BIOST 2155: Assignment 5

YOUR NAME HERE (YOUR PITT EMAIL HERE)

Released: 2026-11-20. **Due by**: 2026-12-04, 11:59 PM

Instructions

- This file is a template. Everything you need to run is already written. **Do not change the code.** Knit the file once, read each result, and write your interpretation in the space under it.
- This assignment covers **Decision Trees** (Lecture 10: CART, bagging, random forests, boosting) and **Model Averaging** (Lecture 11). It uses one data set: `Wisconsin Breast Cancer` (`assign3_data.csv`, continued from Assignments 3-4).
- There are two theory problems (20 points each) and one application with ten results to interpret (6 points each). Total: 100 points.
- In the application, we give you the code and the output. You write the interpretation. A good answer is 3-5 sentences. It quotes numbers from the output and says what they mean for the model or for a clinical user.

Theory (40 points)

T1. The Gini index is maximized at maximal impurity (20 points)

In decision trees for classification, we need an impurity function to choose the best split. The Gini index, $G = 1 - \sum_{k=1}^{K} p_k^2 = \sum_{k=1}^{K} p_k(1 - p_k)$, is one common choice. Here p_k is the proportion of class k in a node.

(a) (5 points) Take $K = 2$ with $p_1 = p$ and $p_2 = 1 - p$. Write G as a function of p. Show that it is maximized at $p = 1/2$, and give the maximum value.

(b) (10 points) For general $K \geq 2$, show that G takes its maximum value when $p_1 = p_2 = \ldots = p_K = 1/K$. Give the maximum value.

Hint: you have the constraint $\sum_{k=1}^{K} p_k = 1$. You may use the Lagrange multiplier method (calculus) or the Cauchy-Schwarz inequality (algebra).

(c) (5 points) What is the value of G for a pure node (all observations in one class)? In two or three sentences, explain what (a) and (b) tell us about the kind of node a tree tries to create when it chooses a split.

Show your working. You may upload handwritten answers.

T2. Why averaging helps: ensemble risk equals average risk minus diversity (20 points)

Fix one input x with true outcome y. Suppose we have M models that give predictions $\hat{y}_1, \ldots, \hat{y}_M$ for y. The unweighted ensemble prediction is $\hat{y}_E = M^{-1} \sum_{k=1}^{M} \hat{y}_k$. Use squared error loss. Define the *diversity* of the ensemble as $V = M^{-1} \sum_{k=1}^{M} (\hat{y}_k - \hat{y}_E)^2$.

(a) (10 points) Prove that

$$(y - \hat{y}_E)^2 = \frac{1}{M} \sum_{k=1}^{M} (y - \hat{y}_k)^2 - V.$$

Hint: write $y - \hat{y}_k = (y - \hat{y}_E) + (\hat{y}_E - \hat{y}_k)$, square it, and average over k. Look at what happens to the cross term.

(b) (4 points) Use (a) to show that the ensemble never has larger squared error than the average squared error of its members. When is it equal?

(c) (3 points) Does (a) show that the ensemble beats the *best* single member? Explain in two or three sentences. A small numerical example is enough.

(d) (3 points) A colleague proposes to add a very inaccurate model to the ensemble "because it is different from the others and will raise diversity". Use (a) to explain why this often does not help.

Show your working. You may upload handwritten answers.

Application (60 points)

Dataset: `Wisconsin Breast Cancer`, saved as `assign3_data.csv` (same data as Assignments 3-4).

We return one last time to the breast cancer data. Having tried SVMs (Assignment 3) and PCA with logistic regression (Assignment 4), we now fit a single tree, two tree ensembles (random forest and boosting), and a logistic regression. We then average them. The code is written for you, and it uses `set.seed(42)`. Your job is to read the results and interpret them. In every question, `M` (malignant) is the positive class.

