REPORT(CIS 6930)

Deep Bodra
58011841
d.bodra@ufl.edu

Classifiers

1. Linear SVM
    a. Hyperparameters
        i. Regularization = 1
        ii. 10 fold cross validation
    b. Output

```
----------Linear SVM Classifier----------
Dataset: tictac_final.txt
Cross Validation accuracy:  [98.96, 98.96, 93.75, 100.0, 98.96, 100.0, 95.83, 98.96, 98.95, 100.0]
Average Cross validation accuracy:  98.44
Normalized Confusion Matrix
[[1. 0.]
 [0. 1.]]
Precision:  [1. 1.]
Recall:  [1. 1.]
Accuracy:  100.0

Dataset: tictac_single.txt
Cross Validation accuracy:  [79.12, 80.31, 82.14, 82.6, 82.29, 84.58, 82.9, 81.53, 84.73, 83.36]
Average Cross validation accuracy:  82.35
Normalized Confusion Matrix
[[0.9668 0.0012 0.0062 0.0019 0.0277 0.0142 0.0074 0.0236 0.    ]
 [0.046  0.7834 0.0247 0.037  0.0392 0.0227 0.0257 0.     0.0109]
 [0.0659 0.0379 0.8068 0.0234 0.0277 0.0085 0.     0.0079 0.0153]
 [0.0332 0.0142 0.0123 0.7505 0.0287 0.0057 0.0165 0.0079 0.0196]
 [0.0441 0.0142 0.0103 0.0019 0.9006 0.0085 0.0147 0.     0.0022]
 [0.0153 0.0107 0.0021 0.     0.0086 0.8385 0.0092 0.0236 0.0044]
 [0.0352 0.0118 0.0051 0.0136 0.0019 0.     0.8474 0.0039 0.0065]
 [0.0109 0.0107 0.0062 0.0136 0.0076 0.     0.0037 0.8071 0.    ]
 [0.0275 0.0095 0.0062 0.0058 0.0086 0.     0.     0.     0.8497]]
Precision:  [0.7766 0.8768 0.9171 0.8851 0.8571 0.9338 0.9165 0.9234 0.9353]
Recall:  [0.9668 0.7834 0.8068 0.7505 0.9006 0.8385 0.8474 0.8071 0.8497]
Accuracy:  86.06319645855595
```

2. K Neighbors
    a. Hyperparameters
        i. Number of neighbors (k) = 1
        ii. 10 fold cross validation
    b. Output

```
----------K-Neighbors Classifier----------
tictac_final.txt
Cross Validation accuracy:  [100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0]
Average Cross validation accuracy:  100.0
Normalized Confusion Matrix
[[1. 0.]
 [0. 1.]]
Precision:  [1. 1.]
Recall:  [1. 1.]
Accuracy:  100.0

tictac_single.txt
Cross Validation accuracy:  [86.74, 89.16, 88.55, 87.79, 90.23, 90.08, 88.55, 89.47, 89.01, 90.53]
Average Cross validation accuracy:  89.01
Normalized Confusion Matrix
[[0.9923 0.0036 0.     0.0019 0.0038 0.0028 0.0037 0.0039 0.    ]
 [0.0013 0.9893 0.     0.0039 0.     0.     0.0074 0.     0.0022]
 [0.0006 0.0036 0.9908 0.0019 0.0019 0.     0.0037 0.     0.    ]
 [0.0013 0.     0.001  0.9844 0.     0.0057 0.     0.0118 0.    ]
 [0.0006 0.0024 0.001  0.     0.9933 0.     0.0037 0.     0.0022]
 [0.0006 0.     0.     0.     0.     0.9943 0.     0.     0.0022]
 [0.0013 0.     0.001  0.     0.     0.0113 0.9871 0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.9961 0.0022]
 [0.0013 0.     0.     0.0019 0.     0.     0.0074 0.     0.9847]]
Precision:  [0.993  0.9905 0.9969 0.9902 0.9943 0.9804 0.9746 0.9844 0.9912]
Recall:  [0.9923 0.9893 0.9908 0.9844 0.9933 0.9943 0.9871 0.9961 0.9847]
Accuracy:  99.05357960616699
```

