Violet McCabe
CS 135
Mike Hughes
11/16/2023

Problem 0

- ~~Code A~~
- ~~Code B~~
- Code C
- ~~Code D~~

Problem 1

- ~~Implementation 1A~~
- ~~Figure 1~~
- Short answer 1a
- ~~implementation 1B~~
- ~~short answer 1b~~

Problem 2:

- ~~Implementation A~~
- Implementation B
- ~~Table 2~~
  - ~~Rly rly close to done~~
- Implementation C
- Short answer 2a
- Short answer 2b
- Table 3

Problem 3

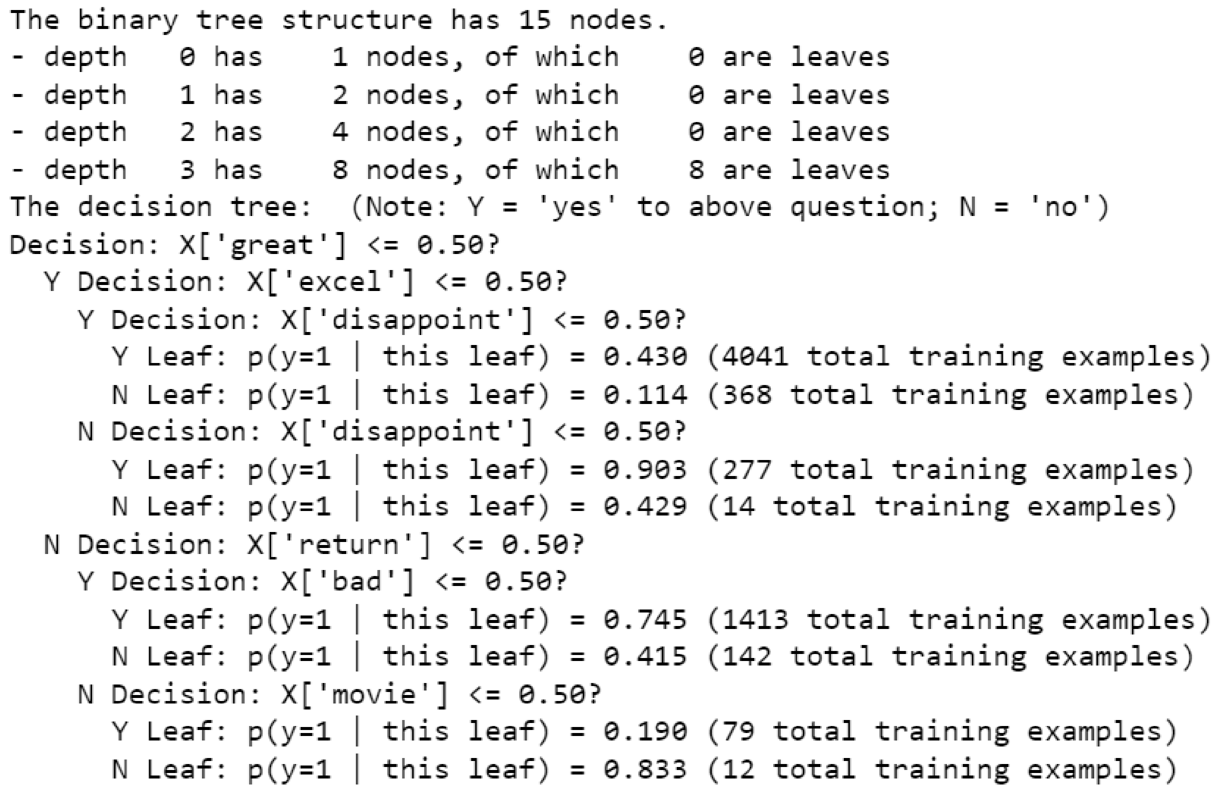
- Short answer 3a

~~0A~~ 0B 0C 0D
1a ~~1b~~ tables
2Ia 2Ib 2Ic
2a 2b Table 2 and Table 3
3a

Problem 0

Problem 1: Decision Trees for Review Classification

Figure 1



```
The binary tree structure has 15 nodes.
- depth   0 has    1 nodes, of which    0 are leaves
- depth   1 has    2 nodes, of which    0 are leaves
- depth   2 has    4 nodes, of which    0 are leaves
- depth   3 has    8 nodes, of which    8 are leaves
The decision tree:  (Note: Y = 'yes' to above question; N = 'no')
Decision: X['great'] <= 0.50?
  Y Decision: X['excel'] <= 0.50?
    Y Decision: X['disappoint'] <= 0.50?
      Y Leaf: p(y=1 | this leaf) = 0.430 (4041 total training examples)
      N Leaf: p(y=1 | this leaf) = 0.114 (368 total training examples)
    N Decision: X['disappoint'] <= 0.50?
      Y Leaf: p(y=1 | this leaf) = 0.903 (277 total training examples)
      N Leaf: p(y=1 | this leaf) = 0.429 (14 total training examples)
  N Decision: X['return'] <= 0.50?
    Y Decision: X['bad'] <= 0.50?
      Y Leaf: p(y=1 | this leaf) = 0.745 (1413 total training examples)
      N Leaf: p(y=1 | this leaf) = 0.415 (142 total training examples)
    N Decision: X['movie'] <= 0.50?
      Y Leaf: p(y=1 | this leaf) = 0.190 (79 total training examples)
