Learning from Data - Week 2

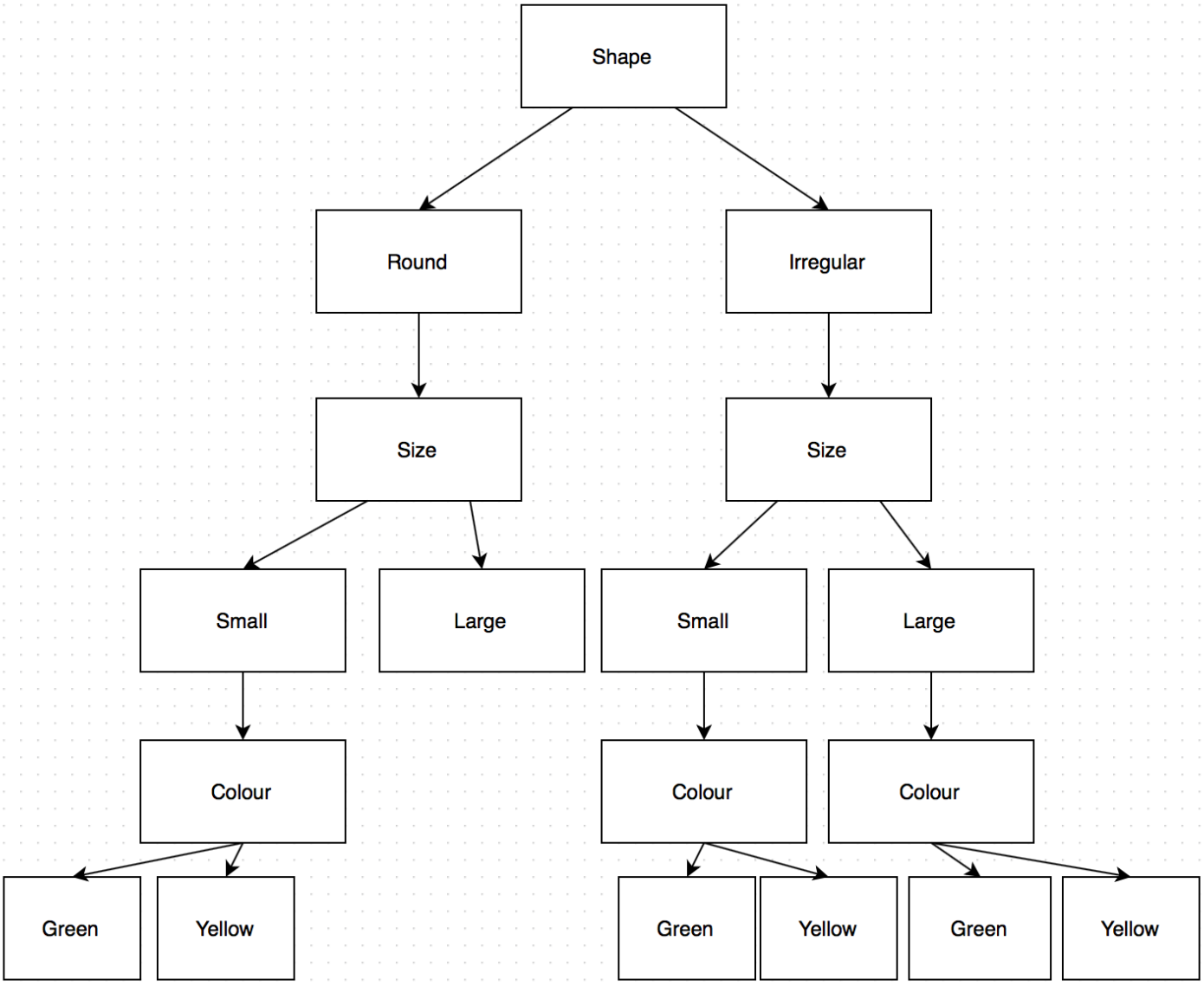
Assignment 2

Mart Busger op Vollenbroek
S2174634

Exercise 2.1

2.1.1

Based on the data the first branch will be shape, because this gives the most information gain (75% of the irregular shaped things are edible). The next branch would be size, because also 75% of the small sized things are edible. The final branch is colour, for which 61,5% of the things are edible. See the tree below where shape, size and colour are the decision nodes and round, irregular, small, large, green and yellow are leaf nodes. There's not a branch for every possible situation, because if a thing is round and large, it is always yellow.