3. MLP Classifier
    a. Hyperparameters
        i.

Parameter	tictac_final.txt	tictac_single.txt
Layer	2 layers with 9 neurons each	2 layers with 27 neurons each
Activation function	Relu	Relu
Optimizer	Adam	Adam
Learning Rate	0.01	0.01

        ii. 10 fold cross validation
    b. Output

```
----------MLP Classifier----------
Dataset: tictac_final.txt
Cross Validation accuracy:  [100.0, 100.0, 95.83, 100.0, 100.0, 100.0, 96.88, 98.96, 100.0, 100.0]
Average Cross validation accuracy:  99.17
Normalized Confusion Matrix
[[1. 0.]
 [0. 1.]]
Precision:  [1. 1.]
Recall:  [1. 1.]
Accuracy:  100.0

Dataset: tictac_single.txt
Cross Validation accuracy:  [82.32, 84.12, 86.56, 82.44, 85.65, 85.95, 86.56, 85.95, 86.26, 87.33]
Average Cross validation accuracy:  85.32
Normalized Confusion Matrix
[[0.9303 0.0142 0.0154 0.0351 0.0258 0.034  0.0331 0.0157 0.0065]
 [0.0064 0.9207 0.0113 0.0214 0.0124 0.0142 0.0239 0.     0.0087]
 [0.0083 0.0142 0.8931 0.0331 0.022  0.0283 0.0368 0.0118 0.0131]
 [0.0026 0.0095 0.     0.924  0.0105 0.0113 0.0165 0.0039 0.0044]
 [0.0058 0.013  0.0021 0.0117 0.9598 0.0057 0.0147 0.0039 0.0065]
 [0.0006 0.0083 0.001  0.0136 0.0019 0.9178 0.0037 0.0276 0.0044]
 [0.0038 0.0024 0.0021 0.0039 0.0019 0.0028 0.9724 0.     0.    ]
 [0.     0.0083 0.     0.     0.0057 0.0028 0.011  0.9213 0.    ]
 [0.0083 0.0237 0.001  0.0097 0.0038 0.0113 0.0018 0.0039 0.8932]]
Precision:  [0.9629 0.9078 0.9645 0.8778 0.9194 0.8926 0.8729 0.9323 0.9535]
Recall:  [0.9303 0.9207 0.8931 0.924  0.9598 0.9178 0.9724 0.9213 0.8932]
Accuracy:  92.76446344069608
```

