# CS 165B – Machine Learning, Spring 2023

## Assignment #2
## Due Thursday, May 4 by 11:59pm

---

**Notes:**

- *This assignment is to be done individually. You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.*
- Be sure to re-read the "Policy on Academic Integrity" on the course syllabus.
- Be aware of the late policy in the course syllabus – i.e., *late submissions will be accepted but 20% deduction per day.*
- Justify every answer you give – **show the work** that achieves the answer or **explain** your response.
- Any updates or corrections will be posted on Piazza, so check there occasionally.
- **All assignments must be clear and legible.** It is recommended to type the solutions on this Microsoft Word file directly and save it as PDF. If you'll be submitting a handwritten assignment, please ensure that it's readable and neat. If your writing is not easily readable, your solution is not easy to follow on the page, or your PDF is not of very high quality, your assignment will not be graded. DO NOT submit picture of your written work. (If you must scan your written work in, use a high-quality scanner. Plan in advance.)
- To turn in your assignment:
  - Use entry code **2K4D2G** to register as a student for **CS 165B Spring 2023 Machine Learning** on Gradescope.
  - Submit a PDF version to Gradescope

---

**Problem #1 [8 points]**
A ranking classifier ranks 20 training examples $\{x_i\}$, from highest to lowest rank, in the following order:

Highest Lowest

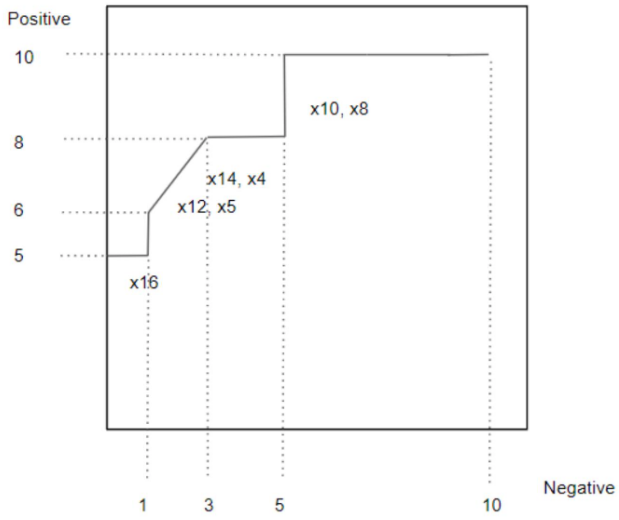
$\mathbf{x_2}, \mathbf{x_3}, \mathbf{x_6}, \mathbf{x_1}, \mathbf{x_7}, x_{16}, \mathbf{x_9}, x_{12}, \mathbf{x_4}, x_{11}, \mathbf{x_{10}}, \mathbf{x_8}, x_{17}, x_{20}, x_{15}, x_{18}, x_{19}, \mathbf{x_5}, x_{14}, x_{13}$

Examples $x_4$ and $x_{14}$, $x_5$ and $x_{12}$, $x_{11}$ and $x_{13}$ all have the same rank.
Examples $x_1$ through $x_{10}$ are in the positive class (which should be ranked higher); examples $x_{11}$ through $x_{20}$ are in the negative class (which should be ranked lower).

(a) How many ranking errors are there?

(b) What is the ranking error rate?
(c) What is the ranking accuracy?
(d) Draw the coverage curve for the ranking classifier on this dataset.

Answer:
(a) 15
(b) 15/100 = 0.15
(c) 1-0.15 = 0.850




(d)

**Problem #2 [6 points]**
Consider a learning problem where real numbers are used as instances, and intervals over the real numbers are used as hypotheses. Each hypothesis in this scenario takes the form of $a < x < b$, where x represents the instance, and a and b are real constants. For instance, $4.5 < x < 6.1$ is a hypothesis that categorizes instances between 4.5 and 6.1 as positive, while others are considered negative. Provide an informal explanation for why there cannot be a hypothesis that is the least general generalization. Additionally, propose a slight alteration to the hypothesis representation that would allow for a least general generalization.
There cannot be a least general generalization hypothesis for any set of positive training examples in this concept learning problem because the hypothesis space consists of intervals over the reals, and the number of possible intervals is infinite. Therefore, there will always be some gaps between any two intervals, and it is always possible to find a more specific hypothesis that covers a smaller range of values.

