**Support Vector Machines (SVM)**

**INTRODUCTION**

The dataset given for our first assignment is called the Iris-Setosa classification set. This assignment aims to see if we could use SVMs and the features in our iris data set to predict whether the flower is an iris setosa or not. This report will go in-depth about how we use SVM to classify our dataset of Iris flowers. SVM stands for Support Vector Machines and is a machine learning model used to help classify data. The background section of this report will explain what SVM is and help explain how it works. Then, this report's Analysis / Method section will show the steps I took to implement the SVM algorithm. Finally, the results section will discuss how well the model performed using SVM on this dataset.

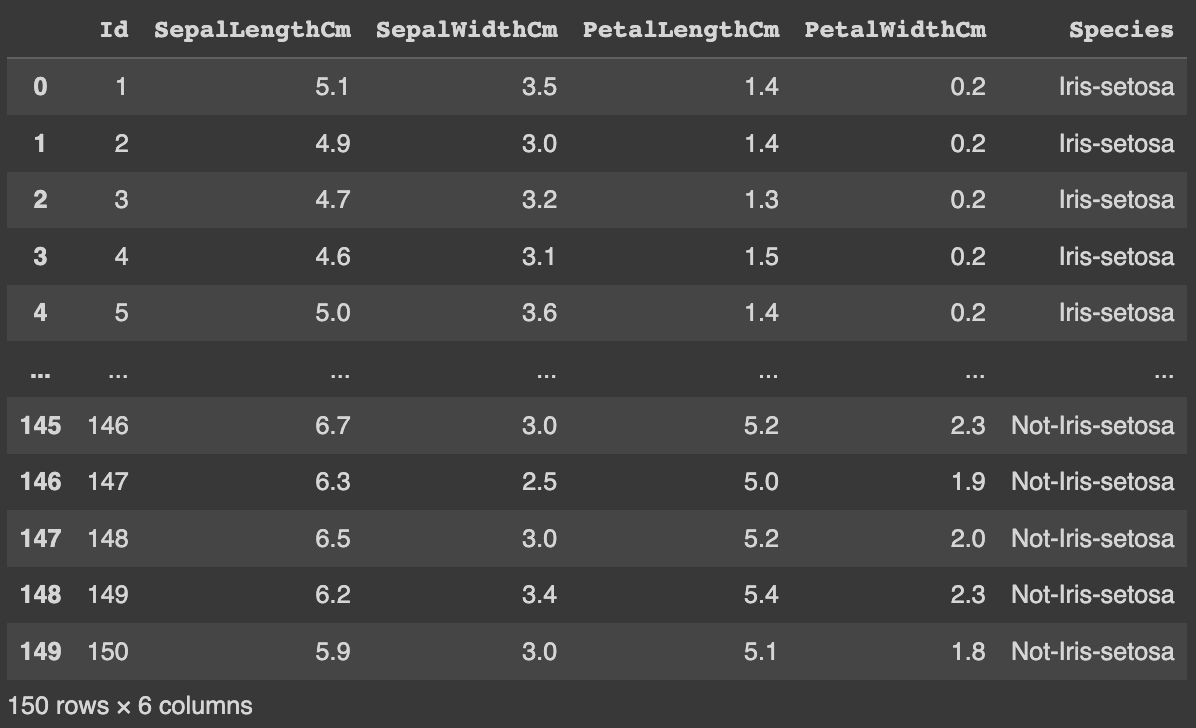
**BACKGROUND**

Support Vector Machine (SVM) is a powerful algorithm used in supervised learning for classification and regression tasks. It aims to separate data points into different classes accurately. This is done by finding the hyperplane that maximizes the margin between the two categories. The margin is the distance between the hyperplane and the closest data points from each class, known as support vectors.

