Data Science 1 – Assignment 2

Joseph Irving
Student ID: 1766731

1C) Explain how the tree size/depth affects model performance in the context of overfitting/underfitting.

Compared to the tree with a depth of 7, the tree at a depth of 3 may be underfitting given that its accuracy, precision, and recall score are all lower. Conversely, at a depth of 11 and 15 the model appears to be overfitting, given that these metrics appear to continue to decrease when increasing depth beyond 7. Of the 4, a depth of 7 achieves the highest of each of these metrics. As such, we can conclude that a size/depth that is too small might underfit and miss important patterns, where as too deep and the model runs the risk of overfitting more toward noise, becoming less generalizable to unseen data.

1D) Explain the meaning of the difference in accuracy, precision, and recall scores in relation to the task; only if there is a significant difference.

Each of these metrics is within 1% difference of each other for each given depth, suggesting no significant difference. While the model with the depth of 7 performs highest across all 3 metrics, its precision is slightly lower than its accuracy and recall, suggesting a slighting increased likelihood of false positives, though still better than other models.

2C) Explain how the number of neighbors affects model performance in the context of overfitting/underfitting.

As neighbors increase, up to 17, accuracy, precision, and recall all continue to improve. At 25 neighbors, we see slight decreases in accuracy and recall and minor precision improvements. With a lower number of neighbors, for example, at 3, the model is more sensitive to noise and influence from individual points and thus more at risk of overfitting. Increasing neighbors makes the model more generalizable and resistant to noise. However, too generalizable and the model becomes underfit missing out on patterns in the data. In the case of my models, increasing the number of neighbors improved accuracy, precision, and recall up to 17 neighbors, suggesting some overfitting in the fewer neighbor models. Beyond 17 seems to be moving towards an underfitted model.

2D) Explain the meaning of the difference in accuracy, precision and recall scores in relation to the task; only if there is a significant difference.

In my models, accuracy, precision, and recall scores all improved with increasing neighbors, to a point. In particular, precision showed the most significant increases. While the 17 and 25-neighbor models are very close on all 3 metrics, the 25-neighbor model has a more noticeable improvement in precision suggesting when it predicts a positive class for each label, it is more often correct. Beyond 3 neighbors, recall stays relatively consistent suggesting for each class, the proportion of actual positives the model correctly identified is more consistent across neighbors.

3C) Discuss the impact of different kernels on model performance.

Using a Euclidean distance metric, the linear kernel performed the highest on accuracy and recall, significantly more than other kernels and was also overall the most balance in terms of the 3 metrics. The polynomial kernel had the lowest of all 3 metrics by a significant margin, suggesting that it's a very poor option for this data set. RBF and sigmoid, though performing worse than linear on accuracy, both had better precision. However, when testing different values for C, at a value of 3 a linear kernel saw worse overall performance but rbf and sigmoid showed scores similar to the linear model with a C of 0.5.

3D) Explain the meaning of the difference in accuracy, precision and recall scores in relation to the task.

For SVM, the choice of kernel significantly impacts how the model draws decision boundaries. In this case, the dramatic difference in accuracy between linear and polynomial kernels suggests the decision boundaries of a linear kernel are more appropriate for this data set. The significantly low precision of the polynomial kernel means most of its predictions are incorrect. Furthermore, the large difference between accuracy and precision suggests an inflated accuracy, possibly from overpredicting a particular class with many samples in the set.

4) Interpret the tables you generated in questions 1B, 2B, 3B; compare the performance of the Decision Tree, K-NN and SVM models. Which model performs better? Why do you think that is the case? What would you recommend to further improve each model's performance?

With all 3 types of models, increasing the given parameter improved performance too a point, before worsening or leveling off. The decision tree's best performance was at a depth of 7 with an accuracy of 60.2%, K-NN was at 17 neighbors with an accuracy of 69.8%, and SVM with a linear kernel had an accuracy of 73.6%. Given this, it is clear SVM with a learn kernel has the best performance of all the models. This is likely due in part to the fact that this is such a high-dimensional data set and SVMs have an advantage in high-dimensional spaces because they can find a hyperplane to separate the classes best. To improve each model's performance, I would consider performing grid searches testing various values for each parameter to find the best group of hyperparameters. For the decision tree model, I would make a grid testing various max depths, sample splits, minimum leaves, and gini vs entropy. For K-NN the grid would consist of additional neighbors, potentially all values from 3 to 17, different weights and distance metrics. The SVM grid would include testing various values for C, gamma, and possible kernels. Though computationally expensive, this would automate the process of finding the best possible model configuration.