# CSE574 Introduction to Machine Learning
# Programming Assignment 3
# **Classification and Regression**


## Group 7

1. Karan Nisar (karankir)
2. Tejas Dhrangadharia (tejassha)


### 1.1 Implementation of Logistic Regression

Here we build a binary classifier for each of 10 categories using the one-vs-all strategy. The following is the overall accuracy obtained.

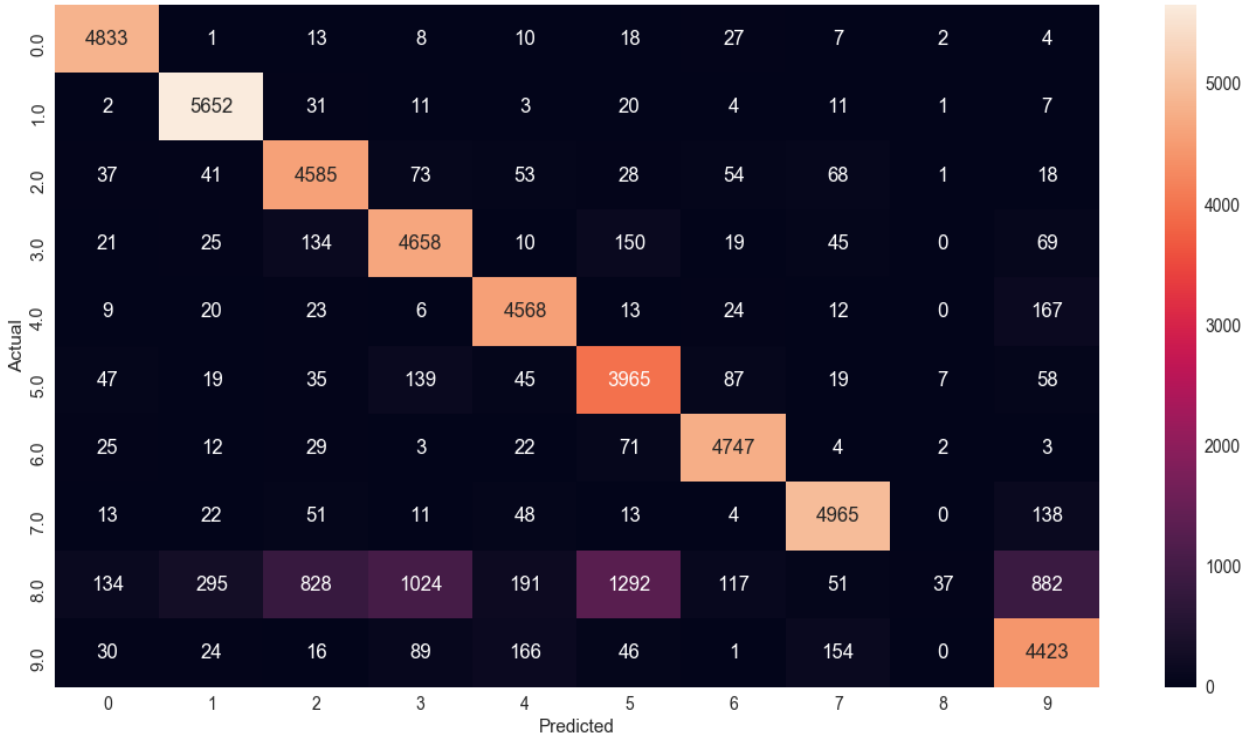
| | Accuracy |
|---|---|
| **Training Set** | 84.866% |
| **Validation Set** | 83.7% |
| **Testing Set** | 84.11% |

To get a deeper insight about error in each category, we obtain the following confusion matrix.

**For Train set**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4833 | 1 | 13 | 8 | 10 | 18 | 27 | 7 | 2 | 4 | 4923 | 98.17 |
| 1 | 2 | 5652 | 31 | 11 | 3 | 20 | 4 | 11 | 1 | 7 | 5742 | 98.43 |
| 2 | 37 | 41 | 4585 | 73 | 53 | 28 | 54 | 68 | 1 | 18 | 4958 | 92.47 |
| 3 | 21 | 25 | 134 | 4658 | 10 | 150 | 19 | 45 | 0 | 69 | 5131 | 90.78 |
| 4 | 9 | 20 | 23 | 6 | 4568 | 13 | 24 | 12 | 0 | 167 | 4842 | 94.34 |
| 5 | 47 | 19 | 35 | 139 | 45 | 3965 | 87 | 19 | 7 | 58 | 4421 | 89.68 |
| 6 | 25 | 12 | 29 | 3 | 22 | 71 | 4747 | 4 | 2 | 3 | 4918 | 96.52 |
| 7 | 13 | 22 | 51 | 11 | 48 | 13 | 4 | 4965 | 0 | 138 | 5265 | 94.30 |
| 8 | 134 | 295 | 828 | 1024 | 191 | 1292 | 117 | 51 | 37 | 882 | 4851 | 0.76 |
| 9 | 30 | 24 | 16 | 89 | 166 | 46 | 1 | 154 | 0 | 4423 | 4949 | 89.37 |
| All | 5151 | 6111 | 5745 | 6022 | 5116 | 5616 | 5084 | 5336 | 50 | 5769 | 50000 | |
| % | 93.82 | 92.48 | 79.80 | 77.34 | 89.28 | 70.60 | 93.37 | 93.04 | 74 | 76.66 | | 84.866 |