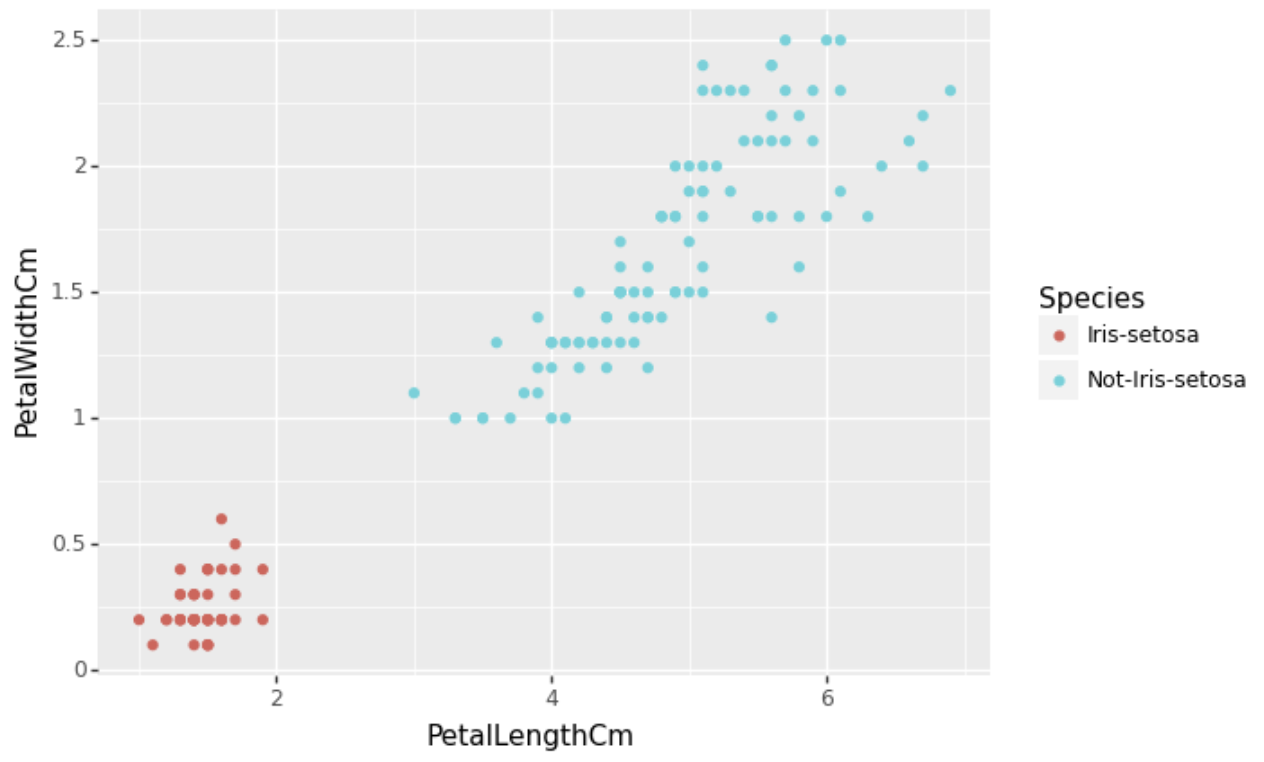
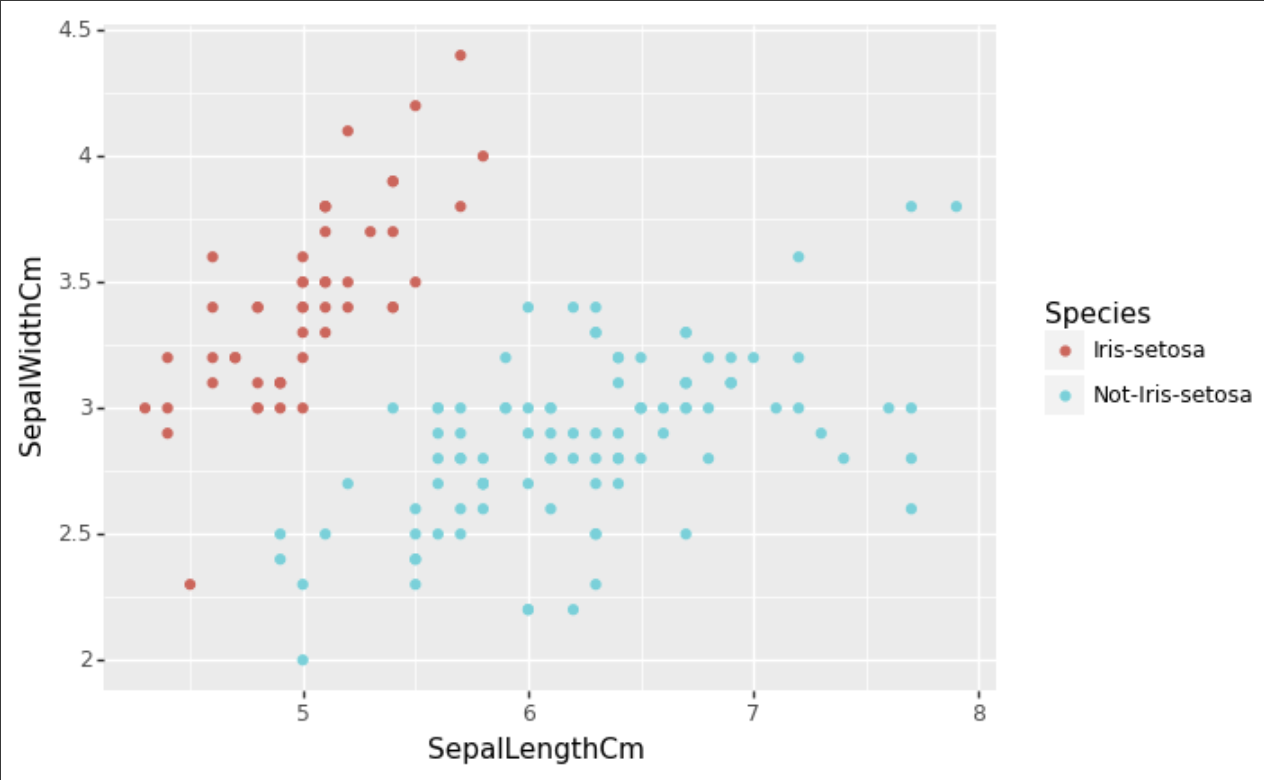
SVM strives to maximize the margin because a larger margin leads to better generalization performance and lowers the risk of overfitting. Generalization performance refers to how well the model performs on new data it has not seen before. By maximizing the margin, SVM can find a decision boundary with a larger separation between the classes, making it more likely to classify new data points accurately. In other words, SVM aims to reduce the risk of misclassifying new data points. Overall, SVM is a robust algorithm that can accurately classify data in two-dimensional and higher-dimensional spaces by maximizing the margin between two classes.

