Part 1: Decision Trees

Question 1.1 (5 marks): Consider the decision tree learning algorithm of Figure 7.7 and the data of Figure 7.1 from Poole & Mackworth [1], also presented below.

Suppose, for this question, the stopping criterion is that all of the examples have the same classification. The tree of Figure 7.6 was built by selecting a feature that gives the maximum information gain. This question considers what happens when a different feature is selected.

Figure 7.7

```
1: procedure Decision_tree_learner(Cs, Y, Es)
2:     Inputs
3:         Cs: set of possible conditions
4:         Y: target feature
5:         Es: set of training examples
6:     Output
7:         function to predict a value of Y for an example
8:     if stopping criterion is true then
9:         let v = point_estimate(Y, Es)
10:        define T(e) = v
11:        return T
12:    else
13:        pick condition c ∈ Cs
14:        true_examples := {e ∈ Es : c(e)}
15:        t1 := Decision_tree_learner(Cs \ {c}, Y, true_examples)
16:        false_examples := {e ∈ Es : ¬c(e)}
17:        t0 := Decision_tree_learner(Cs \ {c}, Y, false_examples)
18:        define T(e) = if c(e) then t1(e) else t0(e)
19:        return T
```

Figure 7.7: Decision tree learner

Figure 7.1

Example	*Author*	*Thread*	*Length*	*Where_read*	*User_action*
e_1	*known*	*new*	*long*	*home*	*skips*
e_2	*unknown*	*new*	*short*	*work*	*reads*
e_3	*unknown*	*followup*	*long*	*work*	*skips*
e_4	*known*	*followup*	*long*	*home*	*skips*
e_5	*known*	*new*	*short*	*home*	*reads*
e_6	*known*	*followup*	*long*	*work*	*skips*
e_7	*unknown*	*followup*	*short*	*work*	*skips*
e_8	*unknown*	*new*	*short*	*work*	*reads*
e_9	*known*	*followup*	*long*	*home*	*skips*
e_{10}	*known*	*new*	*long*	*work*	*skips*
e_{11}	*unknown*	*followup*	*short*	*home*	*skips*
e_{12}	*known*	*new*	*long*	*work*	*skips*
e_{13}	*known*	*followup*	*short*	*home*	*reads*
e_{14}	*known*	*new*	*short*	*work*	*reads*
e_{15}	*known*	*new*	*short*	*home*	*reads*
e_{16}	*known*	*followup*	*short*	*work*	*reads*
e_{17}	*known*	*new*	*short*	*home*	*reads*
e_{18}	*unknown*	*new*	*short*	*work*	*reads*
e_{19}	*unknown*	*new*	*long*	*work*	?
e_{20}	*unknown*	*followup*	*short*	*home*	?

