Circle all correct statements. (https://datascience.stackexchange.com/questions/10713/does-random-forest-re-use-features-at-each-node-when-generating-a-decision-tree)

1 point



~~When building a classification tree, each feature can only be used once.~~ ( same feature can be used in a different branch of the tree at a different level)



~~Individual trees of a bagged decision forest are less correlated in comparison to individual trees of random forest.~~



Boosting is a general approach that can be applied to many statistical learning methods for regression or classification.



When building a classification tree, we can reuse a feature to split a node.



Tree models are not affected by multicollinearity in features.

Question 2

What can be the maximum depth of a (binary) decision tree built on a training data with n samples and p features? Here "binary" means that we only divide each node into two children nodes. Further, assume the n samples are all unique (i.e., the n-by-p data matrix has no duplicate rows).

1 point



n



n-p-1



p-1



n/2-p



n-1



n/2

3.

Question 3

Given 1000 observations, minimum number of observations required to split a node equals 200 and minimum leaf size equals 300, then what could be the maximum depth of a decision tree? (The depth for a tree with just the root node is 0, and the depth for a tree with one split is 1.)

1 point



2



5



4



1



3

4.

Question 4

In random forest, what is randomly selected?

1 point



For each node, we randomly select a set of features as the split candidates from the full set of p features.



For each tree in the forest, we randomly select a set of features as the split candidates from the full set of p features.



For each node, we randomly select a feature from the full set of p features to split.



We randomly select the number of trees in the forest.

5.

Question 5

Which of the following statements are correct about gbm parameters?

1 point



~~"shrinkage" can be any non-negative number.~~



"n.trees" should be inversely proportional to "shrinkage", that is, if decreasing "shrinkage", we should increase "n.trees" when tuning the parameters.



"bag.fraction" should be between 0 and 1.



"n.trees" can be equal to 1.

6.

Question 6

Suppose we are applying adaboosting and randomforest on a classification problem. Below, "error" refers to the ordinary misclassification rate (number of misclassified samples divided by the total sample size) using all trees. When the number of trees increases, (Refer to lecture 10d ISLR Videos)

1 point



The test error of randomforest decreases.

~~~~

~~The test error of adaboosting decreases.~~



~~The training error of adaboosting decreases.~~



The training error of randomforest decreases.



None of the above.

7.

Question 7 (Answered in the sample RCode\_W12\_old\_boost.Rmd file I checked in)

Consider a simple binary classification problem with one-dimensional feature *x* and three training samples:

* sample 1: *x*=1, *y*=−1
* sample 2: *x*=3, *y*=+1
* sample 3: *x*=5, *y*=−1

Next we study how AdaBoost algorithm performs on this data set. At iteration 1, suppose we pick a classifier h\_1(x)*h*1​(*x*) that returns -1−1, if x <= 2*x*<=2 and +1+1, otherwise. Then which point's weight or points' weights will be increased?

1 point



None of the above



sample 2



sample 3



sample 1

8.

Question 8

Continue with the previous problem on AdaBoost. Write down the weight that is assigned to each data point after the first iteration of the AdaBoost algorithm.

* weight on sample 1 = [ans1]
* weight on sample 2 = [ans2]
* weight on sample 3 = [ans3]

**[ans1]=\_\_\_** **0.25\_\_\_\_\_\_\_\_**

1 point



9.

Question 9

Continue with the previous problem on AdaBoost. Write down the weight that is assigned to each data point after the first iteration of the AdaBoost algorithm.

* weight on sample 1 = [ans1]
* weight on sample 2 = [ans2]
* weight on sample 3 = [ans3]

**[ans2]=\_\_\_\_** **0.25\_\_\_\_\_\_\_**

1 point



10.

Question 10

Continue with the previous problem on AdaBoost. Write down the weight that is assigned to each data point after the first iteration of the AdaBoost algorithm.

* weight on sample 1 = [ans1]
* weight on sample 2 = [ans2]
* weight on sample 3 = [ans3]

**[ans3]=\_\_\_0.50\_\_\_\_\_\_\_\_**

1 point



