Machine Learning LAB 3 Report - Classification and Regression

Submitted by

1. SAJID KHAN – 50248743
2. VINAY VARDHAMAN – 50248852

Logistic Regression

Traditionally, Logistic regression is used for binary classification. Since we are classifying hand-written digit images, binary logistic regression classifier is simply not enough. So we use the one vs all strategy to classify each digit or class against the remaining classes. In particular, we use 10 binary classifiers (one for each class) to distinguish a given class from all other classes.

Accuracies recorded:

1. Training set Accuracy: 84.94
2. Validation set Accuracy: 83.72
3. Testing set Accuracy: 84.17

Respective Errors:

1. Training set Error: 15.06
2. Validation set Error: 16.28
3. Testing set Error: 15.83

Analysis:

Logistic regression classifiers work well on data with low input features. They are vulnerable to overfitting with more input features. Since MNIST data has input features close to 700, any considerable change in the input test data will result in large variations across the predictions.

Due to this outlier susceptibility, the accuracies recorded are only around 84 percent. In this example, both training and test set accuracies are close to 84 percent.

Multi-class Logistic Regression

Like previously mentioned, Logistic regression is used for binary classification. We can extend logistic regression to solve multi-class classification. In particular, we do not need to build 10 classifiers. We only need to build one classifier which can classify 10 classes at the same time.

Accuracies recorded:

1. Training set Accuracy: 93.45
2. Validation set Accuracy: 92.50
3. Testing set Accuracy: 92.55

Respective Errors:

1. Training set Error: 6.55
2. Validation set Error: 7.50
3. Testing set Error: 7.45

Analysis:

The accuracies recorded for both training and test sets are close to 92 percent.

Multi-class Logistic Regression vs One vs All Strategy

Regression Type	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**
Binary	84.94	83.72	84.17
Multinomial	93.45	92.50	92.55

It can be observed that Multinomial Logistic regression performed much better than Binary regression. This is because multinomial classifier tries to learn the classes directly. In this way, the parameters for each class are estimated interdependently and the model built may be more robust against outliers.

Support Vector Machines

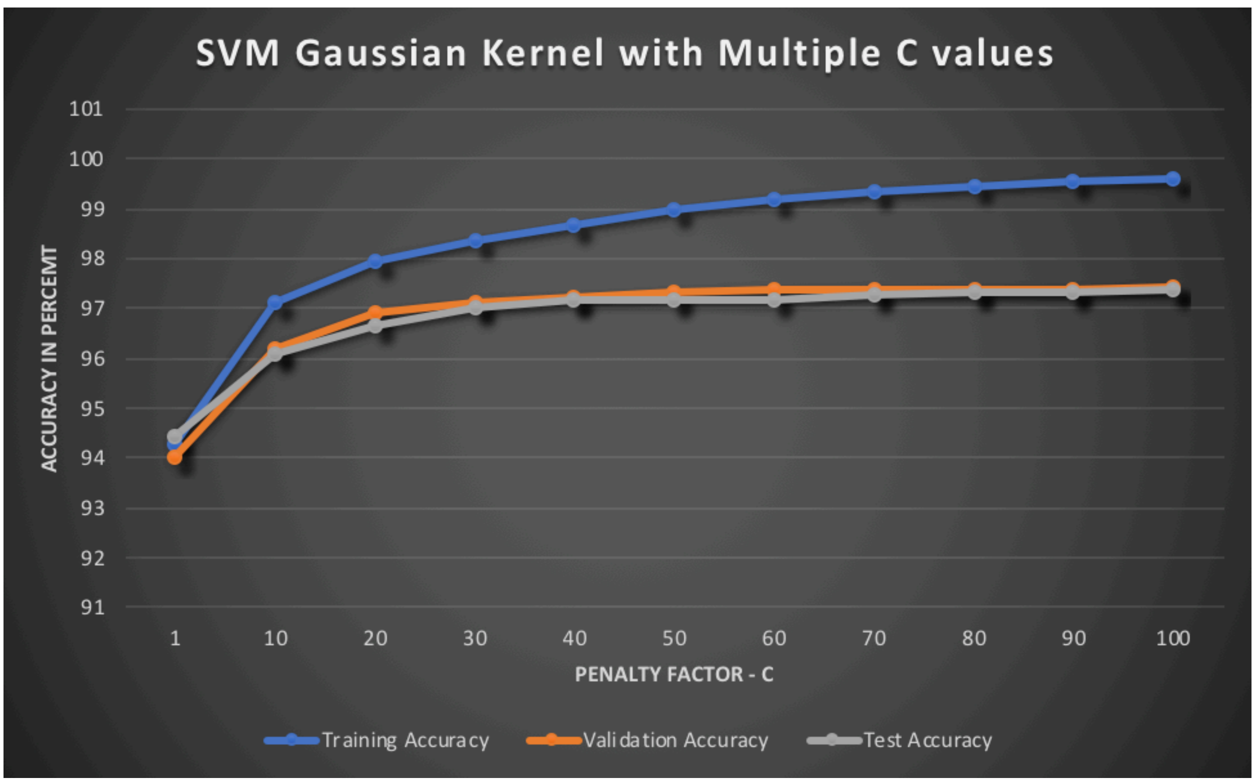
In this part of the assignment, we use Support Vector Machines tool in sklearn.svm.SVM to perform classification on the data set.

Kernel	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**
Linear	97.29	93.64	93.78
RBF with gamma 1	100	15.48	17.14
RBF with gamma 0	94.29	94.02	94.42
RBF with C = 1	94.29	94.02	94.42
RBF with C = 10	97.13	96.18	96.10
RBF with C = 20	97.95	96.90	96.67
RBF with C = 30	98.37	97.10	97.04
RBF with C = 40	98.70	97.23	97.19
RBF with C = 50	99	97.31	97.19

RBF with C = 60	99.19	97.38	97.16
RBF with C = 70	99.34	97.36	97.26
RBF with C = 80	99.43	97.39	97.33
RBF with C = 90	99.54	97.36	97.34
RBF with C = 100	99.61	97.41	97.40

The table shows various results for Support Vector Machines with Linear Kernel and Radial Basis Function(RBF) with different gamma setting and varying penalty factor C.

It can be observed that RBF kernel with gamma setting 0 performs better than Linear Kernel for the test data. With gamma setting 1, though the training accuracy is 100 percent, test accuracy prediction is very low. This happens due to overfitting of the training data.



The plot above shows different accuracies plotted for varying penalty factor – C.

The value of C controls the impact of margin and the margin error. From the plot, we can observe that accuracy increases with value of C. The effect of having lower C value means the weight of error term is low and a larger margin hyperplane is created. Similarly, the impact of higher C value is the increase in the weight of each error term and a smaller margin hyperplane. Thus accuracy increases with value of C. But choosing very high value can overfit the data, like choosing smaller value can underfit. So better to choose an optimal value based on previous results.

Compare Linear and Radial Basis Function Kernel

Kernel	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**
Linear	97.29	93.64	93.78
RBF with gamma 1	100	15.48	17.14
RBF with gamma 0	94.29	94.02	94.42

SVM with Gaussian kernel have the flexibility to transform the data into space of any dimension. And because of this reason, we choose Gaussian kernel when there are more observations than features. Conversely, we prefer Linear kernel when there are more features than observations.

Gamma value controls the impact of training data on the learned hyperplane. When gamma is 1, training data can overfit the model resulting in low test accuracies. Clearly, that is what we observed in the output results.

When gamma is 0, results are better because the model closely relates to a Linear kernel.