Wall393HW1

Aidan Wall
Chapman University

February 22 2022

1 Introduction

The goal for assignment 1 is to use scikit-learn to do SVM classification on a slight variation of the classic iris dataset. The data set contains 4 input variables, Sepal Length in CM, Sepal Width in CM, Petal Length in CM, and Petal Width in CM. These are used to predict the Species of iris, either "Iris-setosa" or "Not-Iris-setosa". Each data point also has its own ID but that is not used in our model. Here is a snippet of what our data looks like:

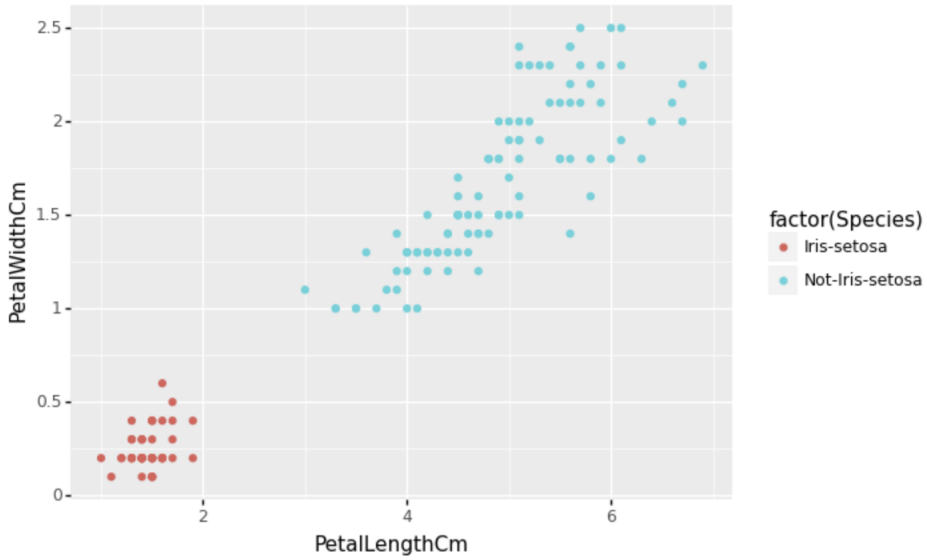
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa

