Predicting Iris Species Presentation

A short synopsis of lab 3

Jean-Sebastien Roger[1]

[1]Florida Atlantic University

Biostatistics, Fall 2020

Abstract

In this experiment we were tasked with constructing a procedure which trains a model to predict the species of iris flower based on sepal and petal dimensions. Amongst other factors, we would expect the model we train to pick up on distinguishing factors such as flower size. Below are the 4 dimensions we used in prediction:

- Sepal width
- Sepal length
- Petal width
- Petal length

Our experiment was successful in the sense that, not only were we able to train a 3-layer neural network model which predicts the species of iris, we were also able to test our results against an out of sample test set showing that our model was accurate.

The table below depicts a sample from the iris dataset we are working with, as you can see, there are the 4 main predictors we will use in our model

Examples

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

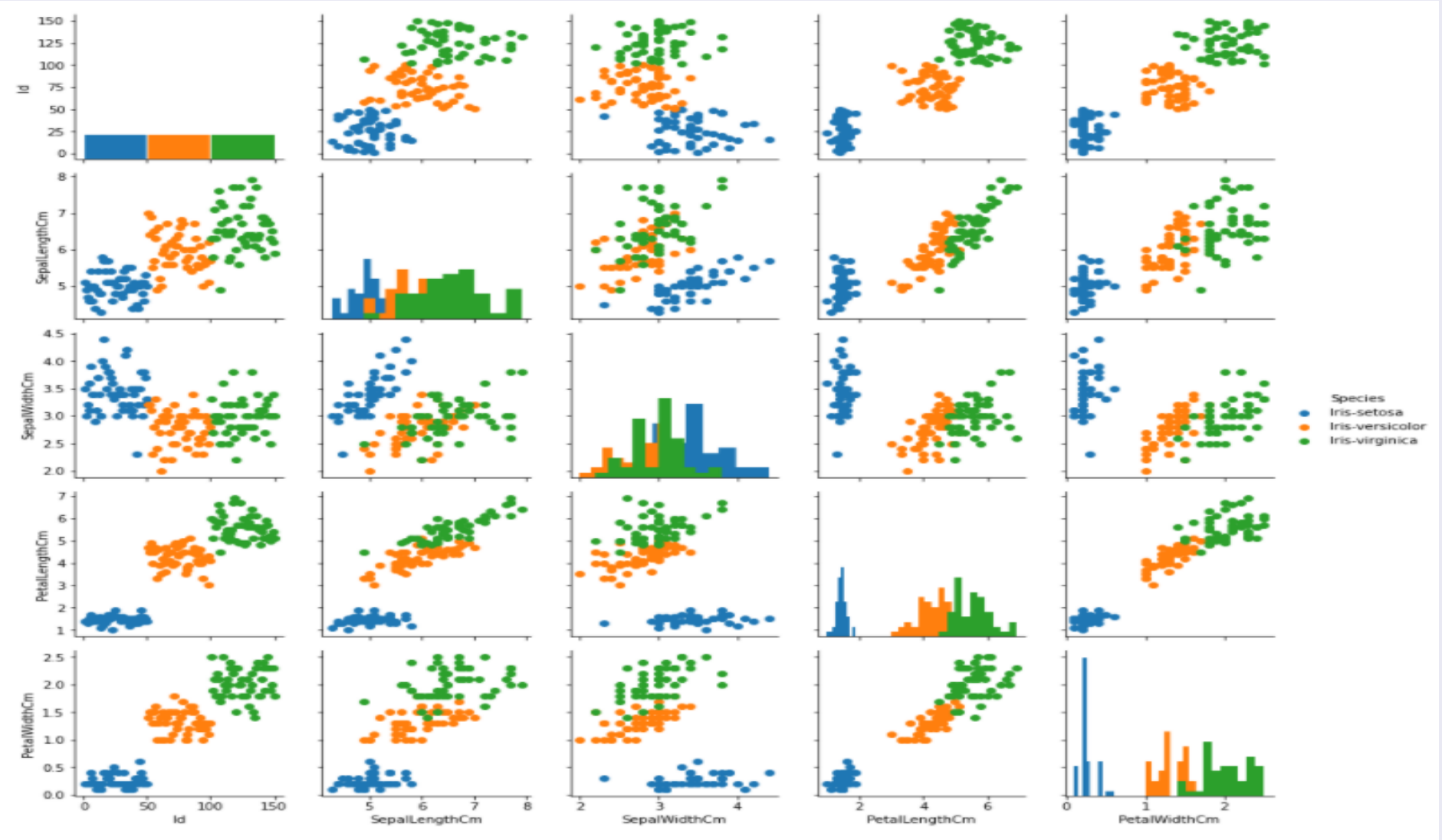
Procedure

Using python code, we:

1. Read in the csv of iris data
2. Plot distributions of the data
3. Split the data set into a training set and validation set
4. Define functions for modeling and optimization
5. Iterate through training the model while minimizing cross entropy
6. Validate model performance on test set

Visualizations

Scatter plot of predictors by species



Model Training

The model training step was the most intensive section of this lab. In the training function we used a backwards gradient descent approach to weight optimization, that way with each iteration, our training function aimed to optimize the model weight which would minimize the cross entropy loss on the training set. Within our training function we had three key tuning tools:

- Cross entropy loss function
- backwards step function
- gradient descent function

In the end, we were left with a model tuning loop which adjust the weights of the layers of our network until the loss is minimized within the number of iterations

Conclusion

From this experiment, we were able to train a model to predict iris species based on flower dimensions and then validate its accuracy on a validation sample. We were able to achieve a validation accuracy of around 96.67%. This seems to be very good accuracy given the limited number of dimensions we have to use as predictors.

The next steps for a lab such as these seems to create a confusion matrix on a bigger dataset, this will help us identify weak points in our model and really speak more towards other evaluation metrics which can support our accuracy claims.