Sean McLean

ALY 6040

Professor Reilly

Module 4

Technique Practice

Introduction

The Module 4 technique practice analyzes a dataset that detects whether a patient has been afflicted with heart disease or not. The analysis of the data is conducted using support vector machines by breaking up the dataset into training and testing samples. The dataset consists of 300 rows and 14 attributes that are all integers except for one that is of a numeric string. Using methods like matrices and plotting of the results will also give a more thorough picture of the data and its findings.

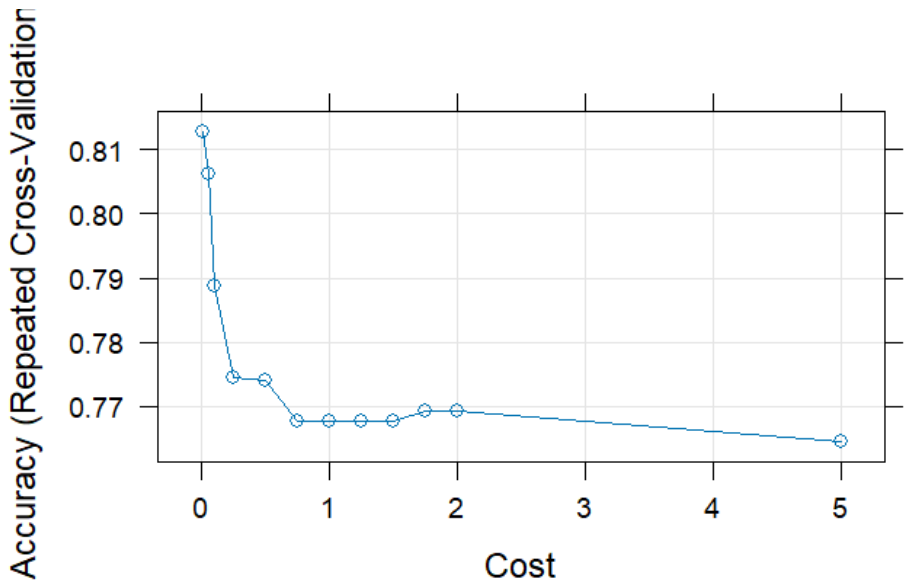
Code Walk Through

The beginning of the coding entails installing the necessary libraries (caret) and packages (kernlab) and importing the dataset and observing the strings and then the head of the data. The caret library and kernlab package will help with the training and plotting of the regression classification and regression models and building support vector machines. The 14^{th} attribute that is the target variable is then split into training and testing sets by using the 'createDataPartition' function with a percentile of 0.7 which provides a 70:30 ratio. These calculations using the 'dim' function show 210 observations in the training group and 90 observations in the testing group. The initial dataset is evaluated with the summary function and there are no null values contained in the entire dataset. The 'set.seed' function is set at 3033 and will provide the data's replicability.

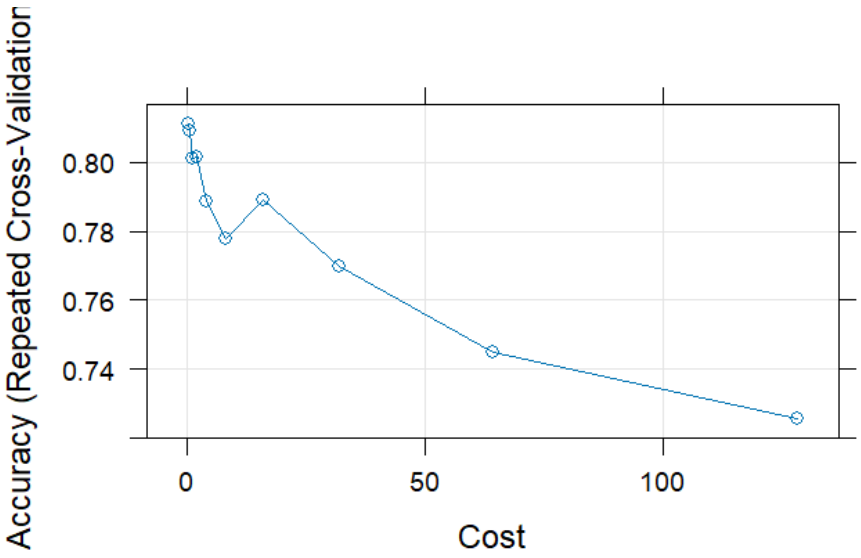
The summary of the target variable shows that all its values are between 0 and 1, indicating that the training data should be converted to a categorical variable by using the 'factor' function. The next step is to train the SVM model by setting up the parameters using the 'trainControl' method and the 'repeatcv' method for the purpose of cross-validation of the values. The number of sampling iterations will be set to 10 and the number of repeats that pertain to sets of folds in the cross-validation process is set to three. When training the SVM classifier with the train method, the 'method' parameter used is 'svmLinear' and includes the target variable V14. The formula 'V14~.' signifies the use of all attributes with V14 as the target. The 'trControl' parameter is then used with outputs from the 'trainControl' method and includes the 'preProcess' parameter for data preprocessing. This procedure standardizes the training data with a mean value near zero and a standard deviation close to one. Additionally, an integer value for the 'tuneLength' parameter is specified to aid in algorithm tuning.

