Name: gtrivell

Rank: 22 as of 9/29

Score: 0.71

My methodology for approaching this project was to leave all the ‘dials’ untouched until I had the program outputting something that could be tested. My initial implementation had an accuracy of 0.61. Throughout this project, time had never been an issue. In every iteration, it completed within 30 seconds, so runtime was not recorded.

These adjustments were made in the given order for the Decision Tree iteration. The row with the highlighted accuracy is the adjustment I kept.

|  |  |  |
| --- | --- | --- |
| Min\_df | Features | Accuracy |
| 0.0 | 88112 | 0.61 |
| 0.0013 | 73567 | 0.67 |
| 0.005 | 55226 | 0.56 |
| 0.1 | 134 | 0.06 |

(Adjusting max\_df did not make a change)

|  |  |  |
| --- | --- | --- |
| Min\_samples\_leaf | Number of nodes | Accuracy |
| 1 | 24 | 0.67 |
| 3 | 21 | 0.67 |
| 7 | 16/17 | 0.71 |
| 8 | 16/17 | 0.46 |
| 9 | 16 | 0.46 |
| 11 | 15 | 0.46 |
| 15 | 14 | 0.45 |

|  |  |  |
| --- | --- | --- |
| Variance Threshold | Number of Features | Accuracy |
| 0 | 73549 | 0.71 |
| 0.0000008 | 73543 | 0.65 |
| 0.000005 | 58776 | 0.63 |
| 0.000015 | 28397 | 0.56 |
| 0.00003 | 3442 | 0.71 |
| 0.000035 | 1630 | 0.62 |

0.71 is the highest accuracy I could achieve with the Decision tree iteration.

The following is for a Multi Layer Perceptron (MLP) iteration of the program. I initially believed that adjusting when it would early stop and how much of the training data was used for validation would be the best approach, but the accuracy for those were terrible (<0.30) for every adjustment. I only began getting passable results when changing the solver from ‘adam’ to ‘lbfgs.’ Unfortunately, lbfgs does not have dials like adam does so there was less to test than there otherwise would have been.

(I kept the same min\_df that I used to vectorize the train data, so the number of features at 0 is the same as the Decision Tree’s)

|  |  |  |
| --- | --- | --- |
| Variance Threshold | Number of Features | Accuracy |
| 0 | 73549 | 0.43 |
| 0.000021 | 13614 | 0.12 |
| 0.000025 | 7405 | 0.67 |
| 0.00003 | 3442 | 0.69 |
| 0.000032 | 2532 | 0.59 |
| 0.000035 | 1630 | 0.57 |
| 0.00004 | 707 | 0.57 |

|  |  |
| --- | --- |
| Alpha | Accuracy |
| 1 | 0.42 |
| 0.1 | 0.45 |
| 0.001 | 0.42 |
| 0.0001 (Default) | 0.69 |
| 0.0000001 | 0.69 |

0.69 is the highest accuracy I could achieve with the MLP iteration.

Adjusting the activation resulted in terrible accuracies, so I kept the default.

At this point, I had just given that something must be fundamentally off with my program that I could not adjust. I would guess that using more types of feature reduction or selection may be the answer, but it is difficult to tell. It seems to peak around 0.70 in both iterations, which does not seem to be the fault of the implementations but of the training data’s format. If I worked on this assignment again, I would adjust the training data more before entering it in to either the Decision Tree or MLP algorithm. The min\_df seemed to have a large impact on the base accuracy so maybe a small adjustment there could have resulted in a large gain.