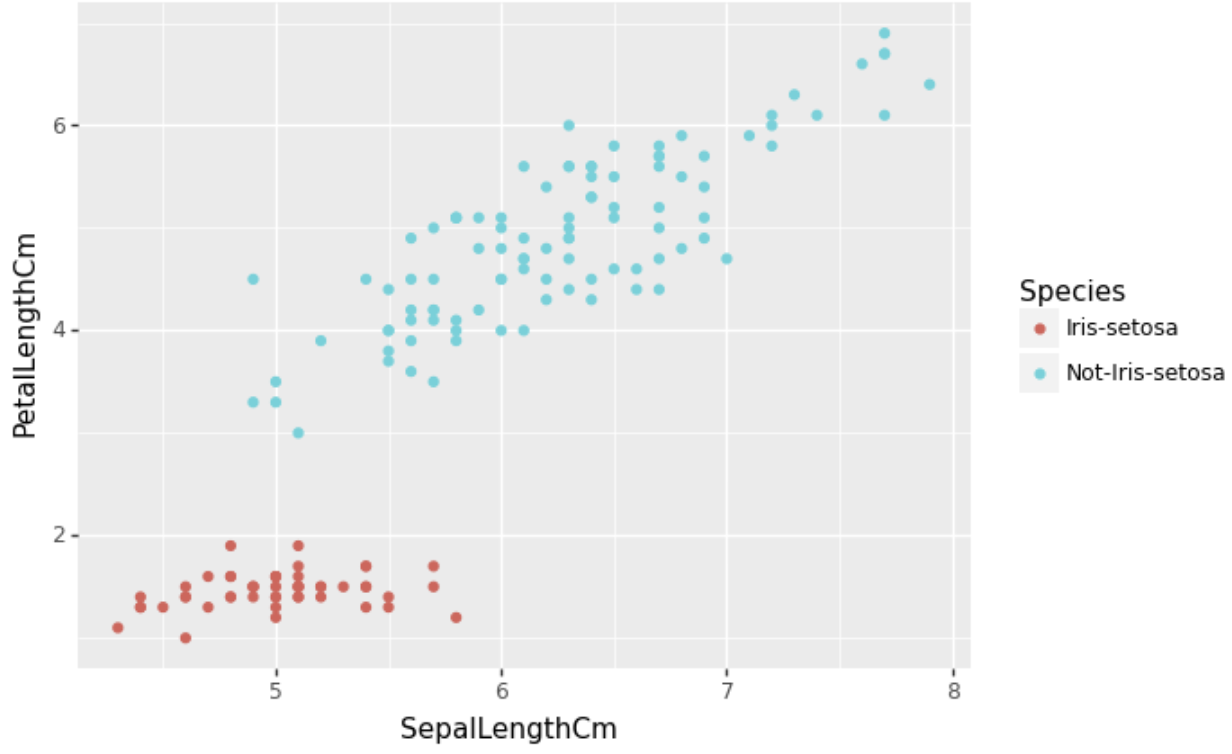
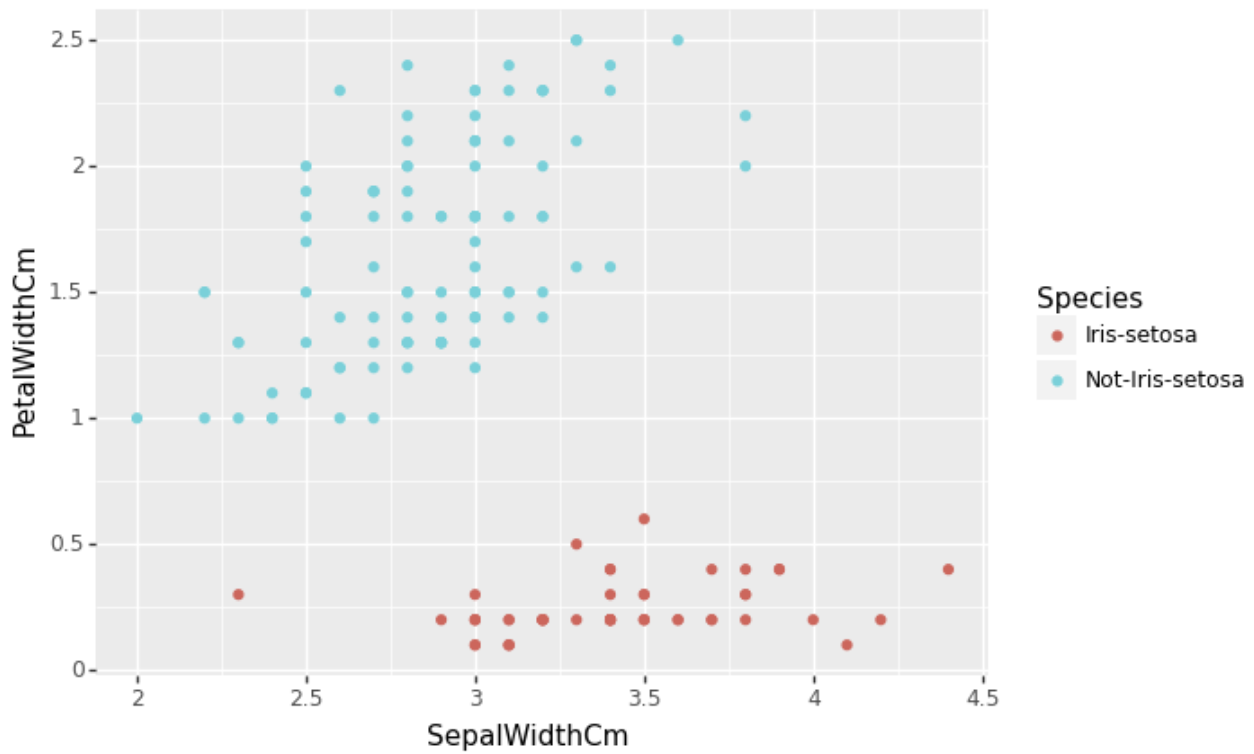
**ANALYSIS / METHOD**

