Report for

CSE 474/574 Introduction to Machine Learning Programming Assignment 3

Classification and Regression

**Course Number**: CSE 474

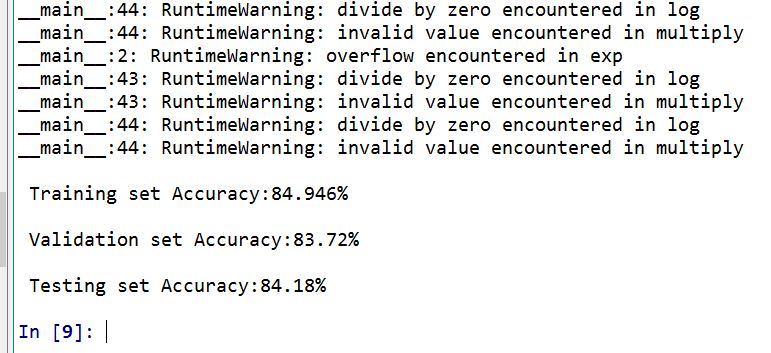
**Group Number**: 28

**Student Name**:

Steven Mundt

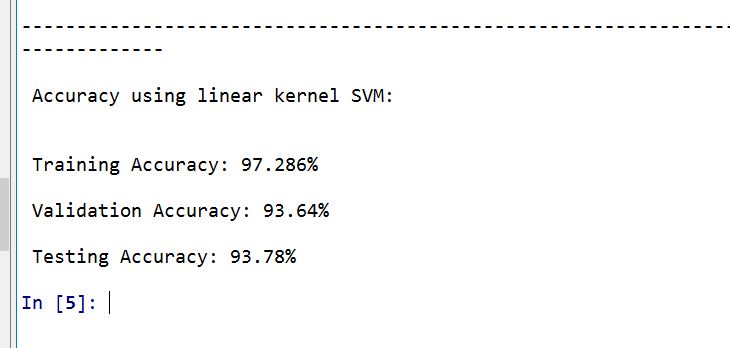
Qinxin Tian

1. **Logistic Regression**



For Logistic Regression, we got 84.946% accuracy on training set, 83.72% on validation set and 84.18% on test set. The training time for Logistic Regression was just a few minutes on my computer. The performance on training set is better than validation and test set which is normal, the model has never seen validation and test set, the accuracies are acceptable.

1. **SVM with Linear Kernel**



The SVM model was imported from scikit-learn library, the linear SVM was obtained by setting kernel=’linear’.

We got 97.286% accuracy on training dataset, 93.64% on validation dataset and 93.78% on test dataset. First of all, the performance of SVM model is much better than logistic regression. Second, the performance of training set is better than validation and test with around 4% advantage. This is not overfitting, the gap is normal because the model was trained on the training set.

1. **SVM with RBF Kernel, gamma=1.0**

------------------------------------------------------------------------------

Accuracy using RBF kernel SVM, gamma = 1.0

Training set Accuracy: 100.0%

Validation set Accuracy: 15.48%

Testing set Accuracy: 17.14%

Setting the gamma to 1.0 we got 100% training set Accuracy. However, we only got 15.48% accuracy for the Validation set and 17.14% accuracy for the testing set. This is a classic case of overfitting.

**SVM with RBF Kernel, all default parameters**

------------------------------------------------------------------------------

Accuracy using RBF kernel SVM, all parameters default

Training set Accuracy: 94.294%

Validation set Accuracy: 94.02%

Testing set Accuracy: 94.42%

When using all of the default parameters of SVM. Compared to the previous example with gamma set to 1. This is not overfitting and the results are better than the previous.

**SVM with RBF Kernel, varying C**

|  |  |  |  |
| --- | --- | --- | --- |
| C | Train Set Accuracy | Validation set Accuracy | Testing Set 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 |

When we altered the value of C for SVM. We can see that as we increased the value of C the accuracies for training set, validation set, and testing set all increased. For this example the optimal value of C is 100 as it resulted in 99.612% training set accuracy, 97.41% Validation set Accuracy, and 97.4% Testing set accuracy.

Comparing RBF to linear kernel when we optimized the value of C for RBF we achieved a higher accuracy for all sets.

Our optimized RBF resulted in 99.612% training set accuracy, 97.41% Validation set Accuracy, and 97.4% Testing set accuracy.

With linear kernel we achieved 97.286% accuracy on training dataset, 93.64% on validation dataset and 93.78% on test dataset. This shows that the optimized C value of RBF has higher accuracy then the linear kernel.