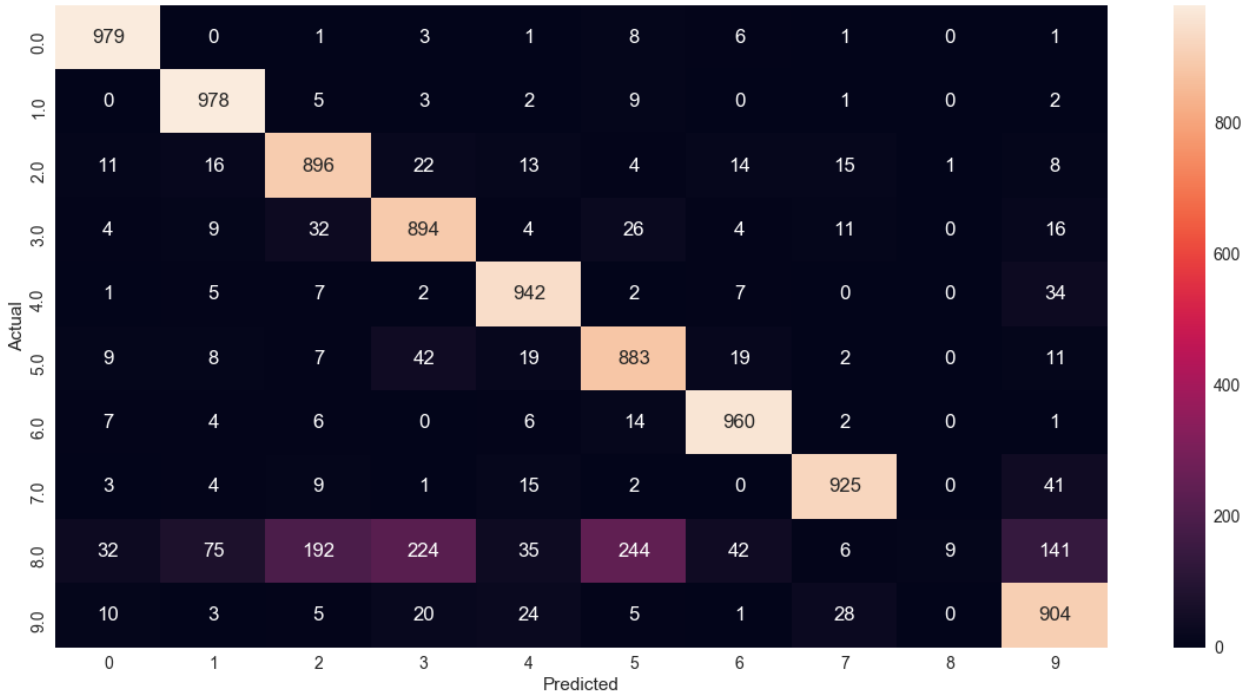
The heatmap of the above confusion matrix is




