Part 1 (KNN)

Report the predicted class labels of each instance in the test set using the kNN with k=1,and the classification accuracy on the test set of kNN with k=1. Make sure you keep the same order as in the test set file. Also, remember to apply the Min-Max normalization for each feature.

Predicted class labels with k=1:

1. 3
2. 3
3. 3
4. 1
5. 1
6. 1
7. 1
8. 2
9. 1
10. 2
11. 2
12. 3
13. 3
14. 3
15. 1
16. 2
17. 3
18. 3
19. 1
20. 1
21. 3
22. 2
23. 2
24. 3
25. 2
26. 3
27. 2
28. 3
29. 2
30. 1
31. 2
32. 1
33. 2
34. 1
35. 2
36. 2
37. 2
38. 2
39. 2

40. 1
41. 2
42. 2
43. 3
44. 1
45. 2
46. 1
47. 3
48. 2
49. 2
50. 1
51. 3
52. 1
53. 1
54. 3
55. 3
56. 1
57. 1
58. 3
59. 1
60. 3
61. 3
62. 1
63. 2
64. 3
65. 2
66. 3
67. 3
68. 1
69. 1
70. 2
71. 1
72. 3
73. 2
74. 2
75. 1
76. 1
77. 1
78. 3
79. 1
80. 1
81. 2
82. 2
83. 3
84. 1
85. 2
86. 1
87. 1

88. 2
89. 1

The classification accuracy with k=1 is 94.382%.

(b) Report the classification accuracy on the test set of the k-nearest neighbour method where k=3, and compare and comment on the predictive performance of the two classifiers (k=1 and k=3).

At k=3, the accuracy is 95.506%. Being higher than at k=1, we can assume that k=1 is overfitting the predictions, meaning that outliers will be given too much power in predicting the classes for test set nodes. Using k=3 then means it is closer to being correctly fit- multiple neighbours are being considered for each node so the chance of outliers affecting the result decreases.

(c) Discuss the main advantages and disadvantages of k-Nearest Neighbour method. Also, discuss the impact of increasing and decreasing k, for example, what happens when k=total size of the dataset?

Advantages of knn:

Works with any number of classes or attributes

Disadvantages of knn:

Tough to implement if any attributes aren't continuous

Need to find optimal k outside of algorithm

Accuracy of k is an elbow shaped graph- low when k is low, but also low with maximum k. the optimal k is some small number. When k is low the algorithm overfits, when it's high it underfits. Setting k to equal the total size of the training set would lead every point to be classified as the most common class, regardless of location. This would make the accuracy equal to the proportion of the testing set occupied by the most common class.

(d) Describe the steps to apply the k-fold cross validation method for this dataset where k=5 (the number of folds). State the major steps.

The dataset would be shuffled, then randomly split into 5 groups. Then, for each iteration, a different group will be taken out to use as the test set while the other 4 are used to train the algorithm. The accuracy of the algorithm will then be checked and recorded, and the model will be discarded to move on to the next iteration. Then, after each of the 5 groups has been used as the test set once, the recorded accuracies will be analysed to evaluate the ability of the algorithm to make predictions.

Part 2 (Decision Tree)

You should first apply your program to the hepatitis-training and hepatitis-test files and report the classification accuracy in terms of the fraction of the test instances that it classified correctly.
Report the constructed decision tree classifier printed by your program.
Compare the accuracy of your decision tree program to the baseline classifier (which always predicts the most frequent class in the training set), and comment on any difference.

The accuracy of classification is 76% (19/25) on the Hepatitis dataset. However, the baseline predictor achieves 80% accuracy. This may be because the dataset has many more instances of one class than the other- more instances are classified as “live” rather than “die”. This leads to the algorithm being less able to predict “die” instances due to having less information.

The decision tree:

```
ASCITES = True:
    SPIDERS = True:
        VARICES = True:
            STEROID = True:
                Class live, prob=1.00
            STEROID = False:
                SPLEENPALPABLE = True:
                    FIRMLIVER = True:
                        Class live, prob=1.00
                    FIRMLIVER = False:
                        BIGLIVER = True:
                            SGOT = True:
                                Class live, prob=1.00
                            SGOT = False:
                                FEMALE = True:
                                    Class live, prob=1.00
                                FEMALE = False:
                                    ANOREXIA = True:
                                        Class die, prob=1.00
                                    ANOREXIA = False:
                                        Class live, prob=1.00
                        BIGLIVER = False:
                            Class live, prob=1.00
                SPLEENPALPABLE = False:
                    ANOREXIA = True:
                        Class live, prob=1.00
                    ANOREXIA = False:
                        Class die, prob=1.00
        VARICES = False:
            Class die, prob=1.00
    SPIDERS = False:
        FIRMLIVER = True:
            ANOREXIA = True:
                SGOT = True:
                    Class live, prob=1.00
                SGOT = False:
                    Class die, prob=1.00
            ANOREXIA = False:
                Class live, prob=1.00
        FIRMLIVER = False:
            SGOT = True:
                BIGLIVER = True:
                    Class live, prob=1.00
                BIGLIVER = False:
                    Class die, prob=1.00
            SGOT = False:
                Class live, prob=1.00
ASCITES = False:
    BIGLIVER = True:
        VARICES = True:
            FIRMLIVER = True:
                STEROID = True:
                    Class die, prob=1.00
                STEROID = False:
                    BILIRUBIN = True:
                        Class live, prob=1.00
                    BILIRUBIN = False:
                        Class die, prob=1.00
            FIRMLIVER = False:
                Class live, prob=1.00
```

”Pruning” (removing) some of leaves of the decision tree will make the decision tree less accurate on the training set. Explain:

What criteria would you use to decide which leaves can be pruned from your decision tree? Why would you choose this criterion?

> It could be good to prune leaves that are at the deepest levels of the tree, as these have been split the most times. This is an indication that they might be contributing to overfitting. Removing leaves from these levels could generalise the tree more for use in other data sets.

Why does pruning reduce accuracy on the training set?

> Pruning decision trees by removing leaves means that there is less splitting, and results in nodes with higher impurity. Therefore, more instances from the training set may be incorrectly classified, as the class of the combined node might not equal the class of the instances.

When pruning, should we expect the accuracy on the test set to decrease as well?

> Pruning makes the tree less accurate when used on the set it was trained on, but reduces overfitting, which could make it more accurate on the test set, as long as underfitting does not occur instead. This would occur if the splitting stopped too

early. Generally speaking, limited pruning would increase the accuracy on the test set.

Part 3 (Perceptron)

(a) Report on the accuracy of your perceptron. For example, did it find a correct set of weights? Did its performance change much between different runs?

The perceptron didn’t find a correct set of weights- it failed to converge within 100 iterations and reached an accuracy of 91.2% at its best. Its performance didn’t change significantly between runs.

(b) Explain why evaluating the perceptron’s performance on the training data is not a good measure of its effectiveness. You should split the dataset and perform a fairer evaluation.

The perceptron’s performance on the training data may not correctly measure its effectiveness as the training data may not be an accurate predictor of other possible data. By splitting the dataset, we can see whether the perceptron accurately predicts data it was *not* trained on, with multiple iterations.