Multiclass Classification on Newsgroups

Accuracy Results:

Classifier	Train Accuracy	Test Accuracy
Random Forest	0.929	0.634
Linear SVM	0.967	0.698
Multinomial Naïve Bayes	0.956	0.700
Bernoulli Naive-Bayes	0.599	0.458

Why these 3 methods:

I picked Random Forest, Linear SVM, and Multinomial Naïve Bayes because each of them has different advantages over each other. Random Forest can reduce overfitting through bagging; Linear SVM can maximize the classification margin; and, Multinomial Naïve Bayes works best with discrete features in the dataset. I want to compare the results of these 3 methods and study the essential difference behind each algorithm.

Result:

The result somehow matches my expectation but I expected higher accuracy for my test accuracy. All 3 classifiers achieved high train accuracy through cross validation; however, the trained classifiers only generated around 60 to 70 percent accuracy on the test set. I was expecting the test result would be closer to the training result as I didn't observe overfitting problems during the training stage.

Method to find best hyper-parameter:

I applied the following steps to tune hyper-parameters for each model.

1. Identifying the critical hyper-parameters for each algorithm
2. Setting the parameter grid for the searching range
3. Applying GridSearchCV() function to find the best values over the parameter grid
4. Tuning parameters to reduce the searching range based on the cross-validation result
5. Checking if the result converges. If not, going back to step 2 until the best values is picked

Classifier	Tuned-Parameter	Final Value
Random Forest	Number of trees (n_estimators)	370
Linear SVM	Penalty (C)	0.5
Multinomial Naïve Bayes	Smoothing Parameter (Alpha)	0.01

Confusion Matrix

Based on the confusion matrix, "comp.os.ms-windows.misc" and "comp.sys.ibm.pc.hardware" are two classes that my best performed classifier is confused about. There are total of 74 wrong classified data between these two classes.

(confusion matrix on next page)

Confusion Matrix:

0 [[0. 2. 2. 0. 1. 3. 3. 3. 6. 14. 3. 3. 4. 10. 15. 67. 8. 15. 5. 13.]

1 [6. 0. 22. 9. 7. 18. 8. 2. 4. 9. 0. 7. 11. 2. 10. 2. 0. 2. 1. 0.]

2 [4. 20. 0. 34. 16. 12. 3. 2. 2. 20. 1. 2. 1. 8. 10. 1. 3. 1. 3. 2.]

3 [1. 14. 40. 0. 24. 6. 11. 3. 0. 8. 1. 2. 27. 0. 1. 0. 0. 0. 0. 0.]

4 [0. 6. 8. 28. 0. 4. 14. 5. 4. 14. 3. 3. 18. 1. 4. 1. 1. 0. 0. 0.]

5 [0. 45. 34. 6. 5. 0. 4. 1. 0. 5. 0. 3. 6. 1. 4. 1. 0. 0. 0. 0.]

6 [0. 2. 2. 18. 11. 0. 0. 7. 2. 11. 1. 1. 7. 1. 3. 1. 4. 1. 0. 0.]

7 [4. 0. 3. 0. 1. 3. 14. 0. 13. 28. 2. 2. 22. 3. 7. 0. 3. 7. 5. 1.]

8 [4. 2. 0. 0. 1. 0. 9. 20. 0. 19. 1. 1. 9. 5. 7. 5. 5. 1. 5. 0.]

9 [3. 2. 0. 0. 1. 2. 7. 1. 6. 0. 26. 0. 1. 1. 1. 6. 0. 0. 5. 0.]

10 [3. 1. 0. 1. 0. 1. 0. 1. 0. 24. 0. 1. 0. 2. 2. 2. 2. 2. 0. 1.]

11 [1. 7. 5. 2. 7. 5. 7. 1. 3. 24. 1. 0. 8. 7. 7. 1. 18. 5. 6. 0.]

12 [3. 14. 11. 23. 11. 5. 20. 8. 11. 15. 2. 17. 0. 10. 9. 2. 2. 1. 2. 0.]

13 [5. 9. 2. 0. 1. 1. 8. 4. 5. 14. 5. 2. 8. 0. 3. 7. 3. 4. 6. 1.]

14 [3. 11. 2. 0. 1. 1. 5. 6. 3. 21. 3. 0. 14. 12. 0. 5. 5. 2. 7. 0.]

15 [18. 2. 3. 0. 0. 1. 0. 0. 1. 18. 1. 1. 2. 6. 4. 0. 1. 1. 3. 5.]

16 [5. 2. 2. 1. 2. 1. 4. 5. 6. 18. 1. 6. 3. 5. 9. 10. 0. 4. 21. 4.]

17 [21. 1. 1. 1. 0. 1. 1. 1. 7. 13. 3. 3. 3. 2. 1. 12. 9. 0. 11. 1.]

18 [15. 1. 0. 0. 0. 0. 0. 5. 5. 10. 3. 3. 2. 7. 11. 7. 95. 10. 0. 4.]

19 [41. 5. 3. 1. 0. 0. 4. 5. 3. 9. 3. 0. 0. 10. 5. 77. 22. 10. 8. 0.]]