Modelling

A feed forward artificial neural network architectures was used to create the model.

NOTE: Data is shuffle. Thus, the result will vary every time. All models were compiled and fit on May 1, 2022.

4.1 Accuracy Test vs Train

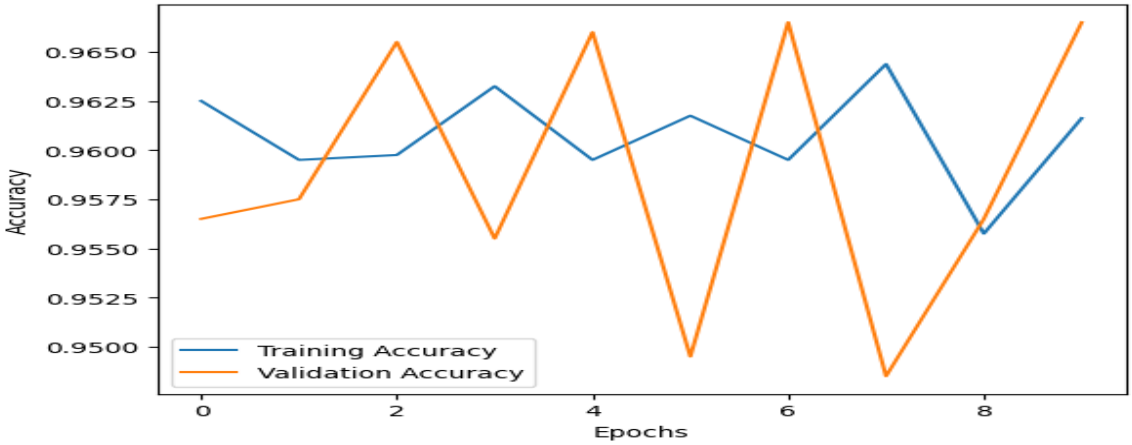

Figure 3: curves showing accuracy of test and train data

As seen from the above figure,the validation accuracy falls suddenly and training accuracy is a bit linear.

5 Model Evaluation

Three essential classification model metrics to evaluate. The table given below shows the precision, recall and f1 score for the neural network model.

1. Precision: what proportion of positive identifications was actually correct?
2. Recall: what proportion of actual positives was identified correctly?
3. F1-Score: evaluation metric for classification algorithms, where the best value is at 1 and the worst is at 0.