To address this issue, one slight modification to the hypothesis representation is to use closed intervals instead of open intervals. A closed interval includes its endpoints, so a hypothesis of the form $a <= x <= b$ is a closed interval that includes both a and b as possible values. With closed intervals, it is possible to find a maximally specific consistent hypothesis for any set of positive training

examples. For example, if the positive training examples are 5, 6, and 7, the maximally specific consistent hypothesis is 5 <= x <= 7, which covers exactly the range of values of the positive examples.

(ML Mitchell 2.7)

**Problem #3 [16 points]**

Build the best Classification and Regression Tree (CART) from the following training samples that use three features, each of which has two values. You should use gini gain (i.e., use gini index as impurity function) as criterion when choosing features:

Buy: (rating=high, market=bull, growth=false)
Buy: (rating=low, market=bull, growth=false)
Buy: (rating=high, market=bear, growth=true)
Buy: (rating=low, market=bull, growth=true)
Sell: (rating=low, market=bear, growth=true)
Sell: (rating=low, market=bear, growth=false)
Sell: (rating=high, market=bull, growth=true)

Answer:

Step1: $Root\ node\ gini = 2\times\frac{4}{7}\times\frac{3}{7} = \frac{24}{49}$

If split by rating: $gini(rating = high) = 2\times\frac{2}{3}\times\frac{1}{3} = \frac{4}{9}$

$gini(rating = low) = 2\times\frac{1}{2}\times\frac{1}{2} = \frac{1}{2}$

$gini\ gain = \frac{24}{49} - \frac{3}{7}\times\frac{4}{9} - \frac{4}{7}\times\frac{1}{2} = 0.0136$

If split by market: $gini(market = bull) = 2\times\frac{3}{4}\times\frac{1}{4} = \frac{3}{8}$

$gini(market = bear) = 2\times\frac{1}{3}\times\frac{2}{3} = \frac{4}{9}$

$gini\ gain = \frac{24}{49} - \frac{4}{7}\times\frac{3}{8} - \frac{3}{7}\times\frac{4}{9} = 0.0850$

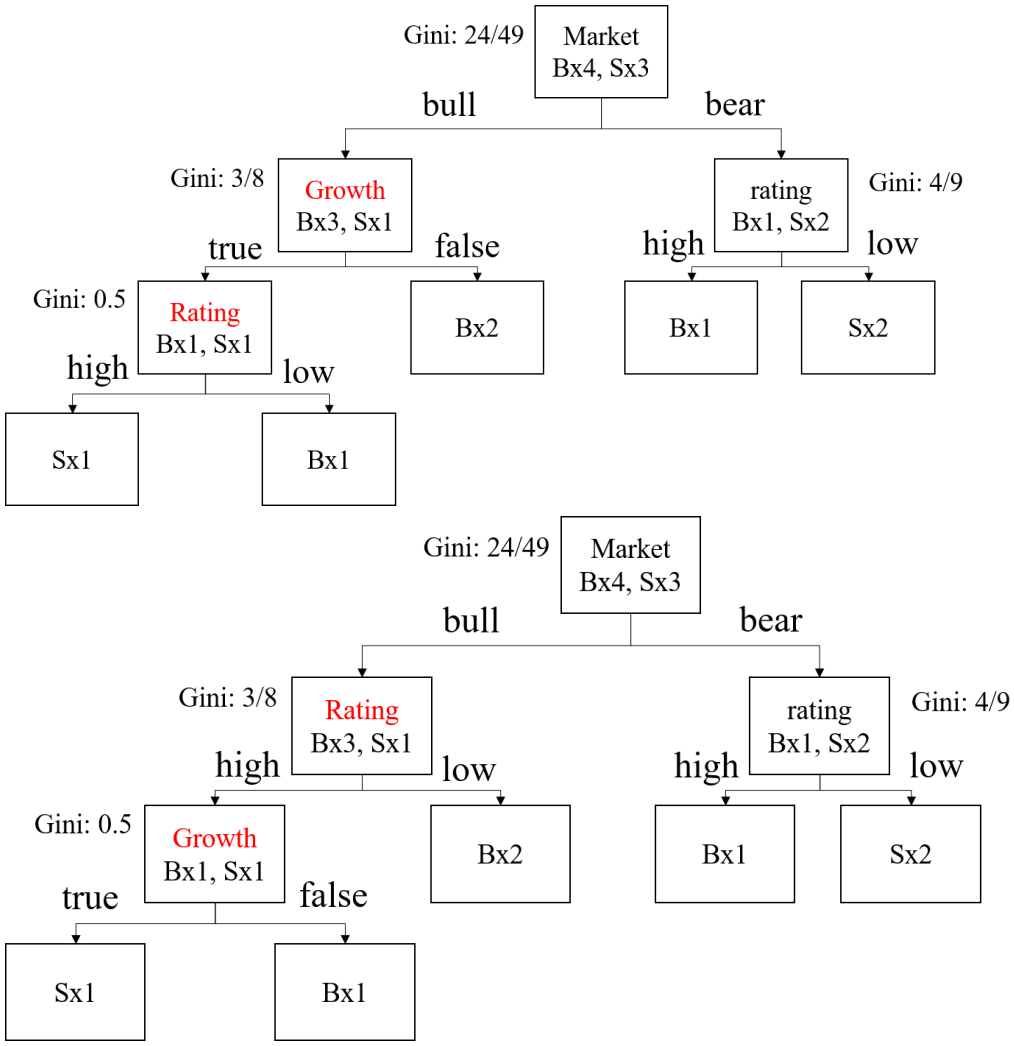
If split by growth: $gini(growth = true) = 2\times\frac{1}{2}\times\frac{1}{2} = \frac{1}{2}$

