Homework 2 Report

CSE590-12 Machine Learning

Jonathan Loyd

The following experiments serve as an exercise to practice using cross-validation on a set of data to further analyze that data with machine learning models such as KNN binary classifier, Logistic Regression classifier, and Linear Support Vector Machines (SVM) classifier. The following experiments were conducted on training and testing data for spam that was provided. The data was already spit into training and testing data, so the only splits in the data were for cross-validation. The training data was split into training and validation data using k-fold cross validation, where k is equal to 5. After this, the training and validation data is used to find the optimal parameters of the 3 previously mentioned learning models. Once the optimal parameters are found and chosen, those parameters are used to predict the testing data.

The 5-fold cross-validation method was first applied to the KNN binary classifier, which takes the K nearest samples and finds their classification, then classifies the testing data based on the classification of most of the K samples. The data for validation was used to find the test scores and training scores for each of the 5 folds, and these scores were averaged to provide the data for each K value tested. The K values tested were in a range of 1 to 20 inclusive.

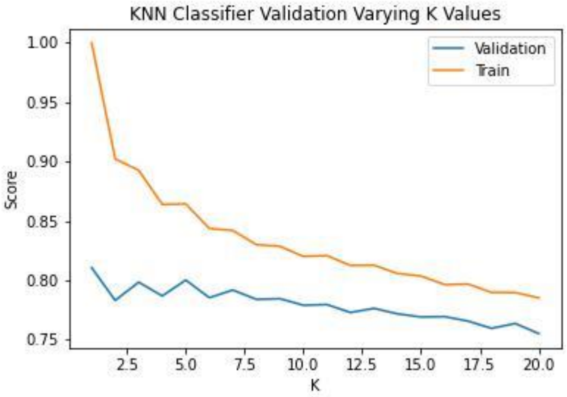

Figure 1: KNN Classifier Validation and Training Scores for Varying K Values

The results shown in Figure 1 indicate that there is an unsteadiness of validation from k values of 1 to 7, then the accuracy slowly drops for both the validation and training scores. The maximum validation score was a value of approximately .8107 at K equal to 1, which also had the best training score. A K value of 1 was used as the parameter for the testing data, and the score for testing was .8028 and the score for training was .9997. These values are extremely close to their validation counterparts. There appears to be some amount of underfitting that is occurring at K equal to 1 that drops off, and there are likely instances of overfitting as K increases since there is a downward trend for both training and validation scores. Using a confusion matrix there were 571 true positives, 103 false positives, 124 false negatives, and 353 true negatives. There was a much larger ratio of false negatives to true negatives than there were false positives to true positives.

The next method was the Logistic Regression classifier, which defines a loss function which is applied throughout the training of the model until the minimum is reached. Logistic Regression also includes regularization, which can be controlled using the C variable using the sklearn library in python. Under this method, decreasing C will increase the amount of regulation on the data by strengthening the Lambda regulator. The C variable was tested on a range of .01 to 1.

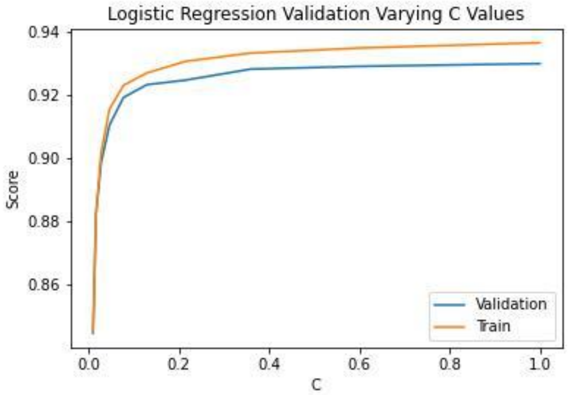

Figure 2: Logistic Regression Validation Varying C Values

The accuracy for both training and validation as seen in Figure 2 seem to remain stable with C values over .2, but smaller values dramatically reduce the performance of both training and validation, which indicates that the model is likely underfitting with more regularization. Smaller C values are also indicative over underfitting as validation and training scores both plummets.

