STAT 6650 Homework 3

María Victoria Liendro
Master's in Data Engineering
mzl0201@auburn.edu

Spring 2025

2) b) The best configuration for each size of nest is shown in Table 1. In Figure 1, we can see all the different configurations calculated. On the X-axis we have maxDepth, on Y-axis nest, on Z-axis q (number of features for each tree), to add a fourth dimension we set the color of the bubbles as the minSamplesLeaf. Finally, the fifth dimension is the accuracy, represented by the size of the bubbles. Similarly, in Figure 2, we plot the Out-Of-Bag (OOB) accuracy). In both plots we can see how as the depth increases the accuracy also increases. Also as q increases the accuracy follows that, we can clearly see it when the Depth is 2. The minSamplesLeaf does not appear to highly influence the accuracy result.

2) c) Using the optimal parameters, the model achieves an accuracy of 0.9998 on the testing set and 0.9932 on the OOB samples. While both results are good, the slightly lower OOB accuracy suggests a trade-off between training performance and generalization.

2) d) In Figure 3, we can see the feature importance for the configuration with the highest accuracy, the most important was the property value.

3) a) The two algorithms to try are the perceptron and averaged perceptron, for both I will follow the pseudocode provided by Daume, Chapter 4. As the metric I will use F1 score, we have an imbalance amount of data, so the F1 score will allow us to smooth that. For model assesment we will use K-fold Cross Validation to compare both models and be sure that the error metric it is not just a coincidence.

3) d) Since our data is imbalanced, we have 1389 and 481 from each class, it was decide to us F1 score. It is calculated as:

$$F_1 = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}$$

Figures 4 and Figure 5, respectively, show the test and train error along with the mistakes found in each epoch. We can see that the model is learning the training set because after each epoch the number of mistakes is lower. Looking at the 6th epoch there is a peak in

the test error, however, the number of mistakes slightly increase in the following epochs to decrease again in the lasts. So that point may have been a local minimum where luckly the model did not get stuck.

3)e) For the averaged algorithm the results are very close. Figures 6 and Figure 7, respectively, show the test and train error along with the mistakes found in each epoch. It was able to make 0 mistakes ain 12 epochs, as in the first algorithm.

3)f) The test error for the first algorithm in the last epoch was 0.0153, while for the second algorithm 0.0164. Since the first algorithm has a better test error and it is simpler, we could say it is a better alternative.

3)g) Figure 8 shows the 15 words with the highest weight and the 15 words with the lowest weight.

Table 1: Best configurations per nest with performance metrics

Nest	**maxDepth**	**minSamplesLeaf**	**q**	**OOB**	**Accuracy**
5	3	1	5	0.99322	0.99987
6	3	1	5	0.99073	0.99882
7	5	1	5	0.96929	0.99973
8	3	1	5	0.96904	0.99987
9	3	1	5	0.97237	0.99987
10	5	1	5	0.97830	0.99886
11	3	1	5	0.97720	0.99960
12	5	1	5	0.98002	0.99889
13	5	1	5	0.97098	0.99889
14	5	1	5	0.97305	0.99889
15	3	1	5	0.97266	0.99960
16	5	1	5	0.96848	0.99889
17	5	1	5	0.97025	0.99892
18	5	1	5	0.97186	0.99872
19	5	1	5	0.97216	0.99886
20	2	1	5	0.94279	0.99882
21	2	1	5	0.93859	0.99882
22	2	1	5	0.94131	0.99859
23	5	1	5	0.96476	0.99872
24	5	3	5	0.96622	0.99872
25	5	3	5	0.96757	0.99892

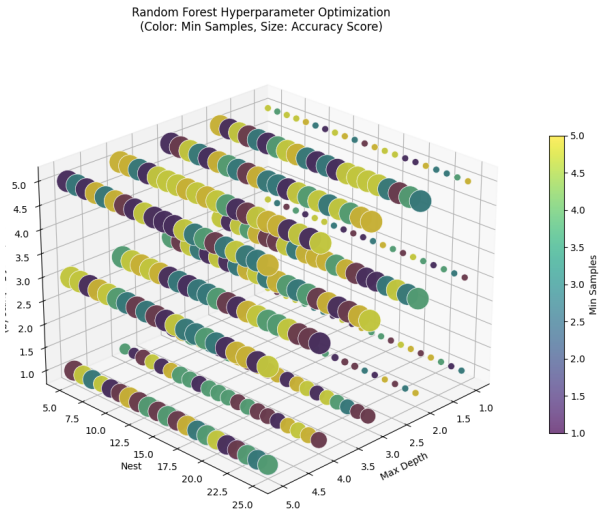

Figure 1:
Model accuracy across configurations.

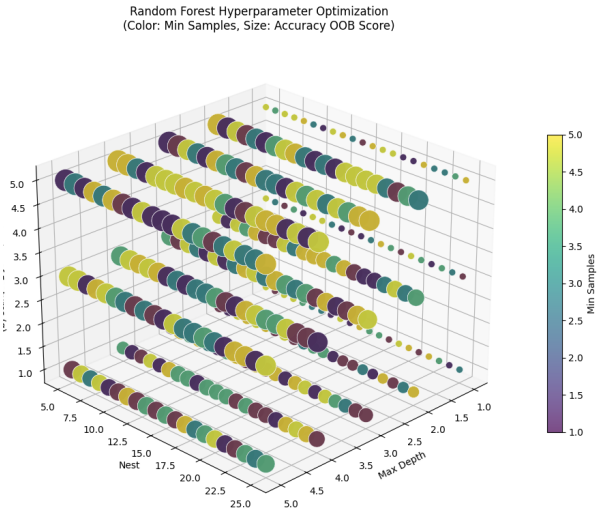

Figure 2:
Model OOB accuracy across configurations.