**For Validation Set**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 979 | 0 | 1 | 3 | 1 | 8 | 6 | 1 | 0 | 1 | 1000 | 97.9 |
| 1 | 0 | 978 | 5 | 3 | 2 | 9 | 0 | 1 | 0 | 2 | 1000 | 97.8 |
| 2 | 11 | 16 | 896 | 22 | 13 | 4 | 14 | 15 | 1 | 8 | 1000 | 89.6 |
| 3 | 4 | 9 | 32 | 894 | 4 | 26 | 4 | 11 | 0 | 16 | 1000 | 89.4 |
| 4 | 1 | 5 | 7 | 2 | 942 | 2 | 7 | 0 | 0 | 34 | 1000 | 94.2 |
| 5 | 9 | 8 | 7 | 42 | 19 | 883 | 19 | 2 | 0 | 11 | 1000 | 88.3 |
| 6 | 7 | 4 | 6 | 0 | 6 | 14 | 960 | 2 | 0 | 1 | 1000 | 96 |
| 7 | 3 | 4 | 9 | 1 | 15 | 2 | 0 | 925 | 0 | 41 | 1000 | 92.5 |
| 8 | 32 | 75 | 192 | 224 | 35 | 244 | 42 | 6 | 9 | 141 | 1000 | 0.9 |
| 9 | 10 | 3 | 5 | 20 | 24 | 5 | 1 | 28 | 0 | 904 | 1000 | 90.4 |
| All | 1056 | 1102 | 1160 | 1211 | 1061 | 1197 | 1053 | 991 | 10 | 1159 | 10000 | |
| % | 92.70 | 88.74 | 77.24 | 73.82 | 88.78 | 73.76 | 91.16 | 93.34 | 90 | 77.99 | | 83.7 |