2.1.2

I've used different parameters with the decision tree classifier. The dataset used for these results is the set supplied on nestor for assignment 1. First I tried to change the tree depth. After using a depth of 30, I gained the highest result which was 0.77182. Other values I assigned to the depth were ten, twenty, forty and fifty. The results are averages using 5-fold cross validation and can be found in table 1.

Depth	Accuracy
10	0.71283
20	0.76565
30	0.77182
40	0.76448
50	0.75965

Table 1: Results from tree depth

After these tests, the depth was fixed at 30. The next parameter in the classifier that was tuned was the amount of samples needed to create a new leaf. This was tested for one, two, three, four and five samples. The results from this can be found in table 2. As it turnes out, the default parameter of 1 gives the best results, so changing that will harm the system for this dataset.

Amount of samples for leaf	Accuracy
1	0.77299
2	0.76632
3	0.76366
4	0.77031
5	0.76549

Table 2: Results from minimum amount of samples for a new leaf

The last parameter that was tuned for this system was the amount of samples needed to create a split in in the tree (e.g. create a new decision node). For this test, the amount of samples used are ranged from 1 to 6 and the results from these tests can be found in table 3.

Amount of samples for split	Accuracy
1	0.77115
2	0.77215
3	0.77066
4	0.77215
5	0.77299
6	0.76849

Table 3: Results from minimum amount of samples for a new split

After changing several parameters from the system's classifier to prune the tree, the settings which gained the best result for me on this dataset are:
- depth=30
- min_samples_leaf=1 (default)
- min_samples_split=5

Seeing as the accuracy score using none of the above mentioned pruning methods is 0.76182 and the highest score using pruning is 0.77215, the amount of change is minimal.

Exercise 2.2

2.2.1

a) is accuracy better with a lower or higher K?
Accuracy (using F-scores) is better when using a lower K, but to a certain extend.

b) does overall performance plateau at a point?
Yes it does, it smoothens while going to K = 20 with a top of 0.76094 at K = 16

c) does class performance change significantly with varying values of K?
This varies per class. Camera, Dvd, Music and Software all relatively stay the same (they are improving of decreasing a little bit how higher K gets). Books however increases greatly when a higher K is chosen. The opposite goes for Health, which decreases fast when higher K's are chosen.

d) how does changing K affect the bias/variance trade-off?
Normally, kNN has high variance and low bias. Dependent on the dataset, changing K can have different effects. If the data is completely random, choosing a higher K may result in a higher accuracy. If the data is more structured however, a lower K will give better results.

2.2.2

	kNN (k=16)	Decision Tree	Naive Bayes
Train time	0,3124721050262451	5,110706090927124	0,34876394271850586
Test time	0,6496279239654541	0,09308290481567383	0,09998202323913574
Total	0,962100028991699	5,2037889957428	0,448745965957642

From these numbers, several findings can be gathered. First, based on training time alone, the decision tree classifier takes much more time than the other two classifiers. The other two score basically the same on training time. On testing time alone however, the decision tree classifier scores best, followed by Naive Bayes and kNN. Overall, Naive Bayes is the fastest classifier, the decision tree is the slowest, leaving kNN in the middle.

2.2.3

For my 'best' model, I chose a MultinomialNB with an alpha parameter of 0.23. I chose this because after testing all different classification methods, Naive Bayes seemed to get best results. After having chosen Naive Bayes, I tested for several alpha settings. The alpha setting is used for smoothing and the default is 1. This means that if the default parameter is used, there is some overfitting which influences the results badly.