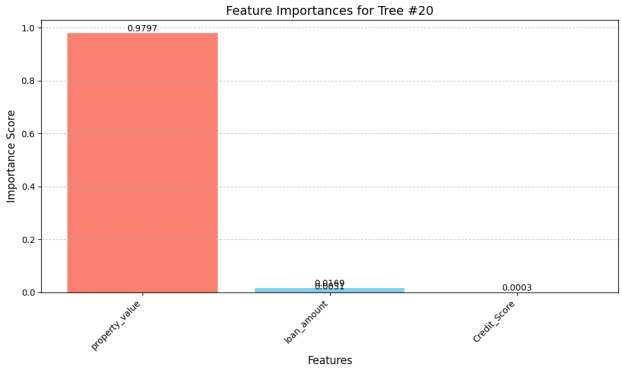

Figure 3: Feature importances for best configuration

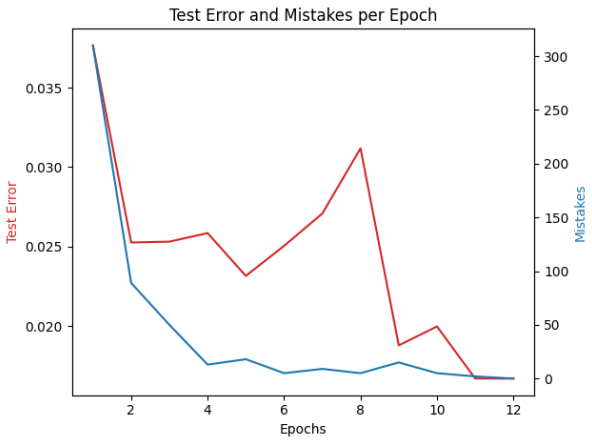

Figure 4:
Test error and mistakes per epoch

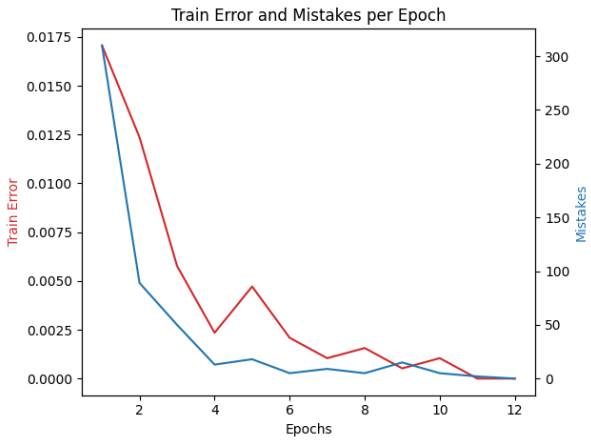

Figure 5:
Train error and mistakes per epoch

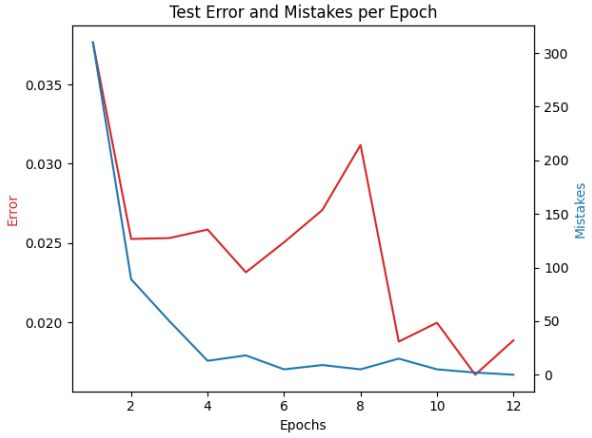

Figure 6:
Test error and mistakes per epoch

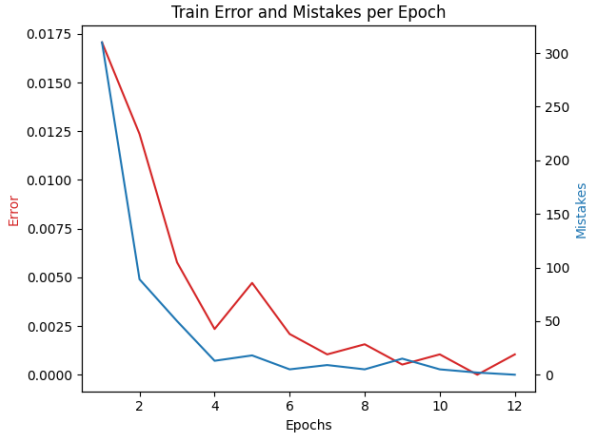

Figure 7:
Train error and mistakes per epoch

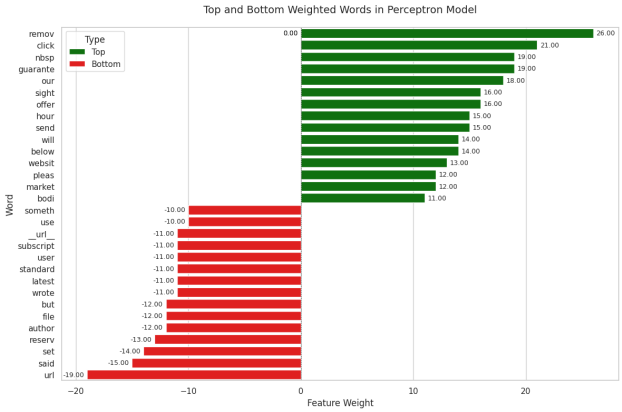

Figure 8: Top and bottom weighted words