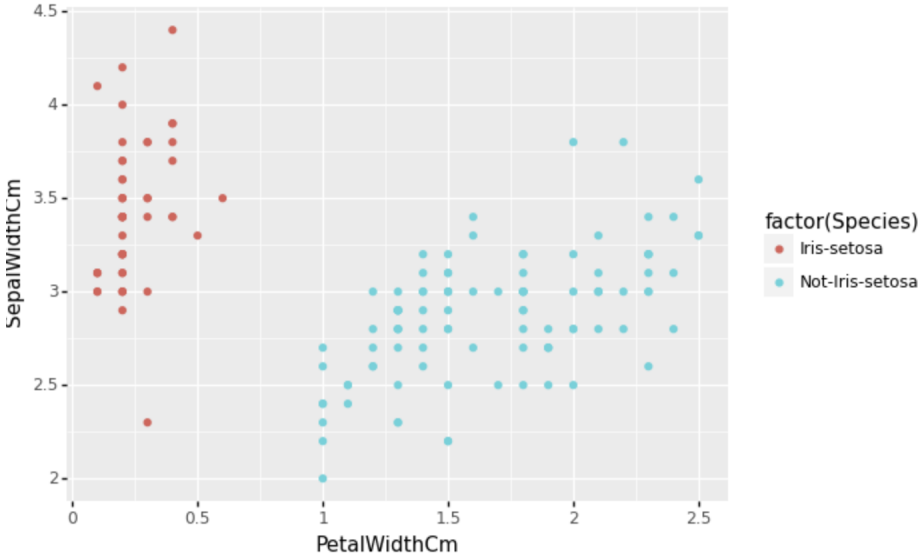
2 Background

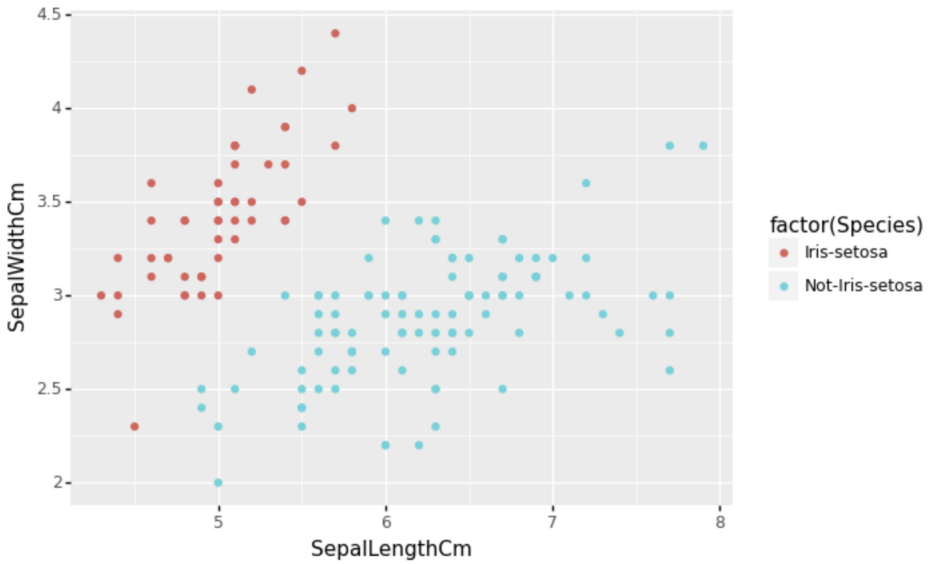
We will look to use SVM(Support Vector Machines) to classify our data into either Iris-setosa or not. SVM's are common and very useful supervised machine learning algorithms as well as a regression algorithm. SVM's are known for the kernel trick to handle nonlinear input spaces and transpose them onto higher dimensions to find and distinguish relationships. The main purpose of this algorithm is to create the best possible decision boundary that will separate two or more variables from each other witht he maximum amount of space in between.
Using the scikit learn package, I created a SVM with 4 features, so I decided to use the Linear Kernel type, because it is versatile when there are a lot of features. I choose this over Gaussian Radial Basis Function (RBF) kernel type because RBF is more for non-linear data which does not really fit our data. I performed a train test split on our data with 80% training data and 20% test data.

3 Findings

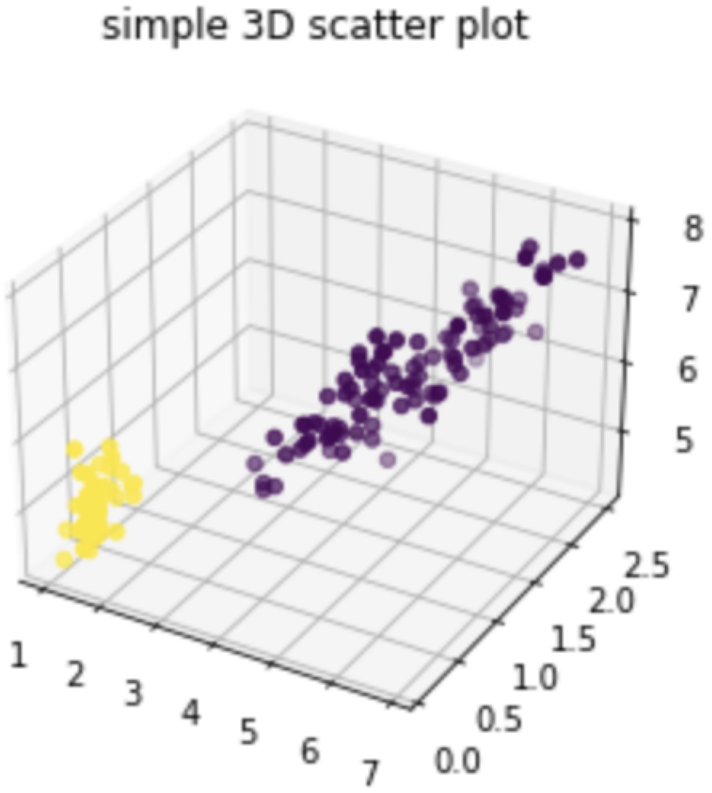
Looking at it Visually
I plotted some of variables against each other to see the SVM's at work in 2 dimensions. This first graph plots Petal Width vs Petal Length. And as shown, there is a clear distinction between Iris-Setosa and Non-Iris-Setosa based on these 2 variables. The same is true when plotting any variable against any other. There is a clear distinction between the two Species and a line can be drawn to perfectly classify the two species.


PetalWidthCm
PetalLengthCm
factor(Species)
Iris-setosa
Not-Iris-setosa


SepalWidthCm
PetalWidthCm
factor(Species)
Iris-setosa
Not-Iris-setosa



I also did it to show the distinction in 3 dimensions using the three variables with the largest impact on species, determined looking at the coefficients of a logistic regression model, Petal Length, Petal Width, Septal Length:



It is possible to draw a plane between the two different classification of groups, that would perfectly classify the species. Since the data is considered linearly separable, this makes it perfect to use SVM and does not necessarily need to use the kernel trick. The kernel trick is used when data is not linearly separable and needs to be transposed into a higher dimension in order to find a separation in the data.

Model Findings

The SVM classification model worked perfectly, and that is not meant lightly. To analyze the performance of the model, I used the accuracy score metric, which describes the percentage of predictions that the model made correctly. My model had a score of 1.0 meaning it made every prediction correctly. I also used the precision score, which returns the percentage of each input that was positive, which was also 1.0.

References

[SVM] SVM for Linearly Separable Data in Python

[3DG] 3D Graphing Using Matplotlib