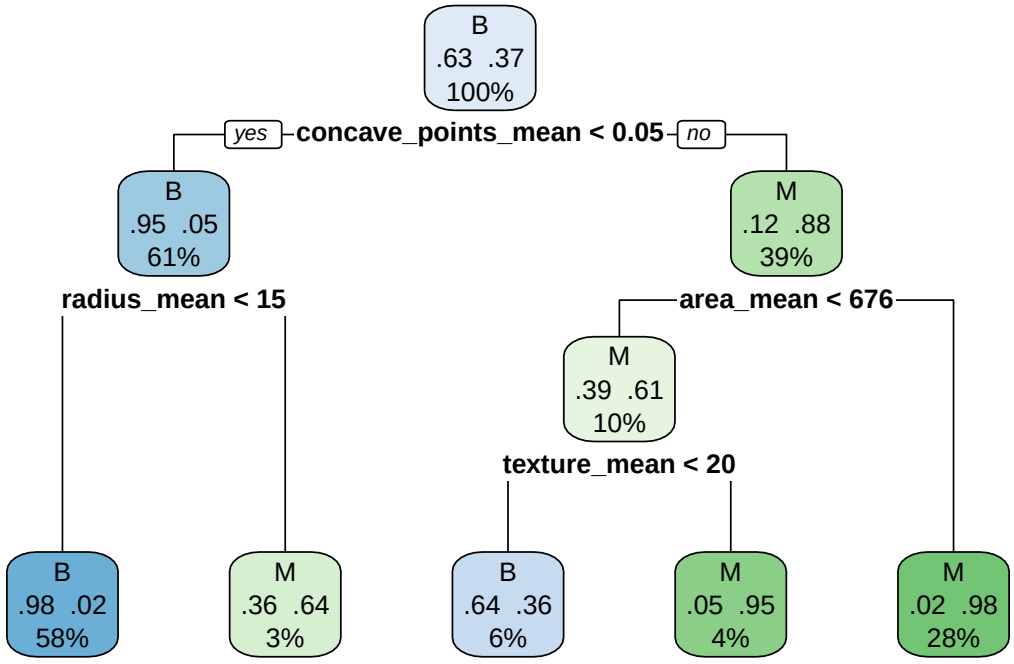
```
rm(list = ls())
pacman::p_load(tidyverse, caret, rpart, rpart.plot, randomForest,
    gbm, pROC)
set.seed(42)
data <- read_csv("assign3_data.csv", show_col_types = FALSE) |>
    dplyr::select(-1) |>
    drop_na()
data$diagnosis <- factor(data$diagnosis, levels = c("B",
    "M"))

train_index <- createDataPartition(data$diagnosis, p = 0.75,
    list = FALSE)
train <- data[train_index, ]
test <- data[-train_index, ]
y_test <- as.numeric(test$diagnosis == "M")  # 1 = malignant
```

Part 1. A single tree

Result 1 (6 points). A classification tree, grown with the default settings.

```
set.seed(42)  # rpart runs a random 10-fold cross-validation internally
tree_full <- rpart(diagnosis ~ ., data = train, method = "class")
rpart.plot(tree_full, extra = 104)
```



Interpret. Read the root split. Which feature and threshold does it use? What share of the training data goes down each branch? Does this feature make clinical sense for telling malignant from benign tumors?

Your interpretation:

Result 2 (6 points). Cost-complexity pruning. The table shows the cross-validated error (`xerror`) for each complexity parameter (`CP`) value. The pruned tree uses the `CP` with the smallest `xerror`.

```
tree_full$cptable
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.78616352      0 1.0000000 1.0000000 0.06282824
## 2 0.02515723      1 0.2138365 0.2327044 0.03656134
## 3 0.02201258      2 0.1886792 0.2767296 0.03951069
## 4 0.01000000      4 0.1446541 0.2452830 0.03744011
```

