CSE 574

INTRODUCTION TO MACHINE LEARNING
PROJECT ASSIGNMENT – III
CLASSIFICATION AND REGRESSION

CSE 574 Group 14
Anuradha Ashavatha Rao – aashavat
Nagadeesh Nagaraja – nagadees
Nagesh Rao – nageshra

Logistic Regression

Logistic Regression is a predictive analysis method which measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function.

Binary Logistic Regression: Binomial logistic regression deals with situations in which the observed outcome for a dependent variable can be classified in only two possible ways. We have ten classes (digits 0-9), which is predicted by the output ten binary classifiers.

Accuracy:

Training set Accuracy	85.676%
Validation set Accuracy	83.57%
Testing set Accuracy	85.13%

Inference:

When compared to other algorithms that we have used so far, the accuracy is reduced when binary logistic regression is used.

Direct Multi-class Logistic Regression: Logistic regression can be extended to solve multi-class classification. Multi-class or multinomial logistic regression can be used to predict outcomes where the output value can have three or more possible types. In this model, only one classifier needs to be built, to classify n classes.

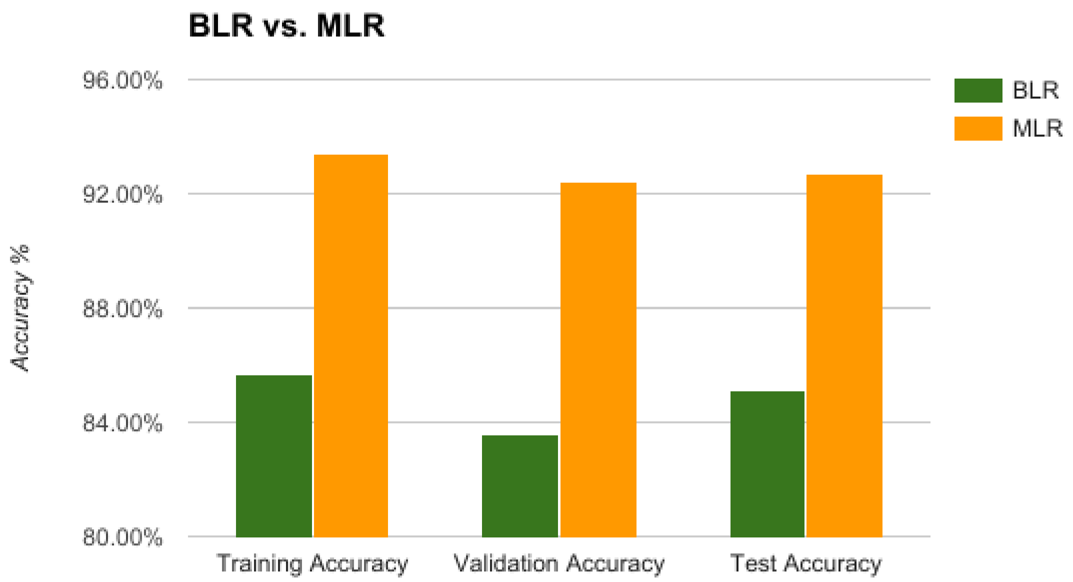
Accuracy:

Training set Accuracy	93.39%
Validation set Accuracy	92.43%
Testing set Accuracy	92.67%

Multi-class logistic regression vs binary logistic regression

From the results observed above, we can see that multi-class logistic regression performs better than binary logistic regression. The reason can be attributed to the fact that the MLR considers all the classes while updating the weights of each class, whereas, BLR updates the weights independent of other classes.

Blr vs. Mlr Accuracy:



Support Vector Machines

Support vector are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The algorithm assigns new samples to the categories by training the already existing samples.

1) *Gamma = 0, C = Default, Kernel = RBF*

Training set Accuracy	94.294%
Validation set Accuracy	94.02%
Testing set Accuracy	94.42%

2) *Gamma = 0, different C values*

C	Training Accuracy	Validation Accuracy	Test 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%

3) *Gamma = 1*

Training set Accuracy	100.0%
Validation set Accuracy	15.48%
Testing set Accuracy	17.14%

4) *Using linear kernel* :

Training set Accuracy	97.286%
Validation set Accuracy	93.64%
Testing set Accuracy	93.78%

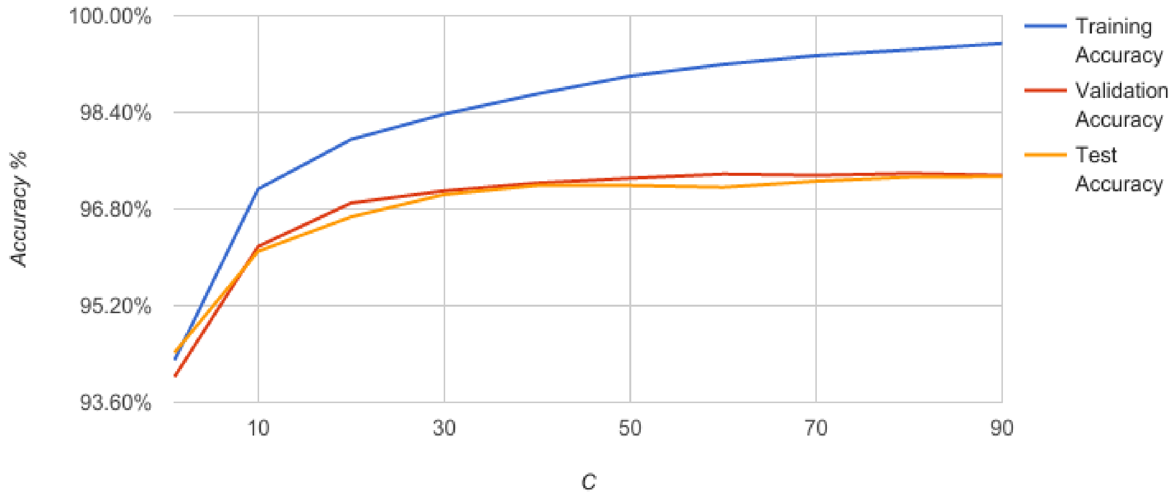
5) *Gamma = Default, Radial Basis Function(RBF)* :

Training set Accuracy	94.294%
Validation set Accuracy	94.02%
Testing set Accuracy	94.42%

We can see that this model gives us the best accuracies, as compared to all the previous models that we have learnt (Neural Networks, Perceptrons, etc.).

The hyper parameter C rectifies the misclassification. (Meaning, higher the value of C, higher the penalty given on the errors). Thus, we could choose C=100, as it gives best results.

Plot of C vs. Accuracies



Linear Kernel vs. Radial basis Function Kernel

Different kernel functions perform differently depending on the data. Linear kernel is a degenerate version of RBF. Thus, RBF is more accurate than the linear kernel. However, if the number of features is large, non-linear mapping does not much improve the performance.