First, I loaded the iris dataset and imported all the necessary libraries, which were pandas, plotnine, and sklearn. I then printed out the dataset to get an idea of the number of predictors and size of the dataset.



The dataset showed 150 rows and six columns. After looking at the data, I chose the variables I would use to predict the target class (whether the species was Iris-setosa or Not-Iris-setosa). The variables I used for this were SepalLengthCm, SepalWidthCm, PetalLengthCm, and PetalWidthCm. Next, I utilized the plotnine library to visualize the relationship between the features and the target class.



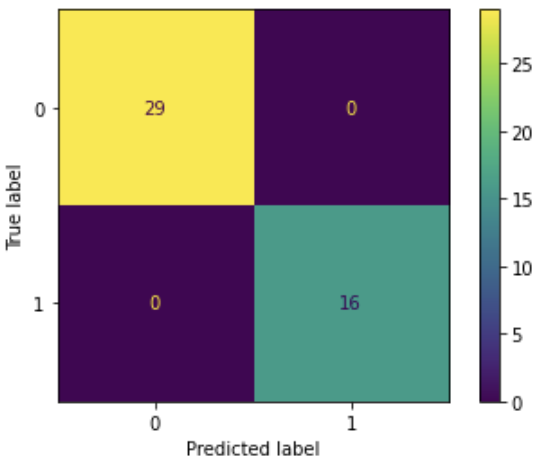
After looking at these plots and seeing potential decision boundaries between predictors, I then decided to begin creating our SVM model. First, in order for our machines to understand the species variable, I converted it from text data to binary data by creating a new column called Species binary and setting the value of each row to 1 if the species is iris-setosa and setting the value to 0 if the species is not iris-setosa. I then split up the data, using 70% of the data to train the model and the remaining 30% to test the new model's performance. After this, I normalized the data by z-scoring it and created the SVM linear model. After training the model with the training set, I had the model predict the data classification in the test dataset.

**Results**

After the model was created, I analyzed how the model performed. Then, I evaluated the performance of my model using accuracy, precision, and recall metrics. Accuracy, precision, and recall are all metrics used to assess the performance of a model.

* **Accuracy** tells us the proportion of correct predictions made by a model out of all predictions made. It is calculated by dividing the number of correct predictions by the total number of predictions made. Accuracy is important because it gives an overall sense of how well a model performs.
* **Precision** measures the proportion of true positives (correctly identified positive cases) out of all positive predictions. It measures the model's ability to make correct positive predictions without too many false ones.
* **Recall** measures the proportion of true positives out of all actual positive cases. It measures the model's ability to identify all positive cases without missing too many. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

The results for each of these metrics were 100%, meaning our model could predict whether or not the species was iris-setosa without making a mistake. To confirm that these metrics are this is correct, I then used a confusion matrix on the test set to visualize the model's output.



As you can see the model predicted both the true negatives (indicated in the 0/0 section of the confusion matrix) and the true positive (indicated by the 1 /1 section of the confusion matrix) perfectly. In the end, these results helped us confirm that we could use an SVM algorithm to classify whether or not the plant is iris-setosa.

**REFERENCES**

Hansen, K., & Zenobia, Kent. (2011). Example short technical report: The benefits of green roofs. Civil Engineer’s Handbook of Professional Practice. https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470950043.app4