Sanjana Agarwal
sa14593

Machine Learning Assignment 4

1) The code for Q1 can be found in script_classify.py
I performed a 70-30 split on the dataset.
I tried the RBF Network for different values of sigma and my evaluation for those values are:

	ACCURACY	ERROR
SIGMA = 1	54.18	45.82
SIGMA = 6	54.48	45.52
SIGMA = 11	55.5	44.5

Average error for Radial Basis Transformation: 45.78 +- 0.0378144005491

Performance of standard logistic regression without the use of any regularizers:

	ACCURACY	ERROR
numruns = 1	74.845	25.155
numruns = 2	74.74	25.26
numruns = 3	74.045	25.955

Average error for Logistic Regression: 25.4566666667 +- 0.401689325318
Thus, we observe,

- The standard logistic regression performs better than Radial Basis Transformation Function.
- The accuracies for RBF can be slightly improved if centers are selected by K-Means or some other clustering technique. I chose to select random centers from the X given points because of which the answers generally vary in every run.
- According to my observations, higher values of sigma correspond to lower error rates.

2) For Q2, the algorithms that I chose are Naive Bayes, Logistic Regression and Neural Networks.

I have performed cross validation on my own and used scikit-learn to present the F1 score, Precision, Recall and Support.
I will tabulate the results individually for each algorithm.

Naive Bayes:

CV = 10 (number of folds for cross validation)

	PRECISION	RECALL	F1	SUPPORT
0.0	1.0	0.92	0.96	99
1.0	0.79	1.0	0.89	31
AVG/TOTAL	0.95	0.94	0.94	130
ACCURACY: 93.85				
AVERAGE ERROR: 0.615384615385 +- 1.07881679653				
Best Parameters: {usecolumnones:False}				

Logistic Regression:

CV = 10 (number of folds for cross validation)

	PRECISION	RECALL	F1	SUPPORT
0.0	1.0	0.85	0.92	101
1.0	0.66	1.0	0.79	29
AVG/TOTAL	0.92	0.88	0.89	130
ACCURACY: 88.461				
AVERAGE ERROR: 3.53846153846 +- 6.20319658004				
Best Parameters: {regwt:0.01}				

Neural Networks:

CV = 10 (number of folds for cross validation)

	PRECISION	RECALL	F1	SUPPORT
0.0	0.83	1.00	0.91	108
1.0	0.0	0.0	0.00	22
AVG/TOTAL	0.69	0.83	0.75	130
ACCURACY: 83.077				
AVERAGE ERROR: 4.30769230769 +- 7.5517175757				
Best Parameters: {'nh': 8, 'stepsize': 0.01, 'epochs': 10, 'ni': 9}				

I achieved the best results with a 80-20 split of the dataset.
I worked on the occupancy datset: http://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+
It has 20K data samples.
The code can be found in script_classify2.py

- Thus, I achieved the best results with Naive Bayes.
- Performance went somewhat low with Neural Network
- Logistic Regression and Naive Bayes had comparable performances.
- All these algorithms outperformed the Random Classifier.
- The variance increases as the accuracy increases.
- To verify the performance of my cross validation code, I compared its performance to scikit-learn’s cross_val_score method and obtained similar results.