Decision Tree:

File upload: NA

A2) Tree can only have 100% accuracy if we are using a part of training data for testing. During the training, tree gained the knowledge about that data, and now if we give same data to predict, it will give exactly same value. But in general, tree is not 100% accurate model of our data if part/all the train test is not used for testing purposes.

A3) Pruning reduces the size of decision trees by removing sections of the tree that are redundant or not critical for the classification task. It is used to overcome overfitting problem. Pruning increases the speed of the classification task and decreases model storage size.

A4) I changed max\_depth, min\_samples\_split and min\_samples\_leaf parameters. max\_depth is the maximum depth of the tree. If it is set to None, nodes expand until all leaves are pure. Limiting this number is pruning the tree. min\_samples\_split is the minimum number of samples required to split an internal node. It gives the best accuracy in our model for value of 50. min\_samples\_leaf is the minimum number of samples required to be at a leaf node. this value also can be used to prune the tree combined with max\_depth parameter. It gives the best accuracy when it is set to 2 in our model.

A5) Yes, results change. If we increase max\_depth and min\_samples\_split till a certain point, accuracy increases, after that point, there is no significant effect on the final accuracy. min\_samples\_leaf parameter gives the similar accuracy between values 2-5, and accuracy drops if it is higher than 5.

KNN:

File Upload: Done

A1) I experimented with four different K values: 3,5, 8 and 10. According to the graph, final accuracy is better with higher K (for K=10, accuracy is 81% approximately). But, after K=8, there is a slight increase in the accuracy. So, bigger K values after 10, accuracy will not change significantly.

A2) According to the graph, for books and dvd classes, there is a significant change in class f-scores. If K is higher, f-score is higher. For music and software there is a significant increase between K=3 and K=5, but for the other values, when K is higher, f-scores for these classes is also higher but slightly compared to each other. On the other hand, for camera and health classes, there is no correlation between K values and f-scores. K=8 has higher f-score for these classes compared to others. For camera class, for K=10; for health class, for K=5 has the lowest f-score.

A3) The K-nearest neighbors’ algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of K which increases the number of neighbors that contribute to the prediction, and in turn increases the bias of the model. When we increase K, the training error will increase (increase bias), but the test error may decrease at the same time (decrease variance). When K becomes larger, since it has to consider more neighbors, its model is more complex.

Comparison:

Time:

-----Naive Bayes Classifier-----

Training time: 0.34999537467956543

Testing time: 0.0670166015625

-----Decision Tree Classifier-----

Training time: 1.1140034198760986

Testing time: 0.05499553680419922

-----Random Forest Classifier-----

Training time: 5.160999536514282

Testing time: 0.2890148162841797

-----K-Nearest Neighbors Classifier combined with TF-IDF Vectorizer-----

Training time: 0.3690190315246582

Testing time: 0.4409801959991455

According to results provided above, Naïve Bayes and KNN classifier have the similar training time approximately with 0.35 and 0.37 secs approximately and their training time compared to other classifiers are faster. On the other hand, Naïve Bayes and Decision Tree classifier have the similar testing time with 0.07 and 0.05 secs approximately, and they are faster compared the other classifiers’ testing time. Random Forest classifier has the worst training time with 5 secs, which is not surprising since during the training many decision trees are built. KNN has the worst testing time with 0.44 secs compared the other classifiers’ testing time, but there is not any significant difference compared to Random Forest classifier’s testing time which is 0.29 approximately.

Best model: As a best model, I choose Random Forest classifier combined with Bag-of-words vectorizer since it has the highest accuracy on both development (89.5%) and test set (87.4%), and final accuracy after the cross validation (87.7%) compared to other methods that I experimented in this assignment (parameter choices can be seen in the code). I realized that KNN classifier only gives decent accuracy (77%) when it is combined with TF-IDF vectorized. Using Bag-of-words, final accuracy was only 44%. So, I decided to not to go with KNN. I tried to tune parameters of Decision Tree classifier, changing the max\_depth, min\_samples\_leaf, min\_samples\_split, and max\_features parameters. The best accuracy that I obtained was 78.3%. Then, I experimented with Naïve Bayes with combining the two vectorizer (Bag-of-words and TF-IDF) with FeatureUnion sklearn package, and then combined it with the Multinominal Naïve Bayes classifier. Final accuracy was 86.6%.