Alek Michael

Final Project

Description: Classifiers and feature selection are implemented using python libraries numpy, mlxtend, sklearn, and matplotlib.

Average MSE and Time of Classifiers (3 runs)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n\_samples | noise mutation chance | MLP | k-NN | SVM | Random Forest |
| 100 | .1 | 0.856  (8.465 secs) | 0.0533  (0.279 secs) | 0.025  (0.023 secs) | 0.055  (2.375 secs) |
| 500 | .1 | 0.439  (24.295 secs) | 0.047  (0.453 secs) | 0.039  (0.371 secs) | 0.0285  (7.414 secs) |
| 1000 | .1 | 0.2975  (30.709 secs) | 0.031  (1.959 secs) | 0.022  (0.742 secs) | 0.043  (13.792 secs) |
| 100 | .25 | 9.941  (11.56 secs) | 2.775  (0.299 secs) | 1.987  (0.045 secs) | 1.95  (3.133 secs) |
| 500 | .25 | 3.709  (24.407 secs) | 1.953  (0.446 secs) | 1.783  (0.740 secs) | 1.976  (8.985 secs) |
| 1000 | .25 | 0.341  (33.151 secs) | 2.195  (2.003 secs) | 1.697  (2.122 secs) | 1.872  (19.488 secs) |
| 100 | .50 | 17.215  (1.049 secs) | 15.222  (0.026 secs) | 18.3225  (0.056 secs) | 16.662  (3.749 secs) |
| 500 | .50 | 15.684  (1.816 secs) | 17.4065  (0.713 secs) | 17.049  (1.518 secs) | 16.83  (12.566 secs) |
| 1000 | .50 | 17.664  (3.356 secs) | 17.086  (2.095 secs) | 16.809  (6.863 secs) | 15.106  (22.428 secs) |

The best on average classifier for this problem is the supper vector machine. Although the random forest classifier was very accurate on low noise mutation, SVM performed more accurately and faster.

By far the worst classifier for this problem is the MLP classifier. Perhaps with more precision on the settings would make a slight improvement but this is the best I could get it.

Settings:

MLP:

* Hidden Layers: [6, 3, 6]
* Activation Function: Rectified linear unit
* Solver: Stochastic gradient-based optimizer
* Maximum iterations: 1000000
* Initial learning rate: 0.001
* Random State: 123

k-NN:

* Neighbors: 10
* Weights: Equal
* Leaf size: 30

SVM:

* C = 1
* Polynomial degree: 3
* Uses shrinking heuristic
* Decision function shape: one vs rest

Random Forest:

* Number of forests: 2000
* Max Features: sqrt(n\_features)
* Uses bootstrap samples

Using SVM for feature selection:

For accuracy >97.5%, a minimum of 20 input features should be selected.

Selected features: ('0', '1', '2', '3', '4', '5', '11', '15', '22', '23', '24', '25', '26', '33', '38', '42', '43', '44', '46', '47')

For accuracy >95%, a minimum of 14 input features should be selected.

Selected features: ('2', '5', '7', '15', '19', '22', '23', '24', '25', '33', '34', '43', '45', '46')

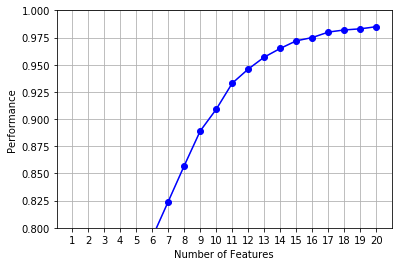
For accuracy >90%, a minimum of 10 input features should be selected.

Selected features: ('0', '1', '5', '12', '23', '25', '26', '40', '41', '43')

For at least 80% accuracy, a minimum of 6 input features should be selected.

Selected features: ('0', '4', '22', '24', '28', '43')

Number of Features vs Performance



Conclusions: Besides MLP, the classifiers performed adequately for this job. The best performing classifier was the SVM model. It performed best to noise and was the fastest performer of all classifiers. It’s notable how much of the 49-sized inputs can be reduced to as low as 6 and still get a high accuracy. Feature selection here has worked quiet effectively!