Figure 7.6

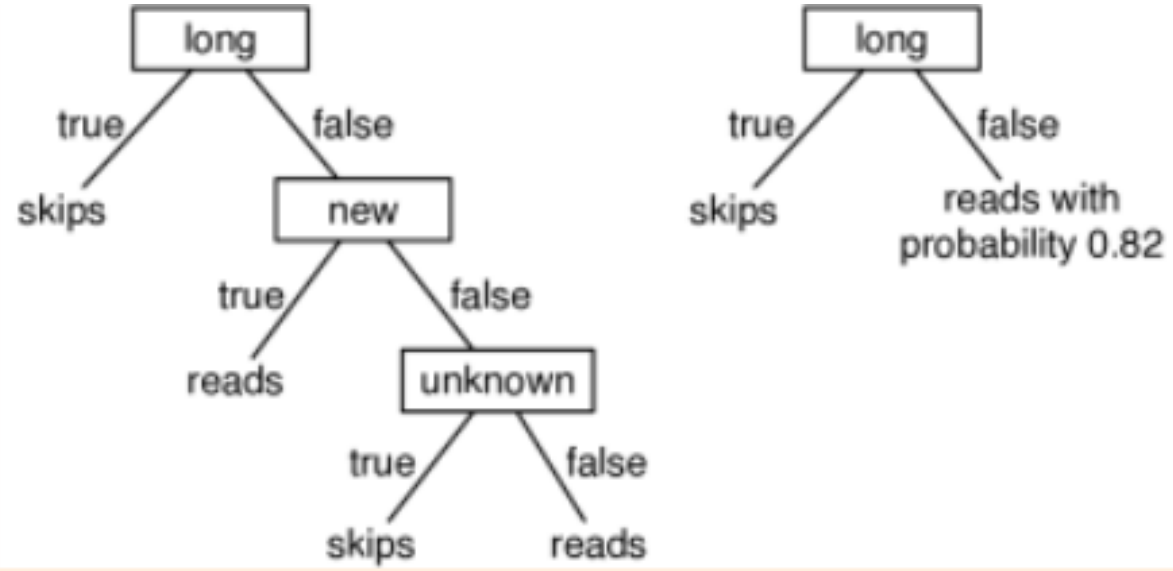


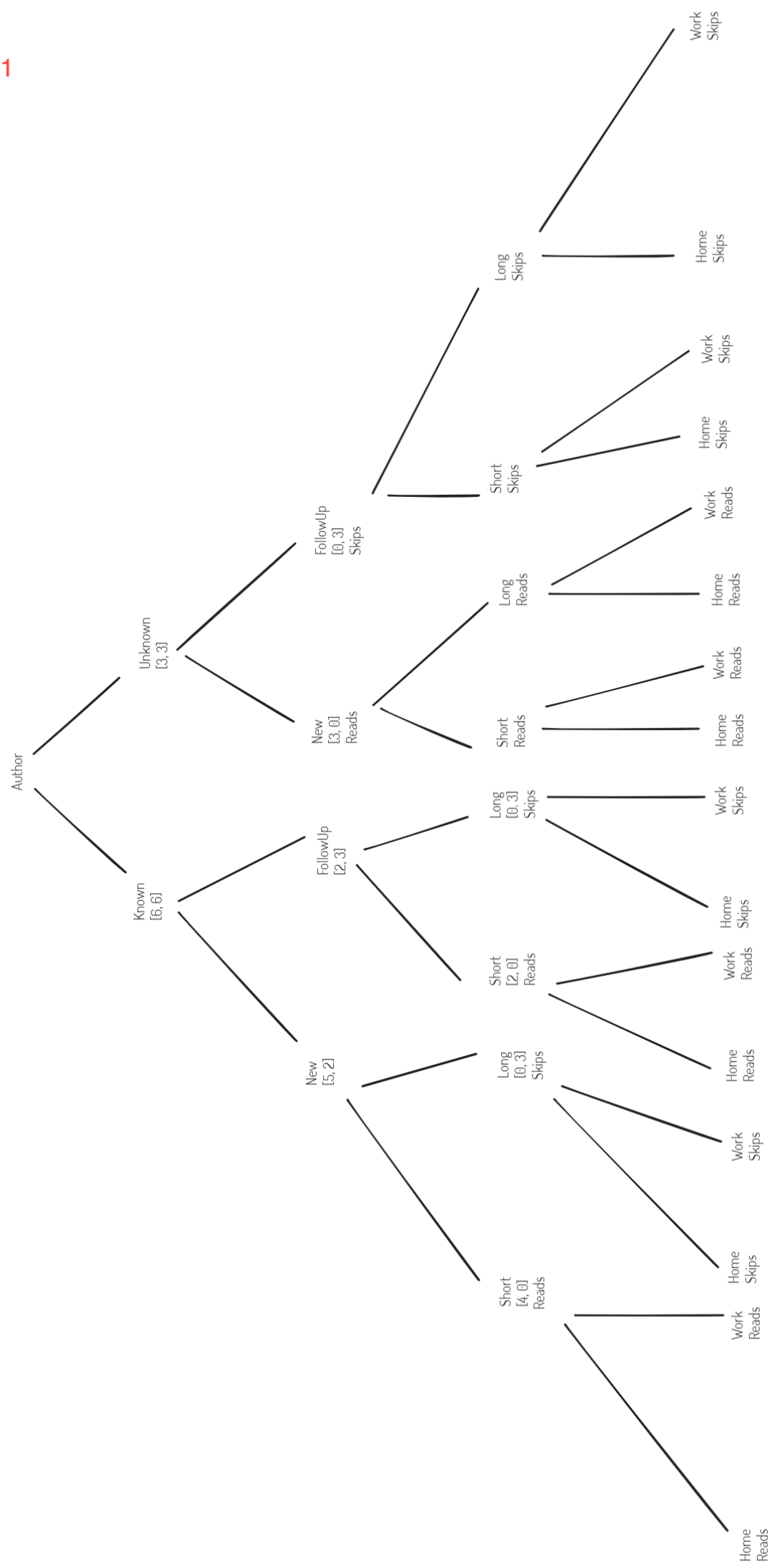
Figure 7.6: Two decision trees

a. Suppose you change the algorithm to always select the first element of the list of features. What tree is found when the features are in the order [Author, Thread, Length, WhereRead]? Does this tree represent a different function than that found with the maximum information gain split? Explain.