The Logistic Regression model performs best with a C value of 1 on the validation data, with a validation score of approximately .9299 and a training score of approximately .9345. Since a C value of 1 performed the best, it was used as the optimal parameter for the testing data. The score for testing was approximately .9227 and the score for training was approximately .9357. The scores for validation and testing were extremely similar, as well as the training scores between validation and testing. There are no coefficients that hit exactly zero, but some coefficients are brought more toward zero than others. Since there are no coefficients that are 0, all features have some amount of importance to the Logistic Regression model to perform well. As C is decreased to a value of .0001, using l1 regularization, the features ‘capital_run_length_longest’ and ‘capital_run_length_total’ are the only coefficients that are nonzero, indicating that they play a large role in some amount of the performance of the Logistic Regression model. Using a confusion matrix on a testing C value of 1, there were 637 true positives, 37 false positives, 52 false negatives, and 425 true negatives. There was a much larger ratio of false negatives to true negatives than there were false positives to true positives.

The last model to validate and test was Linear SVC, which works to create a boundary to separate the two classes of spam and not spam using a cost function. Optimization of this model is to maximize the margin from the boundary to samples on either side to have lower generalization error and prevent overfitting. This model includes regularization using a variable

C. Initial validation was conducted on a range from 1x10^{-5} to 1x10^{5} that shows performance was best at some lower values from .01 to .1. This performance is shown below in Figure 3.

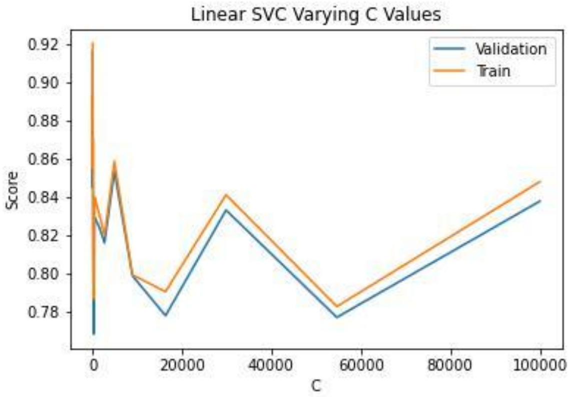

Figure 3: Testing Large Range for Linear SVC C Values

Figure 3 highlights the dramatic underfitting that occurs at values closer to 1x10^{-5}, and the change in performance that occurs as C increases. The results of Figure 3 lead to some further, more fine-tuned testing on values of C in the range of .01 to .1, which are shown in Figure 4.

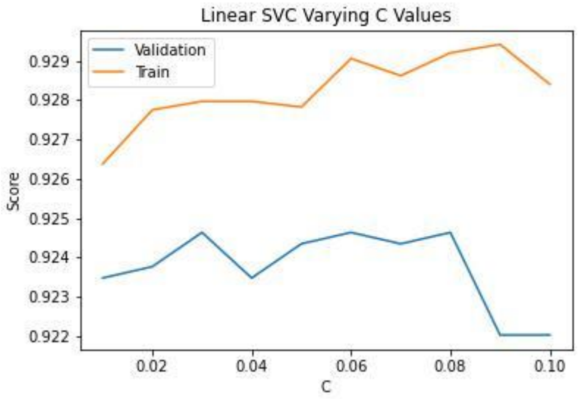

Figure 4: Testing More Optimal C Values

Figure 4 shows best performance at C equal to .03 for validation data, with a validation score of .9249 and a training score of .9276. The C value of .03 was chosen as the best parameter on the validation and was used for conducting testing. The testing data for Linear SVC had a testing score of .9183 and a training score of .9278. The training scores are almost identical, and the testing score was only slightly worse than the validation scores. This is an indication that the validation data and methods were good representations of the data. Using a confusion matrix on the testing C value of .03 there were 636 true positives, 38 false positives, 57 false negatives, and 420 true negatives. There was a larger chance for the model to falsely predict a negative than it was to falsely predict a positive.

With all the methods, there was a trend with their misclassification, and they were more likely to misclassify a sample as falsely negative than they were to classify something as falsely positive. This may be due to there being less data on negatives, but this would require further analysis.

All the methods performed well on the given spam data, but the methods have their strengths. The KNN binary classifier has the worst performance, but it was by far the quickest to train and the easiest to comprehend and explain. Logistic Regression was slower to figure out ranges of C that were beneficial during validation than KNN, but it was marginally faster than Linear SVC in terms of validating and testing. Logistic Regression also had the best performance in terms of accuracy, and it is slightly easier to explain and comprehend than Linear SVC. Finally, Linear SVC could have its performance improved with more time and processing power, so it could slightly outperform the other two models, but it is the most complex model and by far took the most amount of time for validation and testing. With all of this in mind, the best model out of the 3 tested appears to be Logistic Regression, due to its accuracy, timeliness, and the ease of explanation.