$gini(growth = false) = 2\times\frac{2}{3}\times\frac{1}{3} = \frac{4}{9}$

$gini\ gain = \frac{24}{49} - \frac{4}{7}\times\frac{1}{2} - \frac{3}{7}\times\frac{4}{9} = 0.0136$

Therefore, the first split must be based on market.

Repeat this process and we could build the following decision tree.




**Problem #4 [10 points]**
In a 5-class classification problem, 30 training examples are supplied that have the following class labels:

| | | | | | |
|---|---|---|---|---|---|
| 3 | 1 | 5 | 4 | 3 | 5 |
| 2 | 5 | 1 | 3 | 5 | 3 |
| 5 | 3 | 2 | 5 | 5 | 2 |
| 2 | 4 | 5 | 3 | 2 | 1 |
| 3 | 5 | 3 | 1 | 5 | 2 |

From the training data, we wish to estimate the probabilities of each class. Empirically estimate each class probability using (a) relative frequency, (b) Laplace correction, (c)

m-estimate with m=10 and an even distribution of the pseudocounts, and (d) m-estimate with m=20 and an even distribution of the pseudocounts.

Plot these empirical probabilities, with class number on the x-axis and estimated probability on the y-axis – i.e., four plots (one for each approach), each of which consists of five connected points (the estimated class probabilities). Plot all four distributions on a single graph and label them.

Describe the trend as we go from (a) to (b) to (c) to (d) – that is, what does increasing the number of pseudocounts do (in general) to the probability distribution?

Answer:

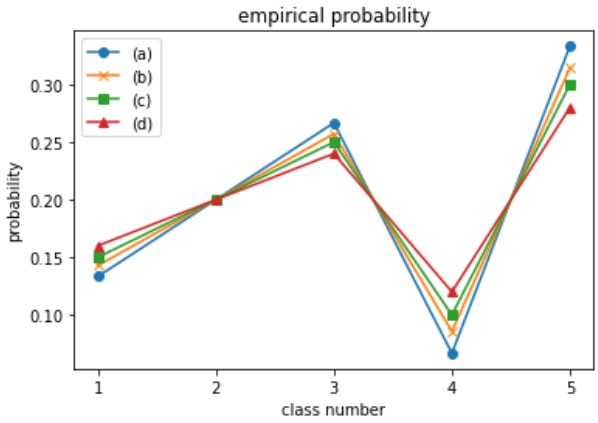
(a) P(class=1) = 4/30 = 2/15
P(class=2) = 6/30 = 1/5
P(class=3) = 8/30
P(class=4) = 2/30
P(class=5) = 10/30

(b) P(class=1) = (4+1)/35 = 5/35
P(class=2) = 7/35
P(class=3) = 9/35
P(class=4) = 3/35
P(class=5) = 11/35

(c) The prior probability is 0.2
P(class=1) = (4+0.2*10)/(30+10) = 6/40
P(class=2) = 8/40
P(class=3) = 10/40
P(class=4) = 4/40
P(class=5) = 12/40

(d) P(class=1) = (4+0.2*20)/(30+20) = 8/50
P(class=2) = 10/50
P(class=3) = 12/50
P(class=4) = 6/50
P(class=5) = 14/50

(e) Increasing the number of pseudocounts will make the line more smooth, i.e., more close to the average. Estimation that is higher than the averaged will be become lower, while estimation lower than the averaged will become larger.