Evaluation of classifiers

1. Training classifiers with 10% of the data
    a. Tictac_final.txt
        i. For this dataset it seems that all of the classifiers almost converged because the task of finding if the board is final or not is not so difficult and can be learnt in small number of supervised examples
        ii. Accuracy of Linear SVM, KNeighbors and MLP were found to be 77%, 80% and 90% respectively
        iii. It is highly likely that the MLP overfitted instead of actually learning something
    b. Tictac_single.txt
        i. For this dataset, 1/10th of the data is not enough to learn
        ii. The highest accuracy was achieved by MLP and it was 68%
        iii. Despite this, all the models were certainly underfitted
    c. Output (1/10th of the dataset)

```
----------Linear SVM Classifier----------
Dataset: tictac_final.txt
Cross Validation accuracy:  [70.0, 60.0, 90.0, 60.0, 90.0, 77.78, 77.78, 66.67, 88.89, 88.89]
Average Cross validation accuracy:  77.0
Normalized Confusion Matrix
[[0.7714 0.1333]
 [0.     1.    ]]
Precision:  [1.     0.8824]
Recall:  [0.7714 1.    ]
Accuracy:  91.57894736842105

Dataset: tictac_single.txt
Cross Validation accuracy:  [65.15, 65.15, 68.18, 59.09, 68.18, 60.0, 60.0, 67.69, 53.85, 67.69]
Average Cross validation accuracy:  63.5
Normalized Confusion Matrix
[[0.9669 0.     0.0233 0.0175 0.0099 0.     0.0139 0.     0.    ]
 [0.0265 0.814  0.0116 0.0175 0.0396 0.     0.0556 0.0303 0.0227]
 [0.0464 0.0465 0.7558 0.     0.0693 0.     0.0278 0.     0.0227]
 [0.0331 0.     0.0581 0.6491 0.0594 0.     0.0278 0.     0.0455]
 [0.0596 0.     0.     0.     0.9109 0.     0.     0.     0.    ]
 [0.0199 0.0349 0.     0.0526 0.0099 0.44   0.0278 0.0303 0.0227]
 [0.0464 0.     0.0233 0.0526 0.0198 0.     0.8056 0.     0.    ]
 [0.0066 0.0465 0.     0.0175 0.     0.     0.0278 0.7576 0.    ]
 [0.0397 0.0116 0.     0.0702 0.0396 0.     0.     0.     0.6591]]
Precision:  [0.7766 0.8537 0.8667 0.74   0.7863 1.     0.8169 0.9259 0.8529]
Recall:  [0.9669 0.814  0.7558 0.6491 0.9109 0.44   0.8056 0.7576 0.6591]
Accuracy:  81.37404580152672
```

```
----------K-Neighbors Classifier----------
tictac_final.txt
Cross Validation accuracy:  [80.0, 80.0, 80.0, 80.0, 80.0, 77.78, 88.89, 77.78, 88.89, 66.67]
Average Cross validation accuracy:  80.0
Normalized Confusion Matrix
[[1.     0.    ]
 [0.0714 0.9434]]
Precision:  [0.9333 1.    ]
Recall:  [1.     0.9434]
Accuracy:  96.84210526315789

tictac_single.txt
Cross Validation accuracy:  [42.42, 53.03, 54.55, 54.55, 54.55, 50.77, 49.23, 56.92, 58.46, 47.69]
Average Cross validation accuracy:  52.22
Normalized Confusion Matrix
[[0.9532 0.0353 0.0345 0.     0.0098 0.     0.     0.0208]
 [0.0117 0.9294 0.0115 0.     0.0098 0.     0.0179 0.     0.0208]
 [0.0058 0.0235 0.9195 0.0208 0.0196 0.     0.0179 0.     0.    ]
 [0.0058 0.     0.     0.9583 0.     0.     0.0179 0.     0.    ]
 [0.0058 0.0118 0.     0.     0.9608 0.     0.0179 0.     0.0208]
 [0.     0.0235 0.     0.     0.     0.9143 0.     0.0435 0.    ]
 [0.     0.     0.0115 0.     0.     0.     0.9821 0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     1.     0.    ]
 [0.     0.     0.     0.0625 0.     0.     0.     0.     0.9375]]
Precision:  [0.9702 0.908  0.9412 0.92   0.9608 1.     0.9322 0.9583 0.9375]
Recall:  [0.9532 0.9294 0.9195 0.9583 0.9608 0.9143 0.9821 1.     0.9375]
Accuracy:  94.80916030534351
```

```
----------MLP Classifier----------
Dataset: tictac_final.txt
Cross Validation accuracy:  [100.0, 100.0, 90.0, 80.0, 80.0, 100.0, 77.78, 77.78, 100.0, 100.0]
Average Cross validation accuracy:  90.56
Normalized Confusion Matrix
[[1. 0.]
 [0. 1.]]
Precision:  [1. 1.]
Recall:  [1. 1.]
Accuracy:  100.0

Dataset: tictac_single.txt
Cross Validation accuracy:  [75.76, 77.27, 60.61, 69.7, 71.21, 58.46, 61.54, 78.46, 63.08, 70.77]
Average Cross validation accuracy:  68.69
Normalized Confusion Matrix
[[0.9937 0.     0.01   0.     0.     0.     0.     0.     0.    ]
 [0.     1.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.96   0.0222 0.0097 0.     0.0169 0.0417 0.    ]
 [0.     0.0123 0.     0.9333 0.0097 0.0263 0.     0.     0.    ]
 [0.0127 0.     0.     0.     0.9806 0.     0.     0.     0.    ]
 [0.0063 0.0247 0.     0.0667 0.     0.8158 0.     0.0417 0.    ]
 [0.     0.     0.     0.     0.     0.     1.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.0169 0.9583 0.    ]
 [0.     0.     0.     0.0222 0.     0.     0.     0.     0.9787]]
Precision:  [0.9812 0.9643 0.9897 0.8936 0.9806 0.9688 0.9672 0.92   1.    ]
Recall:  [0.9937 1.     0.96   0.9333 0.9806 0.8158 1.     0.9583 0.9787]
Accuracy:  97.09923664122138
```