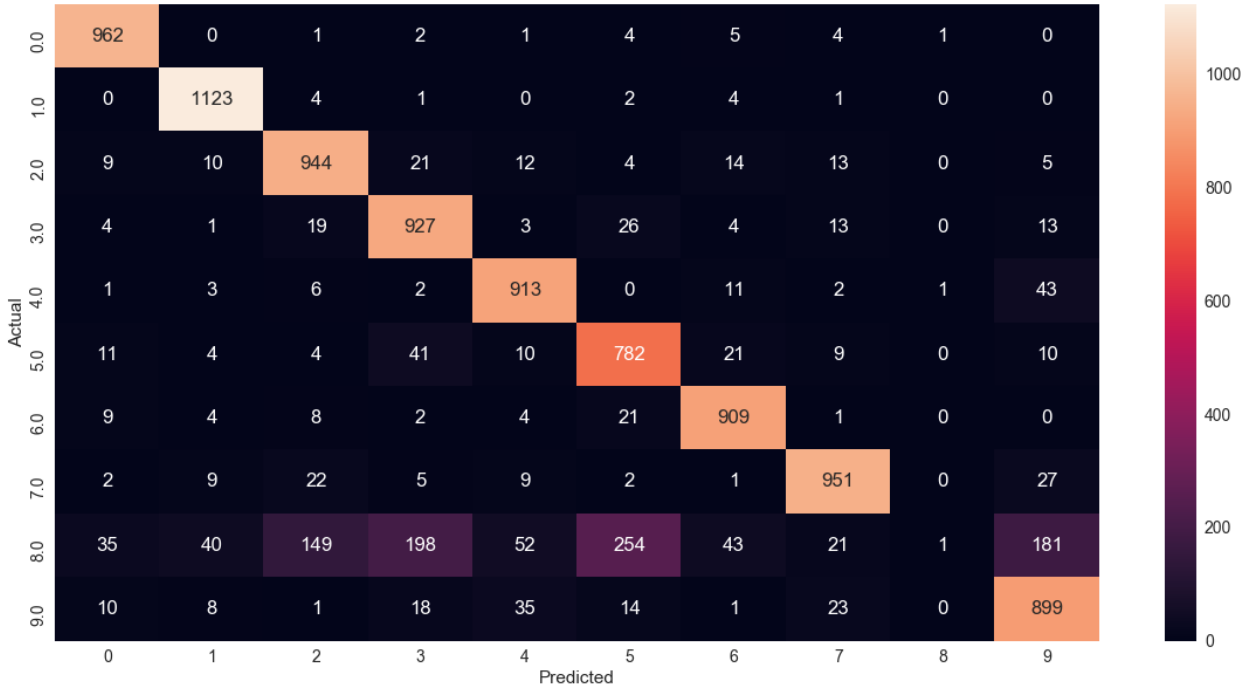
The heatmap is




**For Test Set**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 962 | 0 | 1 | 2 | 1 | 4 | 5 | 4 | 1 | 0 | 980 | 98.16 |
| 1 | 0 | 1123 | 4 | 1 | 0 | 2 | 4 | 1 | 0 | 0 | 1135 | 98.94 |
| 2 | 9 | 10 | 944 | 21 | 12 | 4 | 14 | 13 | 0 | 5 | 1032 | 91.47 |
| 3 | 4 | 1 | 19 | 927 | 3 | 26 | 4 | 13 | 0 | 13 | 1010 | 91.78 |
| 4 | 1 | 3 | 6 | 2 | 913 | 0 | 11 | 2 | 1 | 43 | 982 | 92.97 |
| 5 | 11 | 4 | 4 | 41 | 10 | 782 | 21 | 9 | 0 | 10 | 892 | 87.66 |
| 6 | 9 | 4 | 8 | 2 | 4 | 21 | 909 | 1 | 0 | 0 | 958 | 94.88 |
| 7 | 2 | 9 | 22 | 5 | 9 | 2 | 1 | 951 | 0 | 27 | 1028 | 92.50 |
| 8 | 35 | 40 | 149 | 198 | 52 | 254 | 43 | 21 | 1 | 181 | 974 | 0.10 |
| 9 | 10 | 8 | 1 | 18 | 35 | 14 | 1 | 23 | 0 | 899 | 1009 | 89.09 |
| All | 1043 | 1202 | 1158 | 1217 | 1039 | 1109 | 1013 | 1038 | 3 | 1178 | 10000 | |
| % | 92.23 | 93.42 | 81.51 | 76.17 | 87.87 | 70.51 | 89.73 | 91.61 | 33.33 | 76.31 | | 84.11 |

The heatmap is



## 1.2 Multi-class Logistic Regression

Here we use only one classifier instead of 10 and compare the difference between 2 approaches. We get the following accuracies:

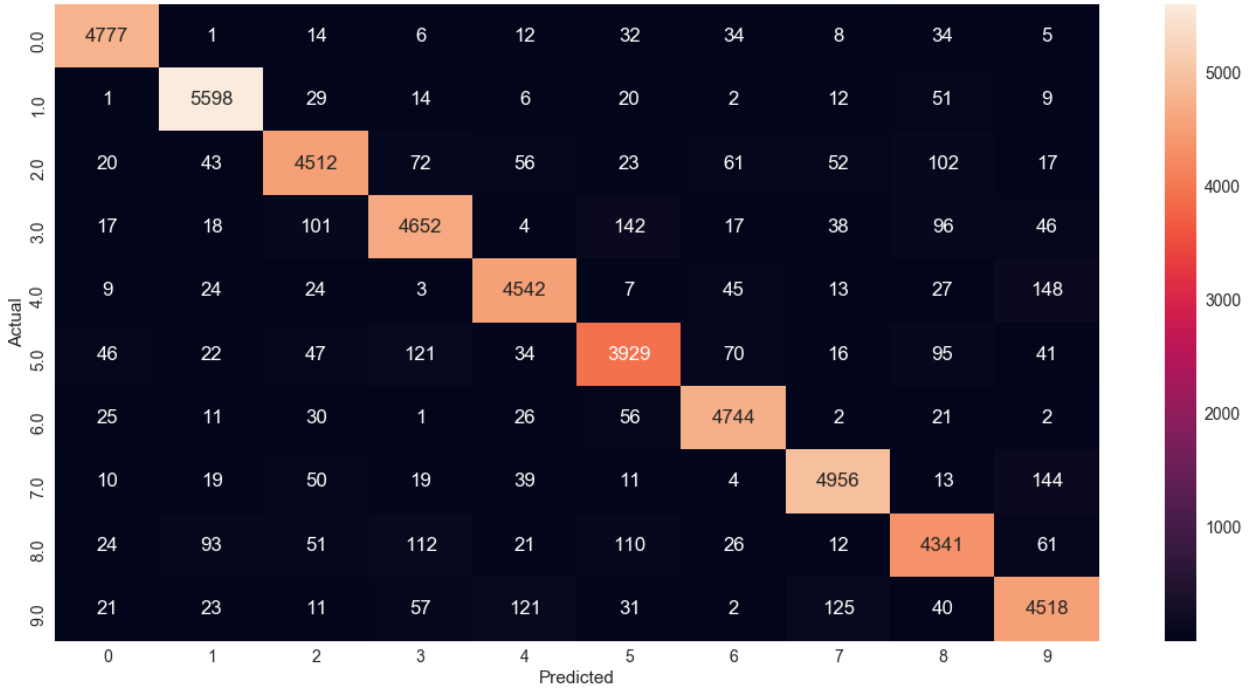
| | Accuracy |
|---|---|
| **Training Set** | 93.138% |
| **Validation Set** | 92.54% |
| **Testing Set** | 92.53% |

The confusion matrix is:

**For Train set**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4777 | 1 | 14 | 6 | 12 | 32 | 34 | 8 | 34 | 5 | 4923 | 97.03 |
| 1 | 1 | 5598 | 29 | 14 | 6 | 20 | 2 | 12 | 51 | 9 | 5742 | 97.49 |
| 2 | 20 | 43 | 4512 | 72 | 56 | 23 | 61 | 52 | 102 | 17 | 4958 | 91.00 |
| 3 | 17 | 18 | 101 | 4652 | 4 | 142 | 17 | 38 | 96 | 46 | 5131 | 90.66 |
| 4 | 9 | 24 | 24 | 3 | 4542 | 7 | 45 | 13 | 27 | 148 | 4842 | 93.80 |
| 5 | 46 | 22 | 47 | 121 | 34 | 3929 | 70 | 16 | 95 | 41 | 4421 | 88.87 |
| 6 | 25 | 11 | 30 | 1 | 26 | 56 | 4744 | 2 | 21 | 2 | 4918 | 96.46 |
| 7 | 10 | 19 | 50 | 19 | 39 | 11 | 4 | 4956 | 13 | 144 | 5265 | 94.13 |
| 8 | 24 | 93 | 51 | 112 | 21 | 110 | 26 | 12 | 4341 | 61 | 4851 | 89.48 |
| 9 | 21 | 23 | 11 | 57 | 121 | 31 | 2 | 125 | 40 | 4518 | 4949 | 91.29 |
| All | 4950 | 5852 | 4869 | 5057 | 4861 | 4361 | 5005 | 5234 | 4820 | 4991 | 50000 | |
| % | 96.50 | 95.65 | 92.66 | 91.99 | 93.43 | 90.09 | 94.78 | 94.68 | 90.06 | 90.52 | | 93.14 |

