R Noah Padilla

11/3/2020

Random Forest Classification of MNIST dataset and Regression of Particles dataset

**Introduction**

The purpose of this exercise was to tune the parameters of both the random forest classifier for classifying examples in the MNIST dataset as well as the random forest regressor for the Particles dataset.

**Methods**

I performed the method of only modifying the n\_estimators or the number of trees in the random forest for both the classifier and the regressor. In the classification, I started -50 from the default and made my way up. With the regressor I started at a lower number and made my way up since it was very time costly on my first test. After discovering that the regressor was to time consuming I decided to make another test and use the optimal max\_depth from my previous exercise to see if it would yield better results. The results are below -

**Results**

CLASSIFICATION DATA – only n\_estimators parameter was modified

|  |  |  |  |
| --- | --- | --- | --- |
| n\_estimators | Train Time | Test Time | Accuracy |
| 50 | 28.825665 secs | 0.179342 secs | 0.967000 |
| 100 | 56.486980 secs | 0.354030 secs | 0.969429 |
| 150 | 85.746600 secs | 0.523751 secs | 0.971286 |
| 200 | 114.060088 secs | 0.692670 secs | 0.971429 |
| 250 | 142.863594 secs | 0.876005 secs | 0.970571 |
| 300 | 182.422549 secs | 1.100109 secs | 0.970429 |

REGRESSION DATA – only n\_estimators parameter was modified

|  |  |  |  |
| --- | --- | --- | --- |
| **n\_estimators** | **Train Time** | **Test Time** | **MSE** |
| 10 | 132.753099 secs | 1.548245 secs | .042233 |
| 20 | 263.488964 secs | 3.179020 secs | .040254 |
| 30 | 385.591375 secs | 4.292018 secs | .039666 |
| 40 | 512.790967 secs | 5.907152 secs | 0.039384 |
| 50 | 639.141176 secs | 7.237906 secs | 0.039186 |
| 60 | 742.210262 secs | 8.695497 secs | 0.039052 |
| 100 | 1271.615759 secs | 67.963341 secs | 0.038782 |
| 150 | 1977.593288 secs | 1412.404979 secs | 0.038656 |

REGRESSION DATA – only n\_estimators and max\_depth parameters were modified

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **max\_depth** | **N\_estimators** | **Train Time** | **Test Time** | **MSE** |
| 12 | 50 | 276.178408 secs | 0.961541 secs | 0.037777 |
| 12 | 100 | 571.345402 secs | 1.933005 secs | 0.037768 |
| 12 | 150 | 855.273388 secs | 2.904613 secs | 0.037766 |
| 12 | 200 | 1125.475703 secs | 3.815182 secs | 0.037763 |
| 12 | 250 | 1409.710879 secs | 4.409895 secs | 0.037763 |
| 12 | 300 | 1669.788249 secs | 5.799232 secs | 0.037762 |

**Conclusion**

In conclusion as the n\_estimators or number of trees in the forest increased, the accuracy and mse improved. Although they improved as the n\_estimators increased, there was a threshold where they peaked and then the accuracy and mse started to decay in performance. Since the results from the classifier was sufficient I decided to keep messing with the regressor by utilizing the optimal max\_depth from the last exercise which improved the mse by more than 1/10%.

**References**

[1] “3.2.4.3.1. sklearn.ensemble.RandomForestClassifier¶.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: 03-Nov-2020].

[2] “3.2.4.3.2. sklearn.ensemble.RandomForestRegressor¶,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Accessed: 03-Nov-2020].