If the algorithm was changed from the maximum information gain split of [Length, Thread, Author] to the order of [Author, Thread, Length, WhereRead], it causes a larger decision tree with the maximum information gain split still becoming a decider of whether to skip or read a certain book.

Figure 1

Figure 1


Author
Unknown [3, 3]
Known [6, 6]
FollowUp [0, 3] Skips
New [3, 0] Reads
FollowUp [2, 3]
New [5, 2]
Long Skips
Short Skips
Long Reads
Short Reads
Long [0, 3] Skips
Short [2, 0] Reads
Long [0, 3] Skips
Short [4, 0] Reads
Work Skips
Home Skips
Work Reads
Home Reads
Work Skips
Home Skips
Work Reads
Home Reads
Work Reads
Home Reads
Work Skips
Home Skips
Work Reads
Home Reads
Work Skips
Home Skips
Work Reads
Home Reads

From figure 1, it represents the decision tree of an algorithm where it selects the first element of the features list. The entropy of author has an information gain of 0 indicating that there is no information gained, hence we can see that there is no change in our decision between the known and the unknown author. When going down the left-hand side of the decision tree for the next feature being threads it has a small information gain of approx. 15%. The new decision node has a higher probability of reads at 5/7 compared to skips being 2/7. Whereas followup has a higher probability of skipping being 3/5 compared to reads being 2/5. However, the feature with the highest information gain being length with a approx. 58% gain determines the decision of the tree where short is always reads and long is always skips. This left-hand side of the decision tree represents the same function as the maximum information gain split as figure1(author, thread, length) == figure7.6(author, thread, length). However, on the right-hand side of the decision tree, there is a slightly different function with figure1(unknown, new, long) = reads, while in figure7.6(unknown, new, long) = skips. This can be due to the data set missing the user action for this function. From figure 1, the feature where_read doesn't give any more information to the decision tree as the decision for reading or skipping is already made before the feature split.

b. What tree is found when the features are in the order [WhereRead, Thread, Length, Author]? Does this tree represent a different function than that found with the maximum information gain split or the one given for the preceding part? Explain.

For the decision tree where the features are ordered as [WhereRead, Thread, Length, Author] there is a similar occurrence to the decision tree from figure 7.6.

