CSE 4/574

Introduction to Machine Learning

Fall 2022 12/12/2022

Classification and Regression

TEAM MEMBERS

MEMBER 1: AKANKSH GATLA

UBIT NUMBER: 50465101

UBIT NAME: akankshg

MEMBER 2: VAISHNAVI ANIL BHIDE

UBIT NUMBER: 50465155

UBIT NAME: vanilbhi

Binary Logistic Regression(BLR):

Set	Accuracy	Error
Training	92.746	7.254
Validation	91.42	8.58
Testing	91.97	8.03

We can see from the above training that our training error has lower value than the test error. Hence, we can infer that our Linear model has better performance on known data when compared to unknown data which is the user case for linear model.

Multi-class Logistic Regression (MLR):

Set	Accuracy	Error
Training	93.22	6.78
Validation	92.21	7.79
Testing	92.76	7.24

We can see from the above training that our training error has lower value than the test error. Hence, we can infer that our Linear model has better performance on known data when compared to unknown data which is the user case for linear model. We get similar errors because we are using the same pattern of data

Performance difference between multi-class strategy(MLR) with one-vs-all(BLR) strategy:

Set	MLR Accuracy	BLR Accuracy
Training	93.22	92.746
Validation	92.21	91.42
Testing	92.76	91.97

- In contrast to one-vs-all (BLR), which only classifies one class with regard to all others at a time, multiclass logistic regression classifies all ten classes of the MNIST dataset simultaneously, making it less time-consuming and less likely to result in overlapping classes.

- We discovered that the multiclass classification was more accurate than the BLR classification. As they are calculated individually due to multiclass parameter's , which helps to prevent incorrect categorization.

Support Vector Machine (SVM):

I. **Using Linear Kernel:**

Set	Accuracy
Training	92.814%
Validation	91.53%
Testing	91.86%

Therefore, since the results are nearly identical to those of the prior linear model we trained, we may conclude from the findings above that the Linear Kernel behaves like a linear model.

II. **Radial Basis Function:**

(a) Using Radial Basis Function (Gamma = 1)

Set	Accuracy
Training	100.0%
Validation	13.46%
Testing	14.91%

As we can deduce from the 100% Training accuracy, this option produces extremely subpar outcomes on test data since the high gamma value contributes to overfitting the training data.

(b) Using Radial Basis Function with value of gamma setting to default (all other parameters kept as default)

Set	Accuracy
Training	92.01
Validation	92.10%
Testing	92.53%

(c) Using Radial Basis Function with value of gamma (default) and by varying value of C (1, 10, 20, 30 upto100)

The best setting for the test and entire dataset with the configuration. C will tell us the weight to give us the so-called slack variable. There is a trade off between margin width c value there.

Below are the results for different values of C on Training, Testing and Validation data:

C	Training Accuracy	Validation Accuracy	Testing Accuracy
1	96.586%	96.19%	96.39%
10	97.364%	96.23%	96.47%
20	97.374%	96.789%	96.97%
30	97.374%	96.789%	96.97%
40	97.374%	96.789%	96.97%
50	97.374%	96.789%	96.97%
60	97.374%	96.789%	96.97%
70	97.374%	96.789%	96.97%
80	97.374%	96.789%	96.97%
90	97.374%	96.789%	96.97%
100	97.374%	96.789%	96.97%

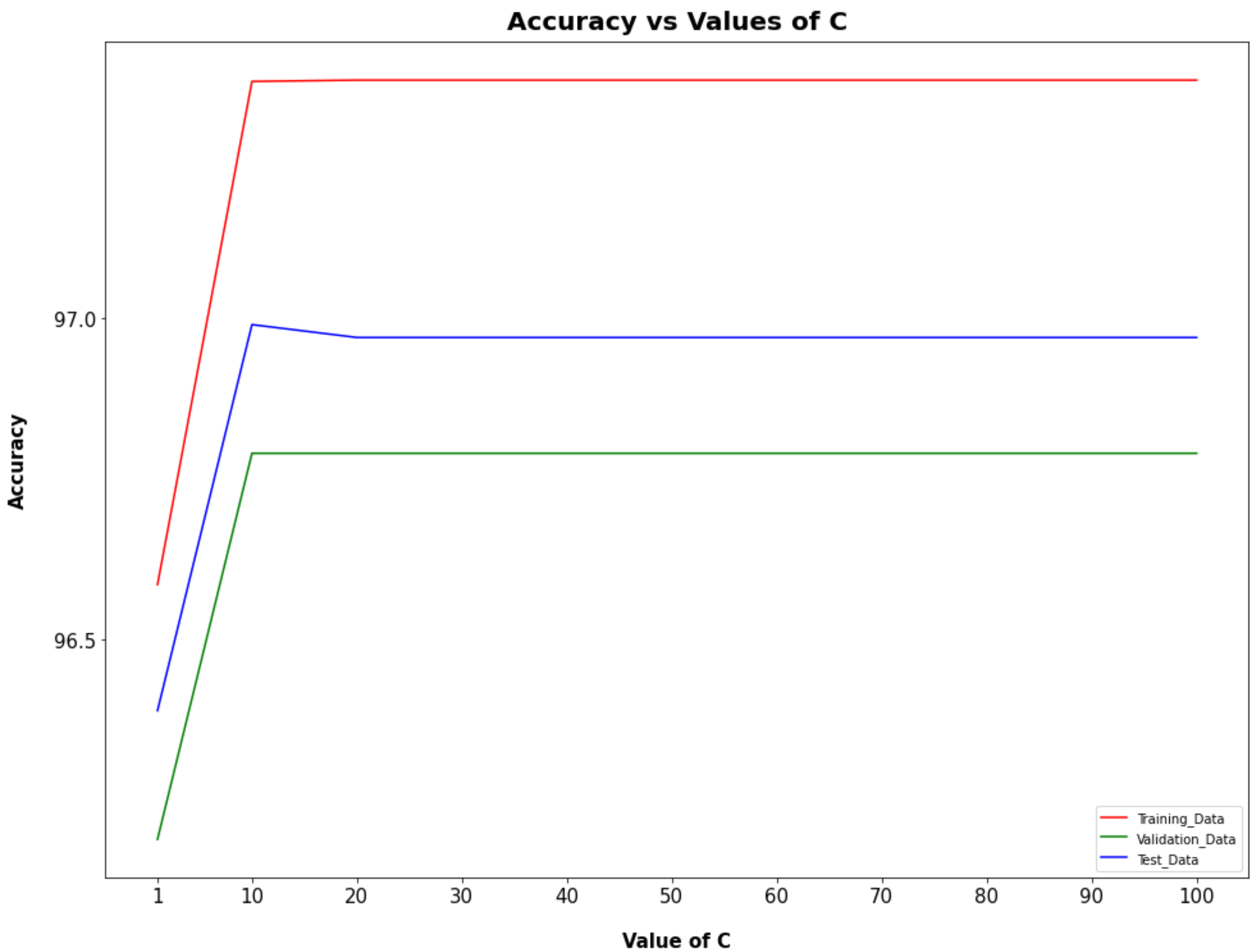
Hence, we can conclude that we are getting the best result by setting gamma = default and C = 20

Results for the whole dataset using optimal parameters:

Kernel	C	Training Accuracy	Validation Accuracy	Testing Accuracy
RBF(Gamma=default)	80	99.34	97.36	93.17

Plot of accuracy obtained on each dataset (Training, Testing and Validation)with

different values of C:



From the below graph, we can infer that our dataset is non-linear because we get better results on this model which is non-linear. Our dataset performs better with this non-linear model, we may also assume that it is non-linear.