REPORT
Programming Assignment 3
Classification and Regression

Pretty Mary Philip - 5024 7311
Tejaswini Reddy Boda - 5024 3913

1.1 Logistic Regression

Training set accuracy: 86.176%
Validation set accuracy: 85.25%
Test set accuracy: 85.41%

Training set error: 0.0970091309001

Confusion Matrix for Training data:

4826	1	14	7	9	21	27	7	7	4
2	5650	32	12	3	17	4	11	4	7
34	42	4582	70	55	24	56	64	14	17
22	25	130	4641	9	162	22	43	11	66
8	20	26	6	4555	13	28	12	4	170
43	18	32	135	50	3956	85	16	30	56
23	13	29	2	19	69	4743	3	14	3
13	22	48	11	44	11	3	4970	3	140
118	272	642	895	176	1079	105	48	742	774
28	27	16	87	163	41	1	154	9	4423

Confusion Matrix for Test data:

960	0	1	2	1	5	6	3	1	1
0	1115	3	1	0	1	4	1	10	0

8	12	917	19	10	4	12	11	34	5
4	0	19	921	2	20	4	13	17	10
1	2	5	3	919	0	10	2	4	36
10	2	1	41	12	761	17	9	29	10
8	4	7	2	4	20	907	1	5	0
2	11	23	5	8	2	1	946	2	28
13	14	8	21	14	29	8	12	844	11
8	8	1	11	34	13	1	23	11	899

After training the logistic regressor with given data and labels, we obtain training, validation and test accuracies, which are recorded here. The validation and test accuracies are almost similar, and training accuracy is slightly higher than both.
We have also used a confusion matrix to report error with respect to each category. The test error is slightly higher than training error, likely due to the model overfitting to the training data and not generalizing well. This is fairly common and not unexpected. Another reason for higher test error is smaller size of test set.

1.2 Multi-class Logistic Regression

Training set accuracy: 93.448%
Validation set accuracy: 92.47999999999999%%
Test set accuracy: 92.55%

Training set error: 0.23787739851431874

Confusion Matrix for Training data:

4786	1	12	7	11	33	30	7	32	4
1	5592	26	17	6	19	2	13	58	8
23	45	4503	72	58	24	59	53	108	13
14	18	95	4654	4	148	15	39	105	39
8	20	21	7	4576	6	42	13	24	125
39	13	36	117	34	3963	68	18	102	31

23	11	29	1	24	52	4758	2	16	2
8	16	49	18	34	9	4	4989	14	124
22	75	51	103	16	113	23	16	4387	45
17	18	9	55	126	30	2	134	42	4516

Confusion Matrix for Test Data:

960	0	0	3	0	6	6	4	1	0
0	1110	3	2	0	2	4	2	12	0
6	8	924	16	10	3	14	8	39	4
4	1	20	914	0	25	3	10	26	7
1	1	6	2	921	0	9	4	9	29
10	2	2	37	10	773	15	6	30	7
9	3	4	2	7	15	914	3	1	0
1	9	19	6	6	2	0	952	2	31
9	8	6	26	9	23	10	8	868	7
11	8	0	10	28	5	0	20	8	919

After training the logistic regressor with given data and labels, we obtain training, validation and test accuracies, which are recorded here. The validation and test accuracies are almost similar, and training accuracy is slightly higher than both.
We have also used a confusion matrix to report error with respect to each category. The test error is slightly higher than training error, likely due to the model overfitting to the training data and not generalizing well. This is fairly common and not unexpected. Another reason for higher test error is smaller size of test set.

Performance comparison of one vs all and multi-class strategies
According to our results, multiclass regression gives higher accuracy when compared to one vs all regression. Since one vs all uses n binary classifiers for n classes, it is not very feasible when we have a large number of classes. Class imbalances also affect its performance more than in multi-class. Multi-class however, uses only one classifier to classify n classes and performs better.

1.3 Support Vector Machines

- SVM linear kernel
  Training set accuracy: 97.286%
  Validation set accuracy: 93.64%
  Test set accuracy: 93.78%

- SVM radial bias function with gamma = 1
  Training set accuracy: 100%
  Validation set accuracy: 15.48%
  Test set accuracy: 17.14%

- SVM radial bias function with gamma = default
  Training set accuracy: 94.294%
  Validation set accuracy: 94.02%
  Test set accuracy: 94.42%

- SVM radial bias function with gamma = default and varying C

C	Training Set Accuracy (%)	Validation Set Accuracy (%)	Test 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

The above graph plots the accuracies of training, validation and test data for varying values of C using SVM with radial bias function (RBF) kernel.
Since C determines the impact of error on training data, it controls the complexity of the learned hyperplane too. Initially when C is low, weight of error term is also low and larger error values are accepted during training, giving a larger margin hyperplane. However, this creates a bigger set of misclassified samples.
On the other hand, when C increases, only lower error values are accepted, thus increasing accuracy and decreasing misclassification. However, a smaller margin hyperplane is created.
From the graph, we see that training set accuracy increases rapidly throughout, while accuracies for validation and test increase slowly. In fact, it decreases for test set from 40 to 60. This points to possible overfitting when increasing C values and complexity of hyperplane.

The gamma parameter tells the influence of a training vector on the others. Smaller gamma means high influence and low bias, and vice versa. As observed in our results, linear kernel has lower accuracy compared to RBF with gamma = 0, and higher accuracy compared to RBF with gamma = 1. As size of training set grows and number of features increases, RBF becomes more complex and does not perform as well as linear kernel. RBF, which is a non-linear kernel, also has higher risk of overfitting. Hence, it is preferable to use it only when dataset is not linearly separable. If the number of features is large, we may not need to map data to a higher dimensional space. That is, the nonlinear mapping does not improve the performance in this case and using the linear kernel is good enough.