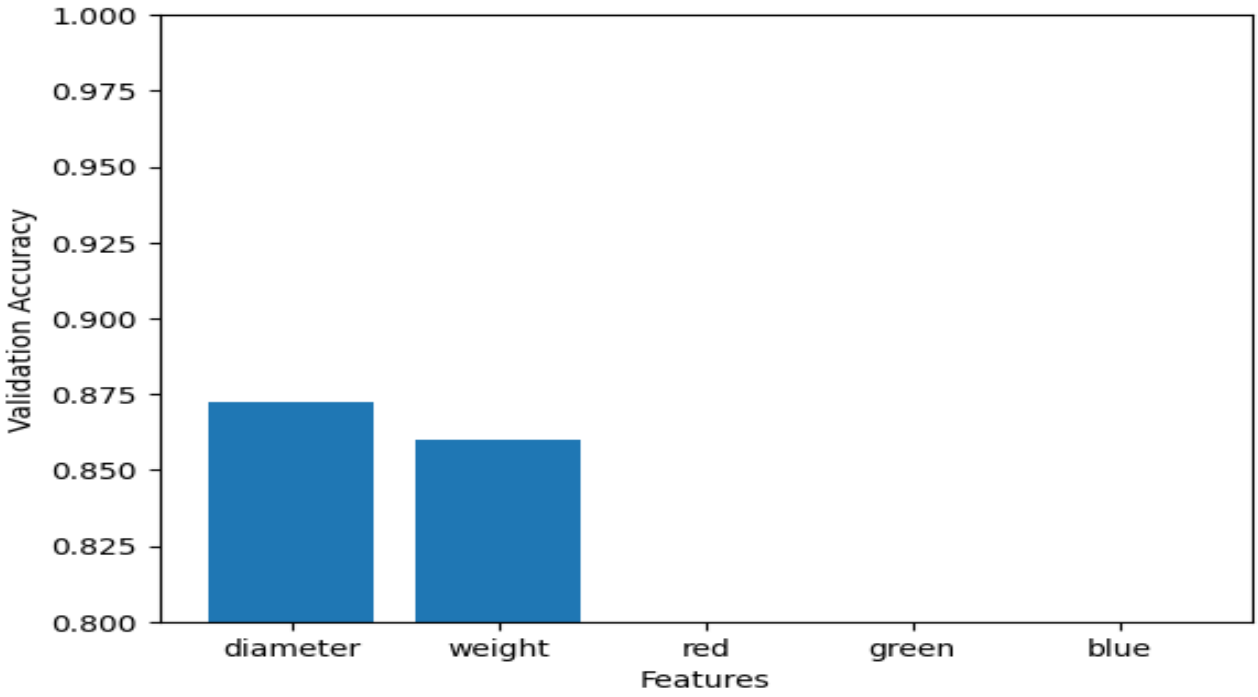

Figure 4: Validation accurancy

A useful tool when predicting the probability of a binary outcome is the Receiver Operating Characteristic curve or ROC curve. The area covered by the curve is the area between the red line and the axis. This area covered is AUC. The bigger the area covered, the better the machine learning models are at distinguishing the given classes. In other words, the AUC can be used as a summary of the model skill. The ideal value for AUC is 1.

5.1 Test Accuracy after removing features

The closer the graph is to the top and left-hand borders, the more accurate the test. Likewise, the closer the graph to the diagonal, the less accurate the test. In a perfect test, it would go straight from zero up the top-left corner and then straight across the horizontal. The figure is given below shows the receiver operating characteristic curve for the different neural network architectures.

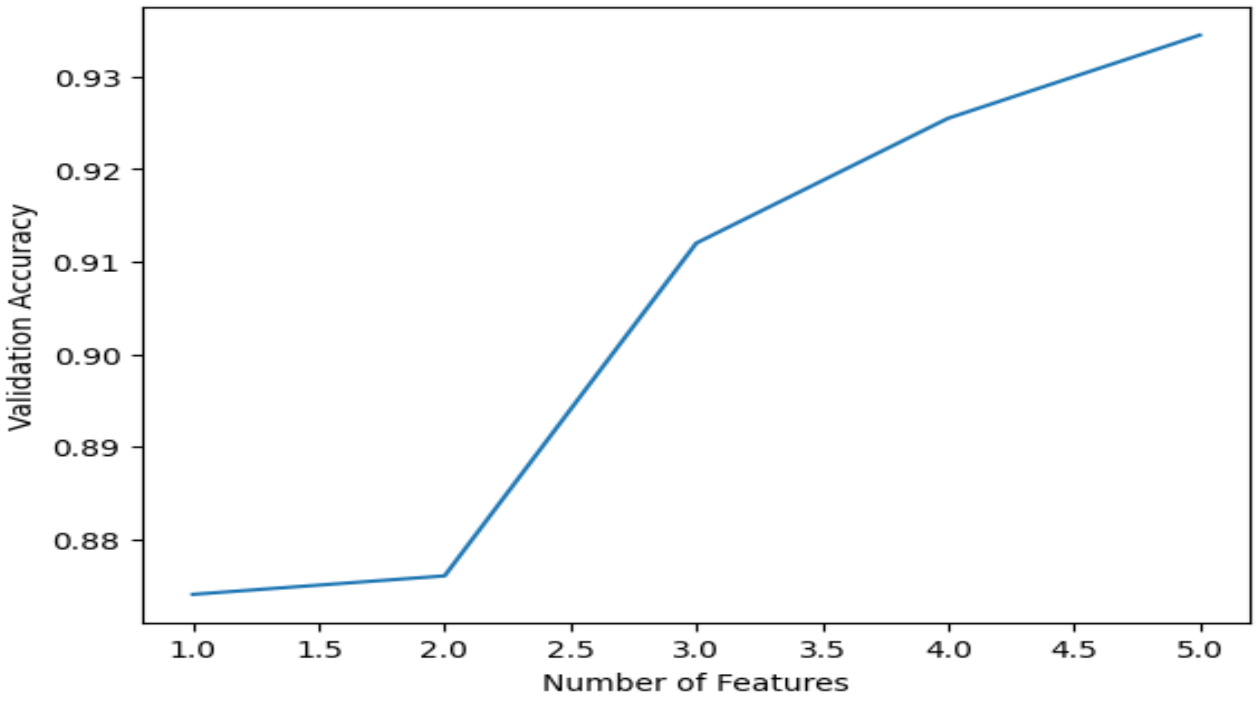

Figure 5: Validation accurancy after feature reduction

6 Challenges Faced

First i started with US Census Data which has categorical and the data conversion is challenging then i decided to get the simple dataset and i choose citrus dataset which is a simple dataset with mininum features.

7 Conclusion

This projects shows a simple classification using binary classification algorithm using tensor flow.