A new variable called 'test_pred' is created that uses the 'predict' function on the parameter 'svmLinear' to predict the results of the model with the C value being trained as one. The method will provide a list of values from the trained model and the testing dataset. A confusion matrix is then made using the 'test_pred' variable and the target variable values from the testing dataset which will be the statistics of the results. From the findings of the matrix, we can generate the 'svmLinear' classifier, and the flexibility to adjust the selection of the C value in the linear classifier according to the requirements. The customization involves entering values into a grid search, and the subsequent code demonstrates the process of constructing and refining

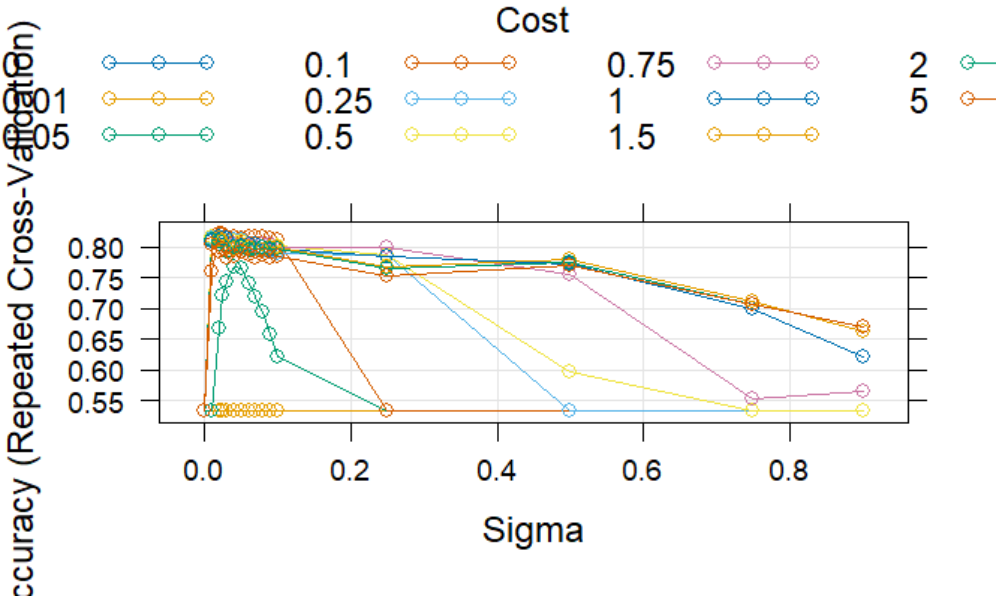
an SVM classifier with varying C values. The ‘grid’ data frame is created with selected C values using the ‘expand.grid’ method. We then utilize the dataset in the ‘train’ method, incorporating the ‘tuneGrid’ parameter to assess the classifier across specific C values. A plot of the grid is then visualized to show the findings of the cost and accuracy that is provided below:



The next step is to construct a model using a non-linear kernel such as the radial basis function. This is conducted by using the ‘test_pred’ variable again on the ‘svm_Linear_Grid’ variable and the ‘newdata’ function for the testing dataset before creating a confusion matrix with the target variable. After employing the ‘set.seed’ function, the RBF kernel is utilized, and the ‘method’ parameter of our ‘train’ method is adjusted to ‘svmRadial’. When using the radial kernel, the suitable values for the ‘C’ parameter are selected and the ‘sigma’ parameter. These values are then calculated, and a plot is created to show the results provided below. From the computed values, the accuracy of the model is evaluated on the test set. To make predictions, the ‘predict’ function is used, specifying the model's parameters as ‘svm_Radial’ and setting ‘newdata’ to testing. From those findings a confusion matrix is made to show the accuracy of the data.