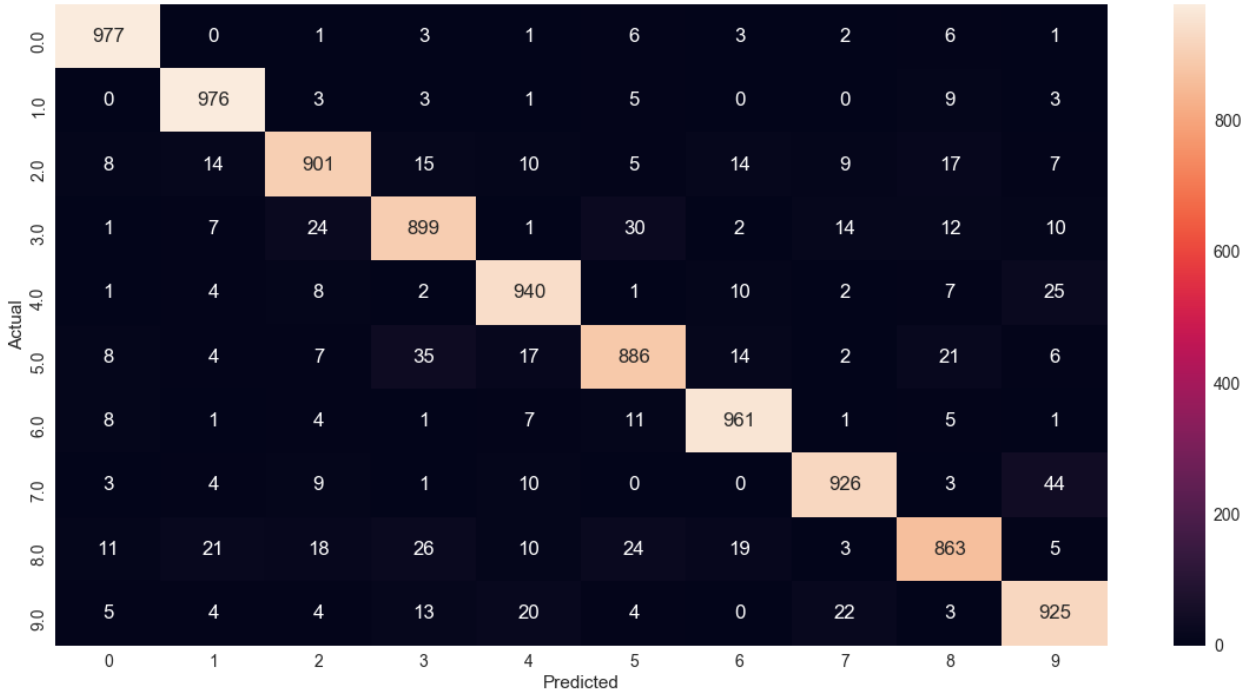
Following is the heatmap




**For Validation Dataset**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 977 | 0 | 1 | 3 | 1 | 6 | 3 | 2 | 6 | 1 | 1000 | 97.7 |
| 1 | 0 | 976 | 3 | 3 | 1 | 5 | 0 | 0 | 9 | 3 | 1000 | 97.6 |
| 2 | 8 | 14 | 901 | 15 | 10 | 5 | 14 | 9 | 17 | 7 | 1000 | 90.1 |
| 3 | 1 | 7 | 24 | 899 | 1 | 30 | 2 | 14 | 12 | 10 | 1000 | 89.9 |
| 4 | 1 | 4 | 8 | 2 | 940 | 1 | 10 | 2 | 7 | 25 | 1000 | 94 |
| 5 | 8 | 4 | 7 | 35 | 17 | 886 | 14 | 2 | 21 | 6 | 1000 | 88.6 |
| 6 | 8 | 1 | 4 | 1 | 7 | 11 | 961 | 1 | 5 | 1 | 1000 | 96.1 |
| 7 | 3 | 4 | 9 | 1 | 10 | 0 | 0 | 926 | 3 | 44 | 1000 | 92.6 |
| 8 | 11 | 21 | 18 | 26 | 10 | 24 | 19 | 3 | 863 | 5 | 1000 | 86.3 |
| 9 | 5 | 4 | 4 | 13 | 20 | 4 | 0 | 22 | 3 | 925 | 1000 | 92.5 |
| All | 1022 | 1035 | 979 | 998 | 1017 | 972 | 1023 | 981 | 946 | 1027 | 10000 | |
| % | 95.59 | 94.29 | 92.03 | 90.08 | 92.42 | 91.15 | 93.93 | 94.39 | 91.22 | 90.06 | | 92.54 |

