Seng 474 Assignment 1 Report

Shayla Grymaloski
V00884262

Introduction

In this assignment, I used the data provided to conduct experiments using svm and logistic regression training models. Due to the length of running the svm gaussian experiment, I trimmed the dataset down to include 1000 data points of class 5 and 1000 data points of class 7. This size of data was used throughout all of my experiments.

Task 1

This first task explored the machine learning model, logistic regression. Specifically, this task looked at the parameter value C that can be used in this model and showed both underfitting and overfitting. Red is showing the test error percentage with each value of C and blue is showing the train error percentage with each value of C.

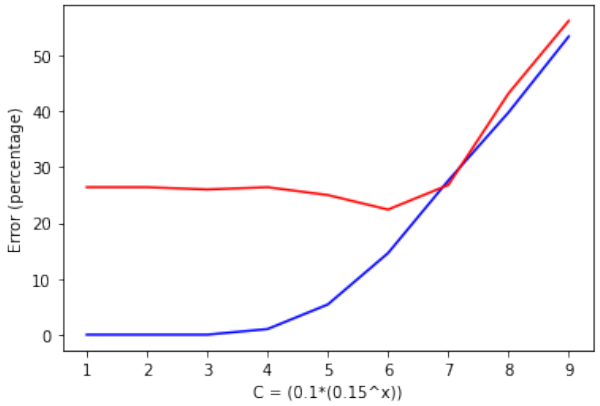

Figure 1, Logistic Regression with changing C values

In this experiment, we see overfitting and underfitting occurring. The overfitting is caused because to value of c is too high, meaning the regularization value is too high. When changing the value of x in the equation of C = Cnot*(alpha)^ x, we can see that overfitting occurs from the x value of 1 to the x value of around 6. We can see that the test error is performing almost perfectly, however it is not generalizing the data. As the regularization value is decreasing, it is helping the training data generalize the information. Underfitting occurs when the training data has not learned enough to generalize the data. You can see this happening around the x value of 7.

Task 2

This second task looked at exploring the machine leaning model support vector machine. Similar to the first task, this looked at the parameter value C used the model and looked at showing examples of both overfitting and underfitting. Red is showing the test error percentage with each value of C and blue is showing the train error percentage with each value of C.

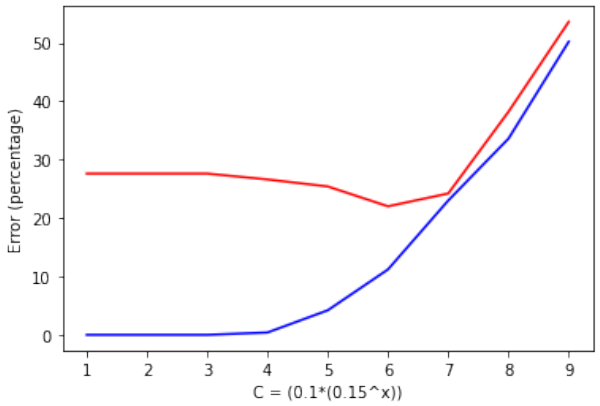

Figure 2, SVM with changing C values

In this experiment, we see overfitting and underfitting occurring. When changing the value of x in the equation of C = Cnot*(gamma)^ x, we can see that overfitting occurs from the x value of 1 to the x value of around 6. The training data here is being too specific and not generalizing for the whole dataset. Underfitting, looks like it may be happening around the value of 7. The data here is not being trained enough and points are missed when trying to generalize.

Task 3

Task three looks at finding the most performant regularization parameter for both logistic regression and support vector machines. To explore and find the most optimal regularization parameter, K-fold cross validation was performed with a k-fold equal to 5. Below are the results of using K-fold cross validation with logistic regression and support vector machines with varying values for C that range around the most optimal result.

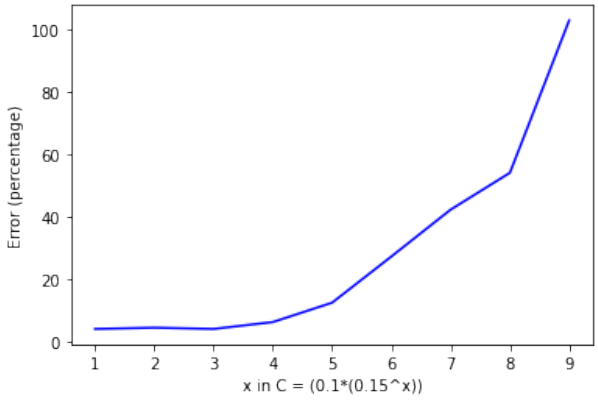

Figure 3, Logistic regression with k-cross validation

From this experiment of logistic regression and I again used k-fold cross validation with the k-fold value of 5. I found that the minimum mean squared error that I could produce was 4.2%. This occurred at the x value of 3 making the value of C = (0.1)*(0.1^3) for the least error prone parameter to use.

