Name: Lucas Keizur

ID: 40987746

case 1 –

noise: 0.1 n\_samples: 100 set size: 1000

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 200

Value of *k* = 3

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy% | Error% | Times(s) |
| MLP | 95.75 | 4.25 | 7.17 |
| *k*-NN | 98 | 2 | .06 |
| SVM | 96 | 4 | .28 |
| Random Forest | 98 | 2 | .21 |

Case 2 –

noise: 0.2 n\_samples: 100 set size: 1000

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 300

Value of *k* = 50

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy% | Error% | Times(s) |
| MLP | 79 | 21 | 5.41 |
| *k*-NN | 85 | 15 | .12 |
| SVM | 78.75 | 21.25 | .41 |
| Random Forest | 85 | 15 | .84 |

Case 3 –

noise: 0.3 n\_samples: 100 set size: 1000

Details:

Number of hidden nodes = 25

Learning constant = .3

Number of decision trees = 400

Value of *k* = 32

**Best Classifier: Random Forest**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 54.5 | 45.5 | 8.69 |
| *k*-NN | 57.75 | 42.25 | .12 |
| SVM | 54.25 | 45.75 | .52 |
| Random Forest | 58.5 | 41.5 | 1.4 |

Case 4 –

noise: 0.1 n\_samples: 200 set size: 2000

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 200

Value of *k* = 55

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 95.875 | 4.125 | 10.77 |
| *k*-NN | 97.625 | 2.375 | .36 |
| SVM | 95.875 | 4.125 | .16 |
| Random Forest | 97.25 | 2.75 | .46 |

Case 5 –

noise: 0.2 n\_samples: 200 set size: 2000

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 300

Value of *k* = 55

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 80.5 | 19.5 | 10.77 |
| *k*-NN | 89 | 11 | .47 |
| SVM | 84.25 | 15.75 | .47 |
| Random Forest | 88 | 12 | 1.19 |

Case 6 –

noise: 0.3 n\_samples: 200 set size: 2000

Details:

Number of hidden nodes = 25

Learning constant = .3

Number of decision trees = 200

Value of *k* = 63

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 56.25 | 43.75 | 17.51 |
| *k*-NN | 62.75 | 37.25 | .51 |
| SVM | 56.5 | 43.5 | 1.08 |
| Random Forest | 61.625 | 38.375 | .96 |

Case 7 –

noise: 0.1 n\_samples: 50 set size: 500

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 300

Value of *k* = 15

**Best Classifier: Random Forest**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 97 | 3 | 2.85 |
| *k*-NN | 97 | 3 | .03 |
| SVM | 95 | 5 | .13 |
| Random Forest | 96.5 | 3.5 | .21 |

Case 8 –

noise: 0.2 n\_samples: 50 set size: 500

Details:

Number of hidden nodes = 15

Learning constant = .3

Number of decision trees = 200

Value of *k* = 25

**Best Classifier: Random Forest**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 74 | 26 | 2.83 |
| *k*-NN | 82.5 | 17.5 | .09 |
| SVM | 79 | 21 | .09 |
| Random Forest | 84.5 | 15.5 | .2 |

Case 9 –

noise: 0.3 n\_samples: 50 set size: 500

Details:

Number of hidden nodes = 25

Learning constant = .3

Number of decision trees = 400

Value of *k* = 30

**Best Classifier: k-NN**

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Error | Times |
| MLP | 57 | 43 | 4.59 |
| *k*-NN | 61 | 39 | .04 |
| SVM | 56 | 44 | .15 |
| Random Forest | 60.5 | 39.5 | .45 |

Explanation of Software:

I wrote a java program which creates arrays of size = 49 and assigns the value of each bit based on a 7x7 view of that particular letter. The program then prints to the terminal a set of 10\*n\_samples vectors in the format of a .arff file. As the program prints each bit, a certain percentage of noise is randomly applied. After the program is done running, I copy and paste the contents of the terminal into my .arff file as the data set to apply the classifiers to. Every time I need to change the noise or n-samples, I need to reprint a new data set to test. After this, I open the .arff file on WEKA and run each classifier on the set of data. I set WEKA to randomly divide the set into 60% to train and 40% to test, then go into the settings for the specific classifier being used and make adjustments such as number of hidden layers and k-NN in order to get the optimal solution for each case.

Observations and Conclusion:

When it came to accuracy, k-NN and Random forest were pretty even in the number of cases in which they had the best performance. However, k-NN was far better in terms of time. It did surprise me that k-NN was generally faster as I just assumed Random Forrest would be the best. However, after watching the WEKA tutorial videos, I remembered the professor mention that k-NN is most useful when trying to classify noise and so it now makes sense to me that k-NN would do better .On average, k-NN was absolutely the best, not just with accuracy but also speed.