**Problem #5 [10 points]**

(a) Prove that for any dataset without conflict samples (samples with the same features but different labels), there exists a decision tree that is consistent with the dataset (the training error is zero). You can assume the tree uses entropy as impurity function and the features are categorical.

(b) Explain the disadvantage of using "minimal training loss" as the criterion for choosing BestSplit (i.e., choose the feature that decrease the training loss most for each node).

(c) What are some ways to avoid the disadvantage in (b)?

Answer:

(a) Suppose there is a leaf node that is not pure. Then there are two possible cases for that node. One is the features of all samples are the same. Another is that the features of all samples are not the same. For the first case, according to the assumption in the problem that there are no conflict samples, all the samples shall have the same labels, thus contradicts with that the node is not pure. For the second case, these features are not the same, then there exists at least one feature that can further split this node into different branches. Since this node is not pure and can further be split, the decision tree algorithm will further split this node into different branches, which contradicts with that this node is a leaf node. Through contradiction, we show there does not exist a leaf node that is not pure, thus the decision tree is consistent.

(b) According to (a), there exists a decision tree with training error 0. Choosing training error to split the node can easily incur over-fitting. We may get a decision tree that are perfectly fit the training data, but performs poorly on the testing data. Another disadvantage is that using training loss will prefer some features with more values, because they will create more branches and thus more likely to decrease the training loss.

Other reasonable explanation is acceptable.

(c) Overfitting: pre-pruning methods such as max tree depth, min num of samples in each node, threshold for information gain; post-pruning methods such as reduced error pruning

Keep using the same feature: randomly select features at each node and build an ensemble (Bagging/Boosting); avoid using a previous feature after it has been used for certain times.

**Problem #6 [10 points]**

For the following set of points: {(-2 , -1) , (1 , 1) , (3 , 2)}

a) Find the least square regression line for the given data points. Show all work.

b) Plot the given points and the regression line in the same rectangular system of axes.

1. a) Let us organize the data in a table.

| x | y | x y | $x^2$ |
|---|---|---|---|
| -2 | -1 | 2 | 4 |
| 1 | 1 | 1 | 1 |
| 3 | 2 | 6 | 9 |
| $\Sigma x = 2$ | $\Sigma y = 2$ | $\Sigma xy = 9$ | $\Sigma x^2 = 14$ |

We now use the above formula to calculate a and b as follows

$a = (n\Sigma x\, y - \Sigma x \Sigma y) / (n\Sigma x^2 - (\Sigma x)^2) = (3*9 - 2*2) / (3*14 - 2^2) = 23/38$

$b = (1/n)(\Sigma y - a\, \Sigma x) = (1/3)(2 - (23/38)*2) = 5/19$

b) We now graph the regression line given by y = a x + b and the given points.

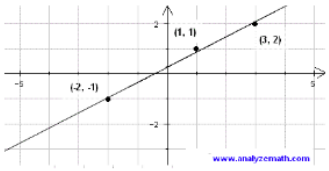



Figure 3. Graph of linear regression in problem 1.

(https://www.analyzemath.com/statistics/linear_regression.html)

**Problem #7 [10 points]**

We have collected the following dataset for predicting heart attacks:

HEART ATTACK: (CHEST PAIN=yes,MALE=yes,SMOKES=no,EXERCISES=yes)
HEART ATTACK: (CHEST PAIN=yes,MALE=yes,SMOKES=yes,EXERCISES=no)
HEART ATTACK: (CHEST PAIN=no,MALE=no,SMOKES=yes,EXERCISES=no)
HEALTHY: (CHEST PAIN=no,MALE=yes,SMOKES=no,EXERCISES=yes)
HEART ATTACK: (CHEST PAIN=yes,MALE=no,SMOKES=yes,EXERCISES=yes)
HEALTHY: (CHEST PAIN=no,MALE=yes,SMOKES=yes,EXERCISES=yes)

a) Build a decision tree from this data by minimizing entropy.