The classifier is then assessed and refined by exploring a range of C and sigma combinations, a process that has made more efficient through grid search. The ‘grid_radial’ data frame will contain the sigma and C values, which will then be assigned to the ‘tuneGrid’ parameter of the ‘train’ method by creating the ‘svm_Radial_Grid’ variable. Following the execution of the SVM-RBF kernel, it analyzes different combinations and identifies the optimal sigma and C values. The accuracy of the trained models on the test set will then be evaluated and the results will be plotted with the results visually displayed below.



From the computed values, the accuracy of the model is evaluated on the test set. To make predictions, the ‘predict’ function is used, specifying the model's parameters as ‘svm_Radial’ and setting ‘newdata’ to testing.

Analysis

The performance of the support vector machine with a linear kernel applied shows after preprocessing and resampling the data has an accuracy of 0.77 and a kappa measure of 0.53. The kappa coefficient evaluates classification accuracy relative to random assignment, ranging from -1 to 1. A 0 implies no improvement over random, negative values indicate worse than random, and those near 1 suggest superior classification (Humboldt, 2014). From those findings a confusion matrix is made to show the accuracy of the data. This would indicate that the results are moderate in their measures of the classes. The confusion matrix based off the ‘test_pred’ variable and target variable shows the accuracy and kappa metrics of the testing data are higher, indicating that the prediction variables have improved the metrics of the dataset. Incorporating different C values into the grid for the training data, the results show that the lower the C value the higher the accuracy and kappa values. The lowest grid value used is 0.01 and provides the highest accuracy and kappa values. A smaller 'C' value results in a more flexible margin, accommodating higher misclassification rates in the training data. However, this adjustment can strengthen the model's ability to generalize to unseen data by minimizing the risk of overfitting.

The confusion matrix statistics that were generated using the ‘svm_Linear_Grid’ classifier that was built show through the process using the training and testing data that the accuracy and kappa values are even higher. The metrics show that most of the accuracy values have reached 0.90 and the kappa is 0.84, indicating that the tuning of the values is working. This is also evident when calculating the support vector machines with radial basis function kernel that through the preprocessing and resampling methods the values of accuracy and kappa are higher as the C values get lower. With a C value of 0.25, the accuracy is 0.81 and the kappa value is 0.62, and the plot shows that the accuracy drops off significantly as the cost increases. The confusion matrix statistics showing the ‘svm_Radial’ variable and the test predictor still show strong accuracy and kappa values but a slight drop-off from matrix statistics of the ‘svm_Linear_Grid’.

The results from the support vector machines with radial basis function function kernal that was constructed indicate that the most optimal model used the largest accuracy value which were a sigma value of 0.025 and a C value of 0.1. The accuracy value generated from the confusion matrix was 0.89, a strong result from tuning the classifiers. These results are validated from the plot that shows a large cluster of data points with high accuracy values and low sigma and C-values.

Interpretations and Recommendations

Overall the goal in support vector machine classifier implementation is to continue to tune the parameters until finding one that gives the highest levels of accuracy and kappa values. By focusing on the target variable, we have managed to find the model that is the most accurate and will allow us to provide stronger predictions on heart disease patients. The usage of the C values is recommended in tuning of the parameters because it will help find the best possible measure of accuracy. There is also the importance of keeping C values as low as possible, so margins stay flexible, accommodating higher misclassification rates in the training data. From the SVM models used on the dataset I would recommend using the support vector machines with the linear kernel model because it had the highest accuracy values and highest kappa values. While some tuning of the parameters could be done with the C values and sigma values, I don't believe you will get much higher than what has been done already.

Conclusion

The SVM models used in the module technique practice provide multiple ways to show how to find the most accurate model. This can be valuable in datasets like the one used in this module where heart disease can be detected at higher rates. Using the proper functions and methods can help fine tune the parameters and classifiers to find the best possible accuracy and kappa values.

References

GSP216 Introduction to Remote Sensing. (2014). *Accuracy Metrics.* Humboldt State University. https://gsp.humboldt.edu/olm/Courses/GSP_216/lessons/accuracy/metrics.html#:~:text=The%20Kappa%20Coefficient%20is%20generated,range%20from%20%2D1%20t0%201.