CSE 574 - Introduction to Machine Learning

Classification and Regression

Group Number: 48

Kirti Hari - 50208065

Shruti Kulkarni - 50207124

Shashank Suresh - 50208025

PROBLEM 1 REPORT: Logistic Regression

Problem Statement:
Binary Logistic Regression

Logistic Regression is a probabilistic classifier. It can also be a classic Discriminative Classifier model. Thus, they can learn faster and with less data.
The classifier build learns 10 Binary classifiers, i.e, one for each of the output classes, so that each class can be distinguished from all other classes.
The probability of the input data point belonging to one class is calculated for each of the classes and the class that has the highest probability is taken as the label for the data point.

Observations:

Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
84.898	83.7	84.16

Problem Statement:
Multi-Class Logistic Regression

Multiclass Logistic Regression is an extension of the Binary Logistic Classifier which learns the weight matrix "w" for the 10 different classes, with 10 different weight vectors and then uses it to predict the training data.
For the prediction, we have implemented the function "mlrPredict" which does the prediction similar to the one done in the function "blrPredict" i.e., it calculates the posterior probabilities of the data point with respect to each class and takes the class that gives the highest probability. It then labels it as the class for that data point.

Observations:

Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
93.448	92.48	92.55

On comparing the performance of multi-class logistic regression to the performance of logistic regression when using the one-vs-all strategy, the multi-class logistic regression provided better accuracies for all the 3 data sets.

PROBLEM 2 REPORT: Support Vector Machines

Problem Statement:
Using linear kernel (all other parameters are kept default)

Observations:

Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
97.286	93.64	93.78

Inference:
Linear SVM performs well as the given data is linearly separable and is highly dimensional.

Problem Statement:
Using radial basis function with value of gamma setting to 1 (all other parameters are kept default)

Observations:

Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
100	15.40	17.14

Inference:
We can see that when gamma is set to 1, it over fits. The training error is zero but accuracies on the validation and test data set are very low.

Problem Statement: Using radial basis function with value of gamma setting to default (all other parameters are kept default).

Observations:

Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
94.294	94.02	94.42

Inference:
The gamma parameter specifies how a single training example will affect the accuracies. Low value of gamma indicates high influence while high value indicates low influence. We see that, gamma = default performs better than gamma = 1 but not as well as linear SVM.

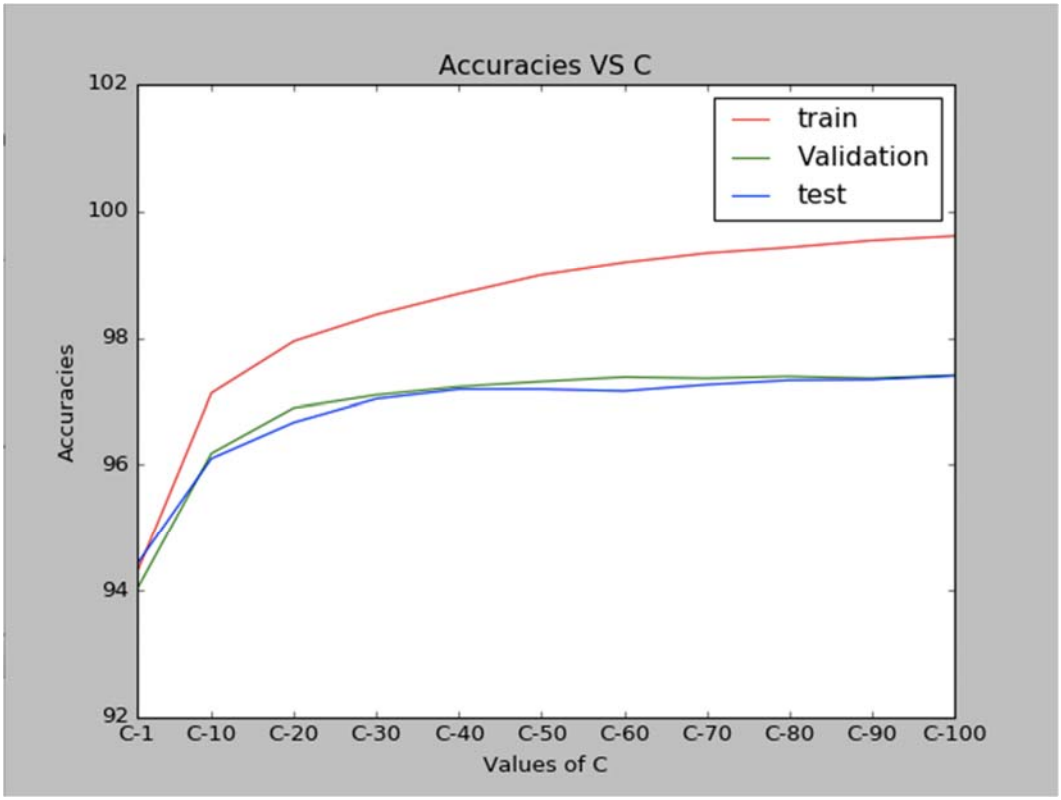
Problem Statement:

Using radial basis function with value of gamma setting to default and varying value of C (1, 10, 20, 30, · · · , 100)

Observations:

C	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
1	94.294	94.02	94.42
10	97.132	96.18	96.1
20	97.952	96.9	96.67
30	98.372	97.1	97.04
40	98.706	97.23	97.19
50	99.002	97.31	97.19
60	99.196	97.38	97.16
70	99.34	97.36	97.26
80	99.438	97.39	97.33
90	99.542	97.36	97.34
100	99.612	97.41	97.4

Graph:



Inference:
The C value helps in extending SVM to data that is not linearly separable. It incorporates an error penalty which specifies to what extent you want to avoid misclassifying each training example. Though this helps in increasing the accuracy of training data, it can lead to over fitting.
As C increases,

- Training accuracies initially increase but the growth towards the end is very slow
- Validation and test accuracies remain almost the same and convergence towards the end.

Linear Kernel VS Radial Basis Function Kernel
Linear kernel is faster to train and test when compares to RBF kernel. Linear kernel is preferred if the data is highly dimensional and informative i.e. has more features. RBF kernel, on the other hand, is preferred when the number of observations of the data are greater than the number of features.

It's been shown that the linear kernel is a degenerate version of RBF, hence the linear kernel is never more accurate than a properly tuned RBF kernel. This can be confirmed for our above observations, though linear kernel provides an accuracy of 93.78% on test data, RBF kernel gives a test accuracy of 97.4% when C is set to 100