Figure 2

Figure 2

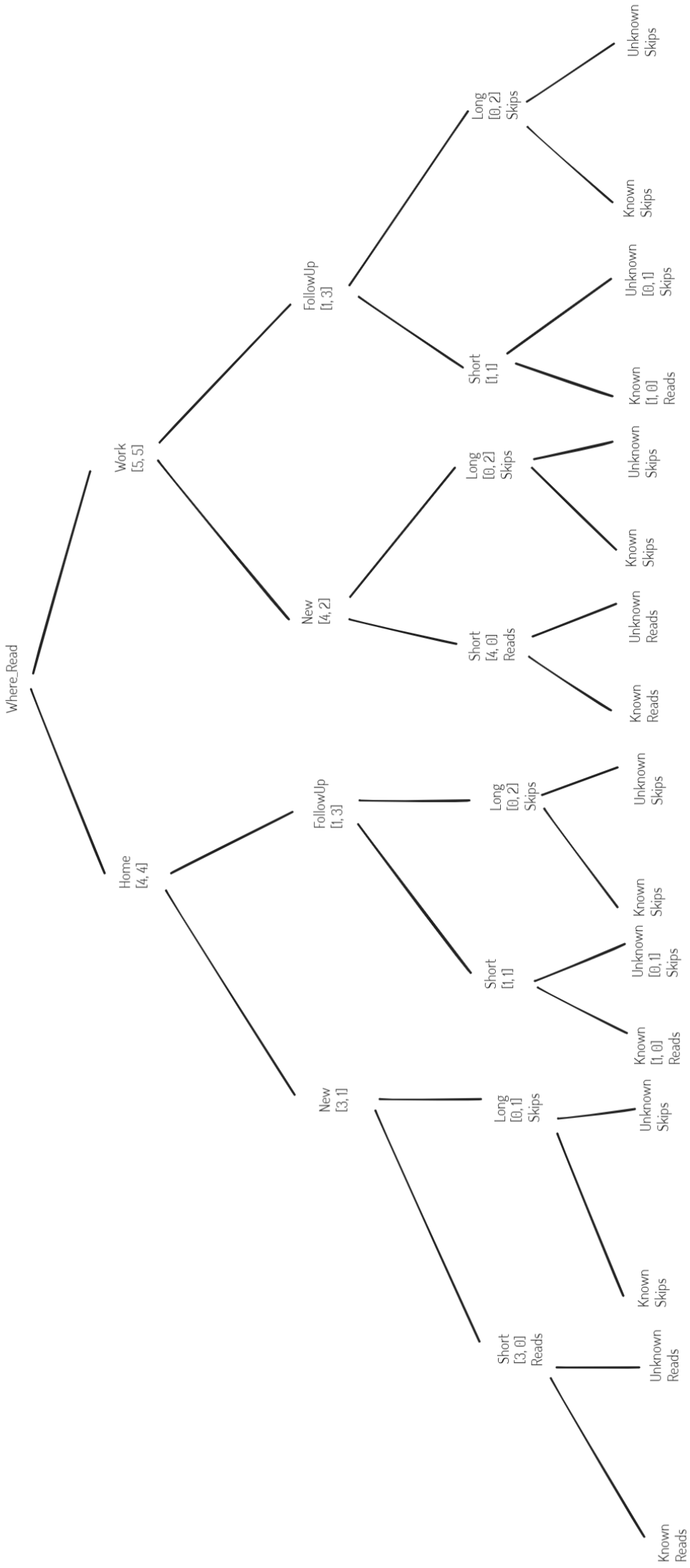

Where_Read
Work [5, 5]
Home [4, 4]
FollowUp [1, 3]
New [4, 2]
New [3, 1]
FollowUp [1, 3]
Long [0, 2] Skips
Short [1, 1]
Long [0, 2] Skips
Short [4, 0] Reads
Long [0, 1] Skips
Short [3, 0] Reads
Long [0, 2] Skips
Short [1, 1]
Long [0, 2] Skips
Short [4, 0] Reads
Long [0, 1] Skips
Short [3, 0] Reads
Unknown Skips
Known Skips
Unknown [0, 1] Skips
Known [1, 0] Reads
Unknown Skips
Known Skips
Unknown Reads
Known Reads
Unknown Skips
Known Skips
Unknown [0, 1] Skips
Known [1, 0] Reads
Unknown Skips
Known Skips
Unknown Reads
Known Reads

Figure 2 represents the same function in figure 7.6 as it classifies every training example the same for reads and skips. However, this decision tree is larger than the decision tree shown in 7.6 as the Where_read feature has an information gain of 0, hence giving liminal information.

c. Is there a tree that correctly classifies the training examples but represents a different function than those found by the preceding algorithms? If so, give it. If not, explain why.

There are 2^4 trees that will have different functions when correctly classifying the training examples as there are 4 training examples that are missing from the tree being [unknown, new, long, work], [unknown, new, long, home], [unknown, new, short, home] and [unknown, followup, short, work].

Question 1.2

The goal is to take out-of-the-box models and apply them to a given dataset. The task is to analyse the data and build a model to predict whether income exceeds $50K/yr based on census data.