2. Effect of Noise in the test labels
    a. If only a few of the labels are noisy, then it can prevent the model from overfitting
    b. If a large number of labels are noisy then the model might not learn anything
    c. In some cases it may learn noise, and give totally incorrect predictions

Regressors

1. Linear Regressor
    a. Output

```
Accuracy:
Model 1
76.12578232330942
Model 2
82.27751488322393
Model 3
76.40054953442223
Model 4
82.00274767211113
Model 5
70.66096779117692
Model 6
81.9874828270493
Model 7
75.92733933750573
Model 8
82.15539612272936
Model 9
76.32422530911312
```

2. K Neighbors Regressor
    a. Hyperparameters
        i. Number of neighbors (k) = 2
        ii. 10 fold cross validation
    b. Output

```
stuxen@Neuron:~/Deep-Learning/Part 1$ python regressors.py
---------- K-Neighbors Regressor----------
Dataset: tictac_multi.txt
Cross Validation accuracy:  [90.21, 90.11, 90.03, 90.96, 90.92, 90.79, 89.55, 90.69, 89.74, 90.55]
Average Cross validation accuracy:  90.35

Accuracy:  97.12851303448159
```

3. MLP Regressor
    a. Hyperparameters
        i. Layers: 3
        ii. Neurons per layer: 27
        iii. Activation function: Sigmoid
        iv. Adam Optimizer
        v. Learning rate: 0.01
        vi. 10 fold cross validation
    b. Output

```
----------MLP Regressor----------
Dataset: tictac_multi.txt
Cross Validation accuracy:  [92.84, 92.33, 92.69, 92.71, 92.35, 92.81, 93.06, 92.54, 92.99, 92.21]
Average Cross validation accuracy:  92.65
Accuracy:  94.20783934598619
```

Evaluation of regressors

1. Why certain methods scale better to large datasets than others
    a. Methods like Linear Regression involve taking transpose, inverse and matrix multiplication of large matrices of the size of the dataset
    b. If the dataset is large enough so that it cannot fit in main memory, then these operation cannot be performed and the weight vectors cannot be calculated using normal equations
    c. However the Neural Networks can be trained in batches that can fit in memory
    d. There are optimization schemes like Stochastic Gradient Descent which do not require the entire dataset but only samples of it
2. Performance at Tic Tac Toe
    a. Linear Regressor
        i. If given an opportunity to win, then it places an O in the optimal spot in most cases
        ii. It sometimes also tries to block the opponent from winning
        iii. Overall it is good only for beginner level opponents
    b. KNeighbors Regressor and MLP Regressor
        i. They are both undefeatable
        ii. They block the opponent from winning at all times
        iii. They also try to win by taking corners and center positions
        iv. If there is a situation where it computers turn and the opponent will win in the next move and it can also win in this move, then it doesn't block the opponent but wins instead. I am amazed.

Conclusion

We used cross validation to avoid overfitting of machine learning models. We shuffled the dataset to facilitate the learning of models. We also used regressors to design a Tic Tac Toe bot that is undefeatable