1. (15 points) Use information theory to construct a minimal decision tree that predicts whether or not a patient is likely to have a heart attack. **SHOW EACH STEP OF THE COMPUTATION.**

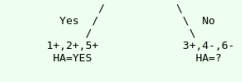
```
       x      1  1/4  2/3  3/4  1/3  1/2
-------------------------------------
 log_2(x)     0   -2 -0.6 -0.4 -1.6  -1
```

- **Selecting the attribute for the root node:**

Heart Attack  YES  NO

  - **Chest Pain:**
    - Yes: (3/6)*[ -(3/3)*log_2(3/3) - (0/3)*log_2(0/3)]
    - No: (3/6)*[ -(1/3)*log_2(1/3) - (2/3)*log_2(2/3)]
    - TOTAL: = 0 + 0.47 = 0.47
  - **Male:**
    - Yes: (4/6)*[ -(2/4)*log_2(2/4) - (2/4)*log_2(2/4)]
    - No: (2/6)*[ -(2/2)*log_2(2/2) - (0/2)*log_2(0/2)]
    - TOTAL: = 0.667 + 0 = 0.667
  - **Smokes:**
    - Yes: (4/6)*[ -(3/4)*log_2(3/4) - (1/4)*log_2(1/4)]
    - No: (2/6)*[ -(1/2)*log_2(1/2) - (1/2)*log_2(1/2)]
    - TOTAL: = 0.53 + 0.33 = 0.86
  - **Exercises:**
    - Yes: (4/6)*[ -(2/4)*log_2(2/4) - (2/4)*log_2(2/4)]
    - No: (2/6)*[ -(2/2)*log_2(2/2) - (0/2)*log_2(0/2)]
    - TOTAL: = 0.667 + 0 = 0.667

Since **Chest Pain** is the attribute with the lowest entropy, it is selected as the root node:




Only the branch corresponding to Chest Pain = No needs further processing.

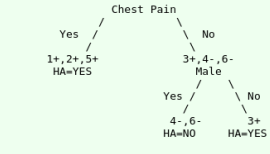
- **Selecting the attribute to split the branch Chest Pain = No:**

Only Examples 3,4,6 need to be taken into consideration.

YES  NO

  - **Male:**
  - Yes: (2/3)*[ -(0/2)*log_2(0/2) - (2/2)*log_2(2/2)]
  - No: (1/3)*[ -(1/1)*log_2(1/1) - (0/1)*log_2(0/1)]
  - TOTAL: = 0 + 0 = 0

Since the minimum possible entropy is 0 and **Male** has that minimum possible entropy, it is selected as the best attribute to split the branch Chest Pain=No. Note that we don't even have to calculate the entropy of other two attributes, i.e., **Smokes,Exercises**, since they cannot possibly be better than 0. BTW, note that the entropy of Exercises is also 0 over examples 3, 4, 6 and so it could also have been used to split this node.




Each branch ends in a homogeneous set of examples so the construction of the decision tree ends here.

2. (5 points) Translate your decision tree into a collection of decision rules.

b) Explain what information gain is, and why it is equivalent to minimizing entropy. Finally, explain what impurity measures are and how this connects to minimizing entropy.

Information gain is a measure used in decision tree algorithms to evaluate the potential of an attribute to split the data into homogeneous subsets based on the values of that attribute. In other words, it measures the reduction in uncertainty about the classification of a data point that is obtained by knowing the value of a particular attribute.
Entropy is a measure of impurity or randomness in a set of data. It is used in decision tree algorithms to determine the best attribute to split the data at each node. The idea is to choose the attribute that maximizes the information gain, which is equivalent to minimizing the entropy.
The entropy of a set S is defined as:
$H(S) = - \text{sum}(p(i) * \log2(p(i)))$
where p(i) is the proportion of elements in S that belong to class i.
When we split a set S based on the values of an attribute A, we obtain a partition {S1, S2, ..., Sm} of S. The information gain of A with respect to S is defined as:
$IG(S, A) = H(S) - \text{sum}(|S(i)|/|S| * H(S(i)))$
where |S(i)| is the number of elements in the subset S(i) and |S| is the total number of elements in S.
The information gain is a measure of how much the entropy of S is reduced after the split. It is equivalent to minimizing the entropy because choosing the attribute with the highest information gain is equivalent to choosing the attribute that produces the smallest possible entropy of the resulting subsets.
Impurity measures are used in decision tree algorithms to evaluate the quality of a split. They are usually defined as a function of the class proportions in the subsets produced by the split. The most commonly used impurity measures are Gini impurity, entropy, and misclassification error. In practice, entropy is often used as an impurity measure because it has desirable mathematical properties and is closely related to information gain. Specifically, maximizing information gain is equivalent to minimizing the entropy of the resulting subsets.