Roro Duran
Assignment 1
LaHaye - CPSC 393

ABSTRACT

Support Vector Machines (“SVMs” or “SVM”) are known for being able to properly classify large datasets. It does so by creating a hyperplane –also known as a decision boundary– that separates distinct classes. The primary objective is to maximize the size of the margin, which is the distance between the hyperplane and the nearest point from each class – while ensuring accurate labeling. To optimize SVM models, several crucial parameters come into play. The 'C' parameter influences the balance between margin size and model accuracy. It can vary from 0 to 1, with lower values prioritizing larger margins and higher values aiming to classify more points correctly. Furthermore, kernels play a pivotal role in handling non-linear data. SVMs employ various kernel types, including the default linear kernel, polynomial kernel, and Radial Basis Function (RBF) kernel. These kernels transform data into higher dimensions, enhancing their suitability for SVM models. The 'gamma' parameter is another critical factor in shaping the decision boundary. Options such as 'scale' (the default setting) and 'auto' are available to fine-tune the SVM model.

INTRODUCTION

This document outlines the process of applying an SVM model to the Iris dataset, specifically modified to distinguish between two distinct outcomes: Iris-setosa and Not-Iris-setosa. It will detail the sequential steps taken to achieve the primary objective: identifying the optimal model for this simplified dataset. Despite its simplicity, this dataset presents challenges such as overfitting due to its small size. This report elucidates the methodology employed to address these challenges and attain a robust classification solution.

DISCUSSION

The analysis commenced by importing essential libraries, including but not limited to pandas, numpy, and scikit-learn metrics. Of particular significance for this study is the 'SVC' class from scikit-learn, which is pivotal for creating and utilizing the SVM model.

Next, the dataset was hosted on GitHub to facilitate remote access to the raw data. This online link was subsequently utilized by the pandas library to import the dataset in CSV format.

Given the presence of two distinct labels, "Iris-setosa" and "Not-Iris-setosa," label encoding was employed to assign numeric values to both categories. Subsequently, the dataset was split into training (70%), validation (10%), and testing (20%) subsets. The initial SVM model was constructed using default parameter values, including 'C' set to 1, 'kernel' set to "linear," and 'gamma' set to "scale." To identify the optimal model parameters, however, a GridSearch and Find Best Params operation was conducted, which revealed that the best-performing configuration involved 'C' set to 1 x 10^(-5), 'kernel' set to "poly," and 'gamma' set to 1.0.

To assess the impact of a small 'C' value on accuracy, scikit-learn functions were utilized to calculate accuracy, recall, and precision scores, all of which yielded perfect scores of 1.0, indicating the model's proficiency in learning the dataset. Additionally, Sum of Squared Errors and Mean Squared Errors were calculated and found to be perfect, confirming the absence of errors between predicted and actual outcomes.

Lastly, a simplified SVM model with a linear kernel and 'C' value set to 1 was developed for ease of visualization, focusing on two independent-variable categories: Petal and Sepal. The model was fitted separately for each variable category, resulting in accurate decision boundaries and margins. The development of a simplified SVM model with a linear kernel and 'C' value set to 1 further demonstrated the suitability of SVMs for this dataset, reaffirming their excellent performance.

CONCLUSION

After achieving perfect results across all aspects, it is possible that the dataset was susceptible to overfitting due to its small size and simplicity. Nonetheless, the model demonstrated exceptional learning capabilities during the training process. A potential solution to mitigate overfitting in future applications could involve training the model on a larger and more complex dataset.

APPENDIX

**Note: Class 0 stands for Iris-setosa. Oppositely, Class 1 stands for Not-Iris-Setosa*