Discussion Session Problems 6

02/19/2026

1. You have a large combination of categorical features. Your friend suggests that you should give each possible combination of features a number and use the binary representation as a vector instead of the one hot encoding in order to reduce the number of features, and claims that it will not affect the performance of the tree.

   (a) You should disagree because this will not allow the decision trees to get the same accuracy as a one hot encoding.

   (b) You should disagree because this will make the decision tree unnecessarily deep because you have to go through all combinations.

   (c) You should agree because if you use one hot encoding it would take too many steps to check each feature as it's own node in the decision tree, making your tree significantly deeper.

   (d) You should agree because it would take too much time to calculate the entropy for each value in the one hot encoding.

   B.

2. Your friends trained a random forest algorithm and are always ending up with the same results from each tree for every new input. You want to check that the problem is not that easy for their algorithm. Which steps will you check that they did correctly first?

   Check that resampling with replacement is done correctly and check that sampling features is done correctly at each node.

3. True or false? Decision tree algorithms are more negatively affected than neural nets if the data used for training is dissimilar to what is used in testing.

   True. Decision trees are more sensitive to the training data, and they are not as good at extrapolation, and incapable of transfer learning.

4. Your friends are training a random forest algorithm to diagnose some medical conditions based on blood work. They are complaining that their random forest algorithm is very bad at recognizing some rare cases that occur to some patients. What do you recommend?

   Using boosting.

5. True or false? When training a decision tree algorithm similarly to training a neural net it is good practice to see if you can overfit on a small portion of the dataset first before you start training on the entire dataset to save time spent on training.

   False, for most cases you can train a decision tree algorithm on the entire dataset in much less time than a neural network.

6. Which of the following is the best feature to pick at a node in a decision tree algorithm?

   (a) Feature 1: splits the dataset into two equal parts. The first part is all false examples and the second part is half true labels and half false labels. Entropy before: 0.81, entropy after: 0.5

(b) Feature 2: splits the dataset in a 70/30 split. The 70% split is 80% true labels and the 30% split is 75% false labels. Entropy before = 0.95, entropy after = 0.7*0.72 + 0.3*0.81 = 0.747

(c) Feature 3: Splits the dataset in a 90/10 split. The 90% split is 70% true labels and the 10% split is 100% false labels. Entropy before = 0.95, entropy after = 0.9*0.88 + 0.1*0 = 0.792

A.

7. Why is feature scaling (e.g., standardization or normalization) generally not necessary for decision tree-based models, unlike for models like logistic regression or support vector machines?

   Feature scaling is generally not necessary for decision tree-based models because decision trees split on feature values based on thresholds rather than distances or gradients. Unlike models such as logistic regression or support vector machines (SVMs), which rely on optimizing a loss function with respect to distances in feature space, decision trees only need to compare feature values to a threshold at each node. Since these comparisons are independent of the magnitude of the feature values, normalization or standardization does not impact the model's performance.

8. Consider three impurity measures used when choosing splits in a decision tree:

   - **Entropy:** $H(p) = -p \log_2 p - (1-p) \log_2 (1-p)$
   - **Gini Impurity:** $G(p) = 2p(1-p)$
   - **Misclassification Error:** $E(p) = 1 - \max(p, 1-p)$

   Here, p is the fraction of examples in a node that belong to class 1. Which impurity measure(s) are largest when the classes are evenly split (e.g., $p = 0.5$ in binary classification)?

   (a) Entropy

   (b) Gini Impurity

   (c) Misclassification Error

   (d) All of the above

   D. All three impurity measures are maximized when the classes are evenly split ($p = 0.5$), because this represents the most uncertain or impure node.

9. True or false? Entropy and Gini impurity will always select the exact same split in a decision tree.

   False. They often choose similar splits but can differ because they measure impurity differently.

10. You are training a decision tree and find that a feature creates many small branches, each containing very few samples. What is a likely problem?

    (a) Underfitting

    (b) Overfitting

    (c) Data leakage

    (d) Feature scaling issue

    B. The tree is likely overfitting by memorizing noise in small partitions.

11. Which of the following introduces randomness into a random forest?

    (a) Sampling training examples with replacement (bagging)

    (b) Sampling a subset of features at each split

    (c) Using different random seeds for trees

    (d) All of the above

D. All of these contribute to randomness and tree diversity.

12. Why does a random forest often perform better than a single decision tree?

    Because averaging predictions across many decorrelated trees reduces variance and prevents overfitting.

13. True or false? Increasing the number of trees in a random forest will always increase overfitting.

    False. Increasing trees usually reduces variance and improves performance until computation becomes the main limitation.

14. In boosting methods such as XGBoost, what happens to examples that are misclassified?

    (a) They are removed from the dataset
    (b) Their weights are increased so the next model focuses on them
    (c) They are ignored
    (d) They are randomly relabeled

    B. Boosting increases their importance so later models focus on hard examples.

15. What is the main difference between bagging (used in random forests) and boosting (used in XGBoost)?

    Bagging trains trees independently on resampled data and averages them, while boosting trains trees sequentially so each new tree corrects errors made by previous ones.

16. True or false? XGBoost can handle missing values automatically during training.

    True. XGBoost learns a default direction for missing values at each split.

17. Your XGBoost model is overfitting. Which hyperparameters might you tune?

    Decrease max depth, increase regularization terms, reduce learning rate, increase subsampling, or use early stopping.

18. Show that entropy is maximized when the classes are evenly split.

    For binary classification, entropy is

$$H(p) = -p \log_2 p - (1-p) \log_2 (1-p),$$

    where p is the probability of class 1.

    (a) Take the derivative of $H(p)$ with respect to p.
    (b) Solve for the critical point.
    (c) Verify that this point is a maximum.

$$\frac{dH}{dp} = -\log_2 p + \log_2 (1-p) = \log_2 \frac{1-p}{p}.$$

    Setting derivative to zero gives

$$2^{\log_2 \frac{1-p}{p}} = 2^0 \Rightarrow \frac{1-p}{p} = 1 \Rightarrow p = 0.5.$$

    Second derivative:

$$\frac{d^2 H}{dp^2} = -\frac{1}{p} - \frac{1}{(1-p)} = -\left(\frac{1}{p} + \frac{1}{1-p}\right).$$

    Since this is negative for all $0 < p < 1$, in particular at $p = 0.5$, the critical point is a maximum. Therefore entropy is maximized when classes are evenly split.