CSC6515 – Machine Learning for Big Data
Assignment 1
Jéssica Pauli de Castro Bonson (B00617515)

(a) Split the data randomly into a training set and a testing set (e.g. 65%-35%). Train a decision tree classifier using the train data. Report the confusion matrix and accuracy for both train and test data. Compare the train and test accuracy. Is there a big difference between train and test accuracy? Why? Finally, visualize the tree.

I split the data in 80%/20%, and used a seed of 6000. I tuned the configurations so the decision tree would have a better result in the training set, since the result with the default settings was barely above 50% of accuracy. The most important configuration parameter was cp (complexity parameter), that I set to 0.0001. The complexity parameter defines the minimum gain that must be obtained at each step in order to make a split worthwhile. So with cp = 0.0001 the tree was much bigger, prone to overfitting, but at least was able to obtain a good training accuracy of 0.6964, since it used various fine-grained rules.

The confusion matrix is below. It shows that the model had a hard understanding the rules for the classes 4 and 8, and frequently mixes the results of the classes 5, 6 and 7 with each other. All the attributes were used in the decision tree, with the most important ones being alcohol, density, and chlorides.

Training Dataset	
Accuracy	0.6964
Confusion Matrix	Reference Prediction 4 5 6 7 8 4 28 5 9 0 0 5 57 864 250 58 13 6 39 253 1340 183 42 7 7 38 141 443 44 8 0 6 19 20 41

The table below has the results for the test data:

Test Dataset	
Accuracy	0.5262
Confusion Matrix	Reference Prediction 4 5 6 7 8 4 3 2 3 0 0 5 21 180 105 14 2 6 7 98 250 79 17 7 1 10 75 75 12 8 0 1 6 8 4

Since there is a significant difference between the train and test accuracy, it can be considered that the model overfitted the training data and thus wasn't able to generalize for the new data in the test data. The tree visualization is in the file tree.pdf, and shows clearly why the model overfitted. The tree has 34947 nodes, so it has very specific rules for small samples of the dataset. It is interesting that since alcohol is the most important feature, it is the first one to create a branch in the tree.

(b) Repeat (a) and after training the classifier, prune the decision tree. Report the confusion matrix and accuracy for train and test data using the pruned tree and compare it to those obtained from (a). Visualize the tree and discuss the effect of pruning on generalization ability of the model. How the train and test accuracy changes by pruning? Why?

Training Dataset	
Accuracy	0.6472
Confusion Matrix	Reference Prediction 4 5 6 7 8 4 22 4 9 0 0 5 68 808 281 47 7 6 36 330 1334 309 55 7 5 18 131 345 63 8 0 6 4 3 15

Test Dataset	
Accuracy	0.5396
Confusion Matrix	Reference Prediction 4 5 6 7 8 4 3 2 3 0 0 5 20 168 93 9 1 6 8 116 289 104 21 7 1 5 53 63 11 8 0 0 1 0 2

The pruning decreased the training score (from 0.69 to 0.64) and increased the test score (from 0.526 to 0.539). This occurred since the more specific and fine-grained rules, created specifically for the training set, were removed from the bottom of the tree. So now it has a better generalization ability and is able to obtain a better score for unseen data. At the same time, the loss of the detailed rules also decrease the performance in the training set.

The visualization file is the archive ptree.pdf. The tree now has 3509 nodes, so around 30000 nodes with too specific rules where removed. There are still a lot of nodes, but around 10 times less than the unpruned tree. All features still are used by the tree, but now the features alcohol and density have an even higher importance, and volatile.acidity substituted chlorides as the third most important feature.

Additionally, even using preprocessing and tuning the parameters, the best performance that I was able to obtain in the test score was near 0.54, so I seems that a model based on rules such as the basic decision tree isn't well-suited for this dataset.

(c) Using 10-fold cross-validation, train and evaluate a random forest and a naïve Bayes classifier. Compare the accuracy of the two methods in terms of mean (μ) and standard deviation (σ) of accuracy in 10 folds. Eventually use a statistical significance test (e.g. student's t-test) and determine whether the two methods are significantly different or not. Use α=0.05 as the significance threshold.

Models trained with a train/test of 70%/30% and seed 6000, with cross-validation with 1 repeat.

Random Forest	
train mean accuracy	0.6771247
train std. dev. accuracy	0.02192085
test accuracy	0.6603
test confusion matrix	Reference Prediction 4 5 6 7 8 4 6 0 2 0 0 5 27 277 83 4 0 6 13 157 543 134 21 7 2 3 31 125 18 8 0 0 0 1 13

Naive Bayes	
train mean accuracy	0.4995739
train std. dev. accuracy	0.01942608
test accuracy	0.4774
test confusion matrix	Reference Prediction 4 5 6 7 8 4 10 13 7 0 1 5 23 255 196 27 5 6 13 149 301 103 16 7 2 19 151 127 26 8 0 1 4 7 4

The Naive Bayes got a much worse train and test score than the random forest, far worse than the difference in the standard deviations. I tried using preprocessing and tuning its configuration but it didn't improve. It may be the case that the assumption that the features probabilities are independent isn't true for this dataset, so it isn't able to model this dataset accurately. It is interesting that the random forest obtained a train score similar to the decision trees, but a way better test score, since it is able to better avoid overfitting. The only problem with the random forest was that it was also much slower than the other two models.

For the statistical significance test I performed 10 runs of each model with different seeds, and then obtained the accuracy score on the test dataset for these runs. Since Naive Bayes is calculated based on the dataset, and don't use random variables, only the random forest results were influenced by the various seeds. I used the non-parametric Mann–Whitney U-test for the hipothesis testing, with the following results:

```
W = 100, p-value = 6.34e-05
alternative hypothesis: true location shift is not equal to 0
```

So indeed the random forest model is statistically significant better than the Naive Bayes model for this dataset.