Ruby Link

Support Vector Machine to Classify Iris'

Introduction

Support Vector Machines (SVMs) are a popular and highly effective model that can be used for classification. They have the unique ability to find the optimal decision boundary between classes by maximizing the margin while also minimizing overall error. SVM models also contain a hyperparameter, a kernel, that allows non-linearly separable data to be classified by the model. A kernel maps the data to a higher dimension and can then accommodate for more complexity and successfully classify non-linear targets. These characteristics make SVM the ideal choice over other, more simple classification methods like simple linear separators. In this example, an SVM model is used to separate flower types and classify them as either Iris-setosa or Not-Iris-setosa based on four input features.

Data

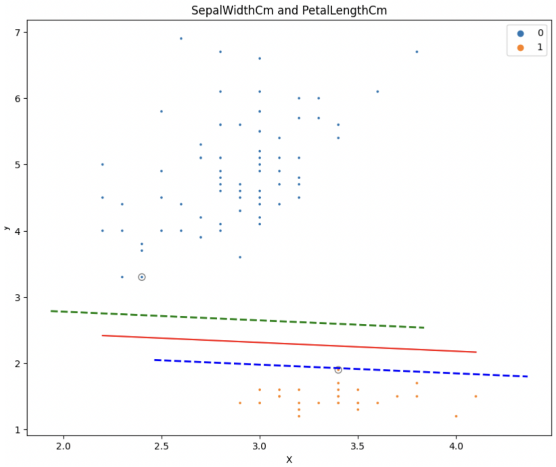
In the iris dataset there are six columns and 150 rows of data. Of the six columns, four are used as features to train the SVM model to accurately classify the target variable. These features include the flower's sepal length (SepalLengthCm), sepal width (SepalWidthCm), petal length (PetalLengthCm), and petal width (PetalWidthCm), all of which are the data type double. These features are all characteristics of a flower that vary between species and will help train the model to correctly categorize each observation. The target variable is Species, a categorical variable representing the type of flower (Iris-setosa or Not-Iris-setosa). Of the 150 observations, 50 of them are Iris-setosa flowers and 100 of them are not. It is important to note that the data is skewed and that there are more Not-Iris-setosa observations for the model to train on. To prepare the data for the model, a transformation was performed to make Species a binary variable. In order to do so, the function get_dummies() was called to change the outcome from Iris-setosa or Not-Iris-setosa to 1 (if Iris-setosa) or 0 (Not-Iris-setosa). The dataset was further cleaned to drop the original Species column and merge the new dummy table with the original table containing the rest of the features.

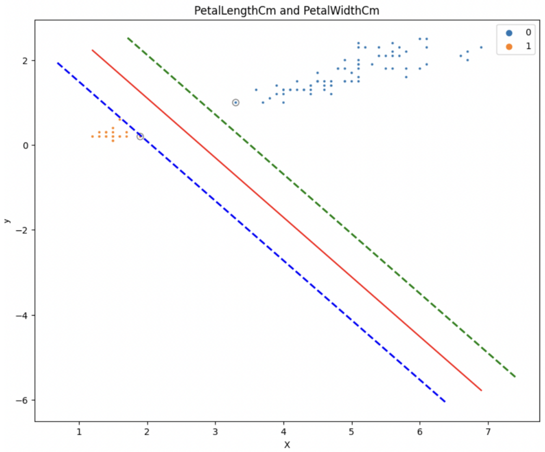
Methods

In order to build the SVM model, the data was explored and cleaned (as mentioned above) in order to understand the number of observations, the distribution of the outcome variable, and to transform Species from a categorical variable to binary. The features and the outcome variable were divided and stored in variables X and y, respectively. The data was then divided into a 70/30 train-test-split with 70 percent of the variables used for training the model and the other 30 percent used as test data to see if the model's predicted classifications match the actual data. An SVM model was then created and the training data was fit to the model. Finally, the SVM model was used to predict the test data. To assess the performance of the model, the accuracy, recall, and precision were calculated (discussed in the results section). Lastly, to visualize the decision boundary, margins, and support vectors, three plots were made using two feature pairs at a time. It is important to notice that not all of the 2D combinations are shown due to reasons discussed below. To make the graphs, x and y points were calculated. X_points is computed by generating 200 points between the max and min for each feature and y_points is calculated using x_points and the weights (w) and bias (b). Using the x_points and y_points, a calculated decision boundary was then plotted onto the graph along with encircled support vectors. A unit vector was calculated (w_hat) along with the margin (margin) which are used to compute the margin lines. Everything is then plotted on a graph and repeated for each of the feature combinations.

Results

To assess the performance of the model, three metrics were calculated: accuracy, precision, and recall. The accuracy reveals how often the classifier is correct, recall shows how effective the model is at identifying all objects of the target class, and precision shows how often a model is correct when predicting the target class. For the SVM model, the results show scores of 1.0 for all three of the performance metrics (100% accuracy, precision, and recall). This indicates that the model is able to successfully classify every point of the unseen (test) data. It is able to perfectly separate and predict the two classes and there are no points crossing over or into the margins, so there is no slack given or needed in our SVM model. However, it is important to note that this perfect score could imply overfitting and potentially not perform as well on data outside this dataset. Another thing to take into consideration is that likely not all of the 2D segments are being used in the model because it finds more accurate classification using a combination of other features. If plotted, some of the 2D graphs would look like the model isn't performing well. The two graphs shown below are likely two of the slices that are used in computing the overall model because as we can see, they are both perfectly linearly separable. Given the perfect score, if we wanted to predict the species of future flowers given information about their sepal and petal length and width (features) this model should in theory be able to successfully classify them into the correct category. We would need to run more unseen data through the model to know this for sure.





Conclusion

SVM is a highly effective model that can be used to classify or predict an outcome given a set of features. For this particular case, the SVM model's ability to not only choose a decision boundary, but the optimal decision boundary ensures that Iris' will not be incorrectly identified as the wrong type, but we do have to take into consideration that there may be overfitting occurring in the model. On the other hand, the decision boundary is placed farthest from the nearest two points of either class (support vectors) which improves the overall generalization of the model. We could run further untrained data through the model to continue to test the accuracy. Regardless, we can conclude that this model is effective overall. It could be useful if implemented at flower shops or for researchers. If unsure of the type of Iris, they could simply measure the sepal and petal length and width, and run the features through the model and an accurate classification should be relayed.