Homework 2

Execution of Part 1:

Set training and testing paths in block one and execute.

Execution for Part 2:

Execute all blocks and outputs will be on the output portion. The validation tests have been commented and you can uncomment those for further validation. GridSearchCV was not used as it was specified to try a variety of parameters and not just penalty and kernel function.

1. SVM Classifier:

Pre-Tuning:

|  |  |  |
| --- | --- | --- |
| No Parameters | Error = 0.02080000000000004 | Accuracy = 0.9792 |

Validation:

|  |  |  |
| --- | --- | --- |
| Parameters | Error | Accuracy |
| kernel = 'rbf' | 0.046666666666666634 | 0.9533333333333334 |
| kernel = 'linear', C = 0.01 | 0.07666666666666666 | 0.9233333333333333 |
| kernel = 'rbf', C = 10, gamma = 'auto' | 0.06266666666666665 | 0.9373333333333334 |
| kernel = 'poly', C =100 | 0.058666666666666645 | 0.9413333333333334 |
| C = 0.001 | 0.882 | 0.118 |
| kernel = 'poly', degree = 2, coef0 = 10 | 0.07466666666666666 | 0.9253333333333333 |
| gamma = 'auto' | 0.08399999999999996 | 0.916 |
| kernel = 'rbf', C = 10 | 0.04066666666666663 | 0.9593333333333334 |
| kernel = 'sigmoid', C = 10, gamma = 'auto' | 0.07666666666666666 | 0.9233333333333333 |
| kernel = 'poly', degree = 10, C = 0.1 | 0.5286666666666666 | 0.4713333333333333 |

Post-Tuning:

|  |  |  |
| --- | --- | --- |
| kernel = 'rbf', C = 10 | 0.01629999999999998 | 0.9837 |

1. MLP Classifier

Pre-Tuning:

|  |  |  |
| --- | --- | --- |
| No Parameters | Error = 0.021599999999999953 | Accuracy = 0.9784 |

Validation:

|  |  |  |
| --- | --- | --- |
| Parameters | Error | Accuracy |
| solver = adam | 0.07533333333333336 | 0.9246666666666666 |
| activation = 'tanh', learning\_rate = 'adaptive' | 0.07399999999999995 | 0.926 |
| alpha = 0.0001, activation = 'logistic', solver = 'sgd' | 0.14933333333333332 | 0.8506666666666667 |
| hidden\_layer\_sizes = (50,100,50), activation = 'relu', alpha = 1 | 0.06999999999999995 | 0.93 |
| alpha = 0.1, learning\_rate = 'adaptive' | 0.06466666666666665 | 0.9353333333333333 |
| hidden\_layer\_sizes = (50,), activation = 'identity', solver = 'lbfgs' | 0.1273333333333333 | 0.8726666666666667 |
| activation = 'logistic', alpha = 0.0001 | 0.07199999999999995 | 0.928 |
| alpha = 10 | 0.11933333333333329 | 0.8806666666666667 |
| hidden\_layer\_sizes = (200,250,100), alpha = 0.005 | 0.06399999999999995 | 0.936 |
| hidden\_layer\_sizes = (200,), alpha = 0.1, activation = 'tanh', solver = 'adam', learning\_rate = 'adaptive' | 0.06799999999999995 | 0.932 |

Post-Tuning:

|  |  |  |
| --- | --- | --- |
| hidden\_layer\_sizes = (200,250,100), alpha = 0.005 | 0.018299999999999983 | 0.9817 |

1. KNeighbours Classifier:

Pre-Tuning

|  |  |  |
| --- | --- | --- |
| No Parameters | Error = 0.0312 | Accuracy = 0.9688 |

Validation:

|  |  |  |
| --- | --- | --- |
| Parameters | Error | Accuracy |
| weights = 'distance' | 0.06999999999999995 | 0.93 |
| p = 5 | 0.06266666666666665 | 0.9373333333333334 |
| n\_neighbors = 3, p = 1, leaf\_size = 40 | 0.08599999999999997 | 0.914 |
| n\_neighbors = 9, leaf\_size = 100, p = 1 | 0.08799999999999997 | 0.912 |
| weights = 'distance', p = 5 | 0.06133333333333335 | 0.9386666666666666 |
| n\_neighbors = 3, weights = 'distance', p = 3, leaf\_size = 100 | 0.06466666666666665 | 0.9353333333333333 |
| n\_neighbors = 7, p = 1, leaf\_size = 10 | 0.08666666666666667 | 0.9133333333333333 |
| n\_neighbors = 3, p = 2, leaf\_size = 50 | 0.07399999999999995 | 0.926 |
| n\_neighbors = 13, p = 1, leaf\_size = 50, weights = 'distance' | 0.09533333333333338 | 0.9046666666666666 |
| n\_neighbors = 41, leaf\_size = 100 | 0.11933333333333329 | 0.8806666666666667 |

Post-Tuning:

|  |  |  |
| --- | --- | --- |
| n\_neighbors = 3, weights = 'distance' | 0.028299999999999992 | 0.9717 |

Best Error Rates:

|  |  |  |
| --- | --- | --- |
| Classifier | Parameters | Error Rate |
| SVM | kernel = 'rbf', C = 10 | 0.04066666666666663 |
| MLP | hidden\_layer\_sizes = (200,250,100), alpha = 0.005 | 0.06399999999999995 |
| KNC | n\_neighbors = 3, weights = 'distance' | 0. 028299999999999992 |

* There are other combinations in KNC which give a better validation score but as they were a poly kernel of high degree, I was unable to train them on my machine. So, the next best choice was selected.