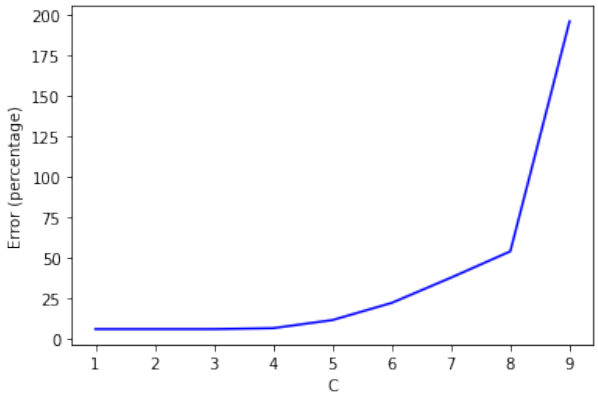

Figure 4, SVM with k-cross validation

After running the k-cross validation with SVM, changing the value of c, I found that the smallest error came to be 5.8% and occurred at the value of x equal to 2 making the value of C = (0.1)*(0.1^2).

Running k-fold cross validation on both logistic regression and svm, I found that in this case, logistic regression was able to get the lowest percentage error. Next, I trained each optimally-tuned method on the entire training set and report the method's test error.

The best performing values of C for both logistic regression and support vector machines was 27.6 percent for logistic regression and 26 percent for support vector machines. In this case logistic regression performed slightly better that the support vector machine method. These minimum test errors are very close in percentage of each other. As well when finding the minimum error value for C in the k-fold cross validation part of this experiment, the percentages of 4% and 5% were close in value to one another as well.

Task 4

This task it to look at svm with the gaussian kernel. For each value γ in a good range of values of the scale parameter, I used k-fold cross-validation on the training set to select the optimal regularization parameter Cγ. I chose to run the k-fold cross validation on the k value of 5. The five values of γ I chose for this experiment were 1e-8, 5e-8, 1e-7, 1e-6, 1e-5, 5e-5. I chose these values because I wanted small gamma values to look at as smaller gamma values should perform better. First, I looked at training the model with each of these gamma values, varying the value of C, to find the best regularization parameter.

From this experiment, the lowest percentage error will be used with the gamma value with the whole test and training data set. The table below shows each gamma value with its corresponding optimized C value with the error percentage that was found when running k-fold cross validation on the five gamma values with the k-fold value of 5.

Next, I ran each of these gamma values with their optimized regularization parameters for the whole training and test set of my data. The figure below shows the results of this experiment. The red points are the test data and the blue points are the training data. Note in order to make the points readable and spread out the data points of gamma, I used the index value of each element of the array of gamma values.

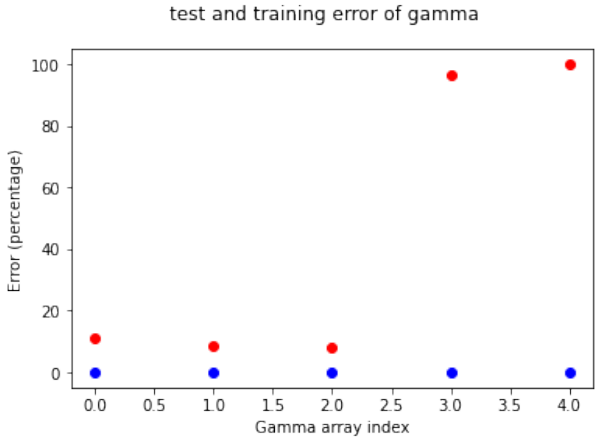

Figure 5, Gamma values with best C value

The minimum error found with the gamma values was 8.1%. When comparing the test value results from using gaussian instead of linear with svm, I found that linear performed better. This is not expected as running a svm model with a gaussian kernel should give the best results due to the nature of its structure comparted to a straight linear line when classifying the data. The gaussian kernel should be better at fitting the data. However, the values of 8.1% for gaussian and 5% for linear show that in the experiments they were close and that there could have been

some errors in the testing. A different k-fold value could be used instead possibly to fix this problem