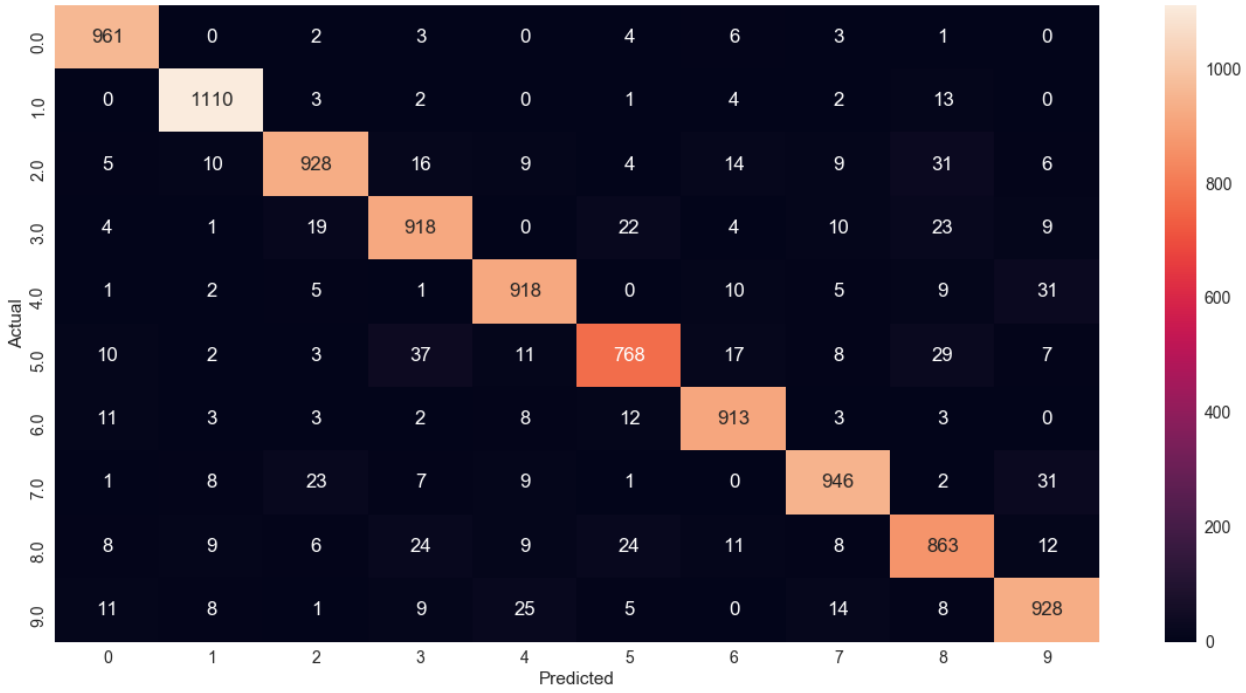
The heatmap is



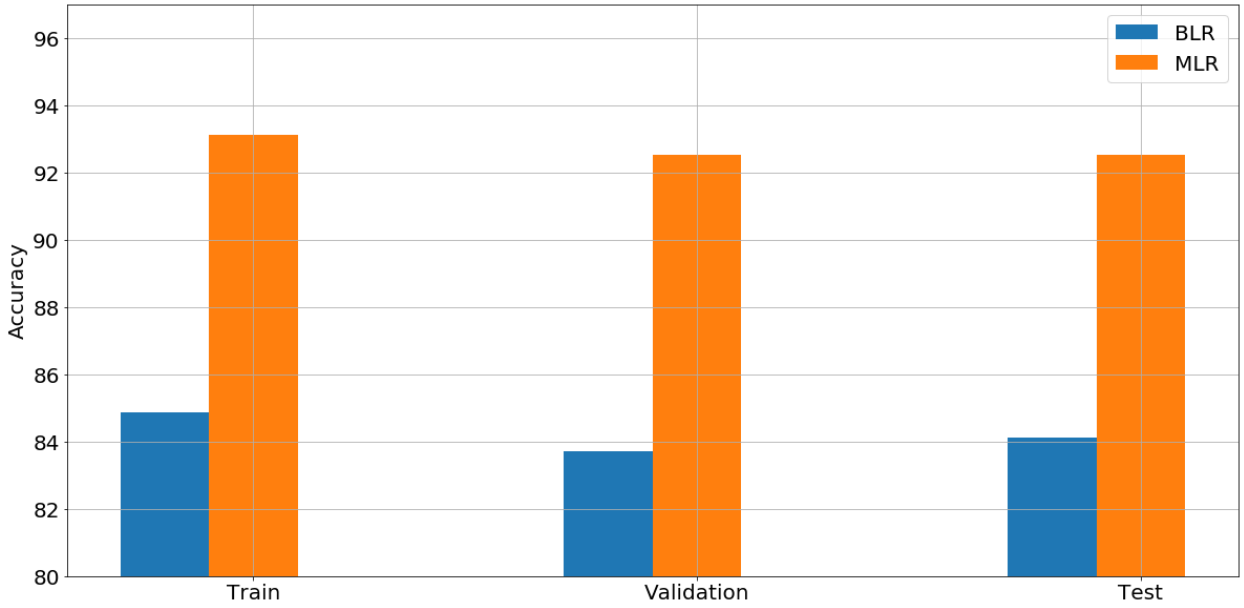

**For Test Set**

| Actual / Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | All | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 961 | 0 | 2 | 3 | 0 | 4 | 6 | 3 | 1 | 0 | 980 | 98.06 |
| 1 | 0 | 1110 | 3 | 2 | 0 | 1 | 4 | 2 | 13 | 0 | 1135 | 97.79 |
| 2 | 5 | 10 | 928 | 16 | 9 | 4 | 14 | 9 | 31 | 6 | 1032 | 89.92 |
| 3 | 4 | 1 | 19 | 918 | 0 | 22 | 4 | 10 | 23 | 9 | 1010 | 90.89 |
| 4 | 1 | 2 | 5 | 1 | 918 | 0 | 10 | 5 | 9 | 31 | 982 | 93.48 |
| 5 | 10 | 2 | 3 | 37 | 11 | 768 | 17 | 8 | 29 | 7 | 892 | 86.09 |
| 6 | 11 | 3 | 3 | 2 | 8 | 12 | 913 | 3 | 3 | 0 | 958 | 95.30 |
| 7 | 1 | 8 | 23 | 7 | 9 | 1 | 0 | 946 | 2 | 31 | 1028 | 92.02 |
| 8 | 8 | 9 | 6 | 24 | 9 | 24 | 11 | 8 | 863 | 12 | 974 | 88.60 |
| 9 | 11 | 8 | 1 | 9 | 25 | 5 | 0 | 14 | 8 | 928 | 1009 | 91.97 |
| All | 1012 | 1153 | 993 | 1019 | 989 | 841 | 979 | 1008 | 982 | 1024 | 10000 | |
| % | 94.96 | 96.27 | 93.45 | 90.08 | 92.82 | 91.31 | 93.25 | 93.84 | 87.88 | 90.62 | | 92.53 |

The heatmap is




On comparing the above two approaches we plot the following comparison bar plot.




The bar graph clearly shows that MLR performs better than BLR. The multi-class strategy compares the weights of all classes and updates them while in one-vs-all strategy the weights are calculated class wise without considering the weights of other classes.