      N Leaf: p(y=1 | this leaf) = 0.833 (12 total training examples)
```

Short answer 1a

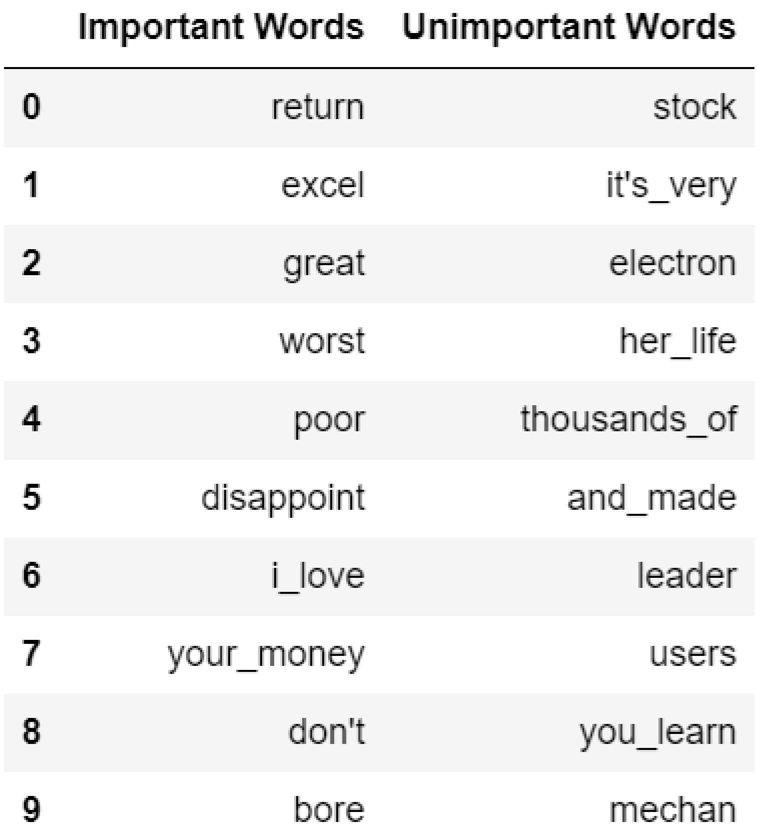
It can occur if the gini impurity is already really low and it is not worth splitting further. This tree does not have an internal node with two child leaves with the same sentiment.

Short answer 1b

The best tree uses a max depth of 128 and a min_samples_leaf of 1.

Problem 2: Random Forests for Review Classification

Table 2



	Important Words	Unimportant Words
0	return	stock
1	excel	it's_very
2	great	electron
3	worst	her_life
4	poor	thousands_of
5	disappoint	and_made
6	i_love	leader
7	your_money	users
8	don't	you_learn
9	bore	mechan

Short Answer 2a

Max_features: 10

Maximum max_features: 6346 b/c size of dataset

Tuning max_features: if it is put automatically at the maximum value it causes overfitting and is really computationally inefficient.

Short Answer 2b

- Overfitting, high computational cost, and influences the biases.
- You can overfit the model if you make n_estimators too large

Table 3

Method	Max Depth	Number of trees	Train BAcc	Valid BAcc	Test BAcc
Simple Tree	3		0.646	.649	.639
Best Tree	128		0.998	0.737	0.726

Simple Forest	3	100	0.816	0.816	0.816
Best Forest	32	100	0.816	0.816	0.816

Problem 3: Analysis

Short Answer 3a

I think the runtime would be O(D). Since the search is making a prediction on a single test feature vector, the runtime would be only based on the depth of the tree. The decision at each node is constant runtime. It doesn't matter how many times it happens with nodes. You are only looking at one set of input features so it is also constant time.