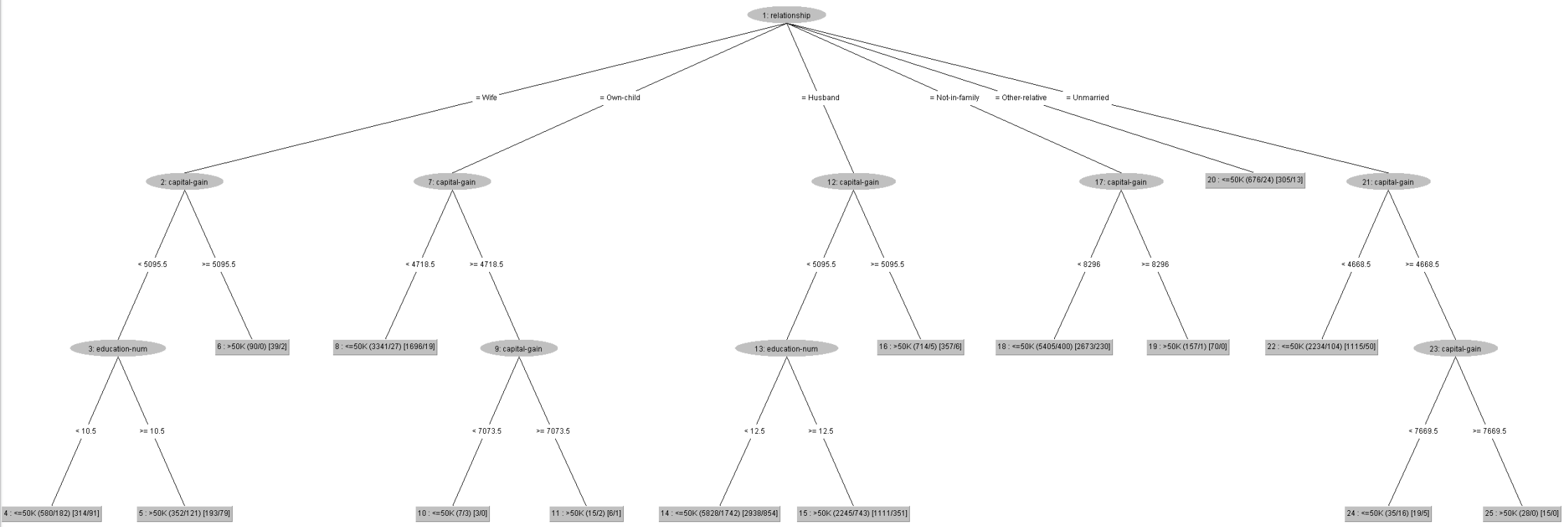
Experiment with pruning the tree, including trying different pruning settings. What affect do these settings have on the size of the tree and the accuracy?

In using the J48 Weka program on the adult data set to build a model to predict whether income exceeds $50k/yr I have used the following pruning settings being, testing on the supplied test within the adult_test file, cross validation, percentage split, Weka's ranker and attribute selection. In using the default training setting supplied by Weka the decision tree has a size of 710 with 564 leaves and an accuracy of 87.8566%. In using cross validation for the pruning of the tree for 15 folds or a percentage split of 66%, the tree is the same size with an accuracy of approx. 86% for both instances. When using the select attribute module and using the evaluator of information gain using the ranker method it shows that the best information gain that is higher than 0.1 is as follows relationship, marital-status and capital-gain. Based on this test, feature engineering is then implemented with removing all features except for income, relationship, marital-status and capital-gain. The tree produced is of size 64 with 35 leaves and an accuracy of 80%, which is a 7% decline. However, the tree is still very large so then I used a filter on the results using attribute selection but with BFS and the tree now includes the features education-num, relationship, capital gain and income. Then setting the maximum depth of the tree to 3 the following tree is built with the feature set [relationship, capital-gain, education-num, income]. These results show only an approx. 3% decrease in accuracy, but a large decrease to tree size to 25.

This creates the resulting decision tree below.

```
=== Summary ===

Correctly Classified Instances       27490               84.4262 %
Incorrectly Classified Instances      5071               15.5738 %
Kappa statistic                          0.5232
Mean absolute error                      0.2286
Root mean squared error                  0.3381
Relative absolute error                 62.5094 %
Root relative squared error             79.0638 %
Total Number of Instances            32561
```


1: relationship
= Wife
= Own-child
= Husband
= Not-in-family
= Other-relative
= Unmarried
2: capital-gain
7: capital-gain
12: capital-gain
17: capital-gain
20 : <=50K (676/24) [305/13]
21: capital-gain
< 5095.5
>= 5095.5
< 4718.5
>= 4718.5
< 5095.5
>= 5095.5
< 8296
>= 8296
< 4668.5
>= 4668.5
3: education-num
6 : >50K (90/0) [39/2]
8 : <=50K (3341/27) [1696/19]
9: capital-gain
13: education-num
16 : >50K (714/5) [357/6]
18 : <=50K (5405/400) [2673/230]
19 : >50K (157/1) [70/0]
22 : <=50K (2234/104) [1115/50]
23: capital-gain
< 10.5
>= 10.5
< 7073.5
>= 7073.5
< 12.5
>= 12.5
< 7669.5
>= 7669.5
4 : <=50K (580/182) [314/91]
5 : >50K (352/121) [193/79]
10 : <=50K (7/3) [3/0]
11 : >50K (15/2) [6/1]
14 : <=50K (5828/1742) [2938/854]
15 : >50K (2245/743) [1111/351]
24 : <=50K (35/16) [19/5]
25 : >50K (28/0) [15/0]