## 1.3 Support Vector Machines (SVM)

Following is the output:

1. For Linear Kernel (all parameters kept to default)
   We see that the linear method gives good accuracies on the entire dataset. Therefore, the data is linearly separable.

| | Accuracy |
|---|---|
| **Training Set** | 97.286% |
| **Validation Set** | 93.64% |
| **Testing Set** | 93.78% |

2. For radial basis function with value of gamma setting to 1 (all other parameters are kept default)
   Gamma is the kernel co-efficient. A large gamma means a Gaussian function with a small variance. So SVM tries to find complex boundaries to distinguish between different categories. In doing this there is a high chance of over-fitting. Due to the complexity, it takes a very long time to run on the entire dataset, so we computed this on a subset of 10000 datasets. We can clearly see that it overfits giving a 100% accuracy on training and extremely low on validation and test dataset.

| | Accuracy |
|---|---|
| **Training Set** | 100.0% |
| **Validation Set** | 15.03% |
| **Testing Set** | 16.16% |

3. For radial basis function with value of gamma setting to default (all other parameters are kept default)

   When gamma is default, then only 1/n_features are used. This gives a good generalization leading to much lower runtime on the entire dataset and good accuracies compared to previous case.

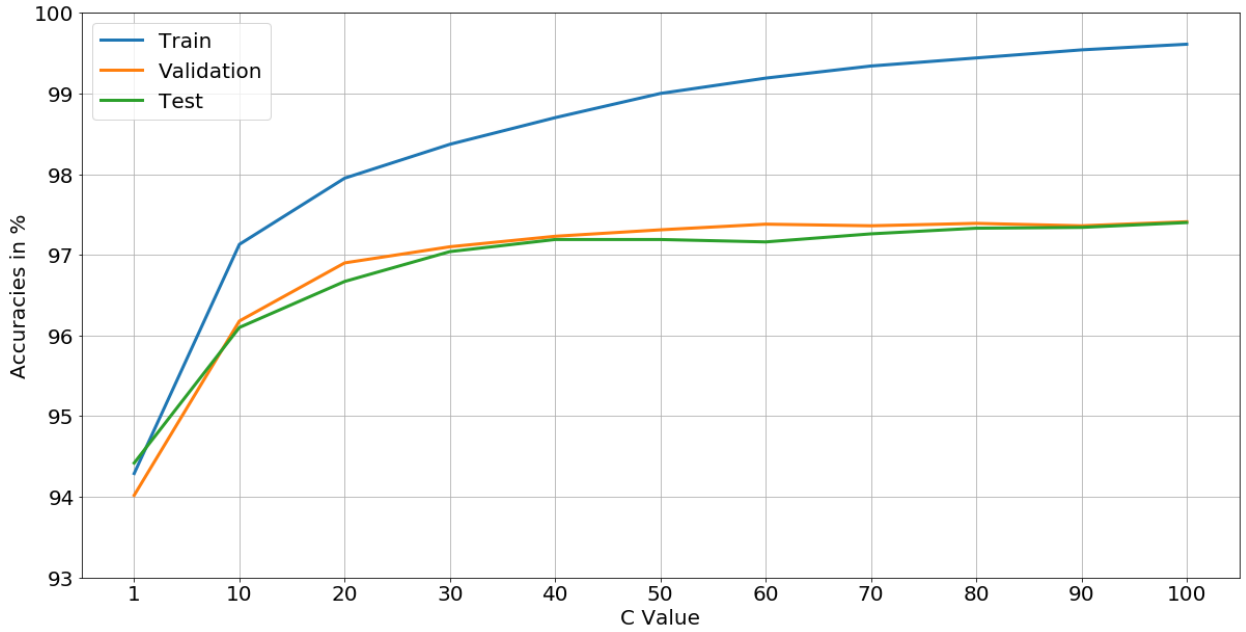
| | Accuracy |
|---|---|
| **Training Set** | 94.294% |
| **Validation Set** | 94.02% |
| **Testing Set** | 94.42% |

4. For radial basis function with value of gamma setting to default and varying value of C (1, 20, 30, … , 100)

   Here C is the penalty parameter of the error term. Following is the accuracies for different C's

| C Value | Train Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| **1** | 94.29 % | 94.02 % | 94.42 % |
| **10** | 97.13 % | 96.18 % | 96.1 % |
| **20** | 97.95 % | 96.9 % | 96.67 % |
| **30** | 98.37 % | 97.1 % | 97.04 % |
| **40** | 98.7 % | 97.23 % | 97.19 % |
| **50** | 99 % | 97.31 % | 97.19 % |
| **60** | 99.19 % | 97.38 % | 97.16 % |
| **70** | 9.34 % | 97.36 % | 97.26 % |
| **80** | 99.44 % | 97.39 % | 97.33 % |
| **90** | 99.54 % | 97.36 % | 97.34 % |
| **100** | 99.61 % | 97.41 % | 97.4 % |

Following is the graph comparing the accuracies for different C Values




The accuracies obtained from SVM for C = 1 is comparable to MLR (Multi-class Logistic Regression).