PROGRAMMING ASSIGNMENT 3 – CLASSIFICATION AND REGRESSION

CSE 574
Fall 2021 - Introduction to Machine Learning

Group Number : 15

Submitted By :

S.No	Name	Email	UB Id Number
1	Lavanya Nadanasabapathi	lnadanas@buffalo.edu	50367936
2	Aravind Balakrishnan	balakri2@buffalo.edu	50412479
3	Anirudh Sowmyanarayanan	asowmyan@buffalo.edu	50415686

Binary Logistic Regression (BLR) :

Type	Accuracy	Error
Testing	91.94%	8.06%
Training	92.66%	7.34%
Validation	91.46%	8.54%

These are the results after running Binary Logistic Regression on Testing, Training, Validation Data.

- We observe that the Training error is less than the Testing error.
- We can infer that this Linear model performs better on the seen data, but when it gets the unseen dataset it gives a little more error.

Multi-class Logistic Regression (MLR) :

Type	Accuracy	Error
Testing	92.51%	7.49%
Training	93.17%	6.83%
Validation	92.46%	7.54%

These are the results after running Multi-class Logistic Regression on Testing, Training, Validation Data.

- We observe that the Training error is slightly less than the Testing error.
- We can infer that this Linear model performs better on the seen data, but when it gets the unseen dataset it gives a little more error.

Performance between MLR and BLR:

Type	MLR	BLR
Testing Accuracy	92.51%	91.94%
Training Accuracy	93.17%	92.66%
Validation Accuracy	92.46%	91.46%

- In multiclass logistic regression we classify all the ten classes of MNIST dataset at once, whereas in BLR we only classify one class with respect to all other at a particular given time, so multiclass has less time complexity and has less chances of overlapping.
- It is observed the accuracy of the Multiclass was better than the BLR classification. That's because parameters are estimated independently in multiclass which helps to prevent wrong classification.

Support Vector Machine :

Linear Kernel :

Type	Accuracy
Testing	91.41%
Training	92.37%
Validation	91.49%

So, we can infer from the above results that the Linear Kernel operates like a linear model, as the results are very similar to the previous linear model we trained.

Radial Basis Function :

1. RBF when Gamma = 1

Type	Accuracy
Testing	19.03%
Training	100.0%
Validation	17.96%

This particular setting gives bad results on the test data as this high value of gamma helps in overfitting the training data and we can get that from the 100% Training accuracy.

2. RBF when setting Gamma to default :

Type	Accuracy
Testing	92.23%
Training	92.04%
Validation	91.97%

This particular setting gives good results on the test data as the gamma is set to default.

3. RBF when Gamma is default but setting different values for C (C=1,10,20,30,40,50,60,70,80,90,100):

We iterate through the C values and record the optimum setting and then we test the entire data on that setting. This C variable controls the importance we are giving to the Slack variable.

Below are the results for Different Values of C on Testing, Training and Validation data:

C	Testing Accuracy	Training Accuracy	Validation Accuracy
1	96.26%	96.535%	96.17%
10	96.93%	97.224%	96.86%
20	96.91%	97.223%	96.85%
30	96.91%	97.223%	96.85%
40	96.91%	97.223%	96.85%
50	96.91%	97.223%	96.85%
60	96.91%	97.223%	96.85%
70	96.91%	97.223%	96.85%
80	96.91%	97.223%	96.85%
90	96.91%	97.223%	96.85%
100	96.91%	97.223%	96.85%

We can conclude that we get the best results when gamma is set to default and C =10.

Kernel	C	Testing Accuracy	Training Accuracy	Validation Accuracy
RBF (Gamma is set to default)	10	97.23%	97.421%	96.89%

Accuracy obtained on each of the training, testing and validation with different values of C is plotted below :