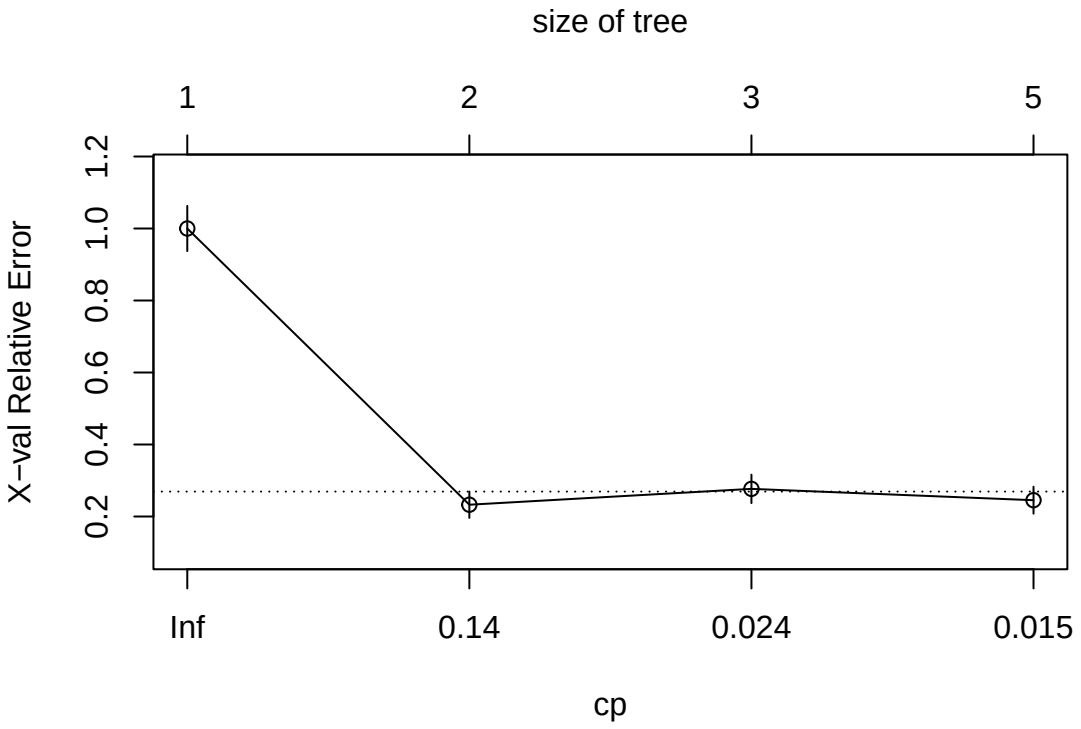
```
best_cp <- tree_full$cptable[which.min(tree_full$cptable[,
    "xerror"]), "CP"]
tree_pruned <- prune(tree_full, cp = best_cp)
cat("Leaves in full tree:  ", sum(tree_full$frame$var ==
    "<leaf>"), "\n")
```

```
## Leaves in full tree:   5
```

```
cat("Leaves in pruned tree:", sum(tree_pruned$frame$var ==
    "<leaf>"), "\n")
```

```
## Leaves in pruned tree: 2
```

```
plotcp(tree_full)
```



Interpret. Which `CP` did pruning pick, and how many leaves does the pruned tree have? Compare `rel error` (training) with `xerror` (cross-validated) as the tree grows. What do you see, and what does it say about bias and variance? Does the extra training accuracy of the larger trees carry over to cross-validated accuracy? (`xstd` is the standard error of `xerror`.)

Your interpretation:

Result 3 (6 points). The pruned tree on the test set.

```
p_tree <- predict(tree_pruned, test, type = "prob")[, "M"]
pred_tree <- factor(ifelse(p_tree > 0.5, "M", "B"), levels = c("B",
    "M"))
confusionMatrix(pred_tree, test$diagnosis, positive = "M")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 79  5
##          M 10 48
##
##                Accuracy : 0.8944
##                  95% CI : (0.8318, 0.9397)
##     No Information Rate : 0.6268
##     P-Value [Acc > NIR] : 5.498e-13
##
##                   Kappa : 0.7784
##
##  Mcnemar's Test P-Value : 0.3017
##
##             Sensitivity : 0.9057
##             Specificity : 0.8876
##          Pos Pred Value : 0.8276
##          Neg Pred Value : 0.9405
##              Prevalence : 0.3732
##          Detection Rate : 0.3380
##    Detection Prevalence : 0.4085
##       Balanced Accuracy : 0.8967
##
##        'Positive' Class : M
##
```

Interpret. Report accuracy and sensitivity. How many malignant tumors were missed? For a screening setting, is this a good enough model? Why or why not? (Recall the cost of a false negative from Assignment 3.)

Your interpretation:

Part 2. Bagging, random forests, and boosting

Result 4 (6 points). A random forest with 500 trees. The out-of-bag (OOB) error uses, for each tree, the observations that tree did not see.

```
set.seed(42)
rf <- randomForest(diagnosis ~ ., data = train, ntree = 500)
rf
```

```
##
## Call:
##  randomForest(formula = diagnosis ~ ., data = train, ntree = 500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
```

```
##         OOB estimate of  error rate: 5.39%
## Confusion matrix:
##     B   M class.error
## B 256  12  0.04477612
## M  11 148  0.06918239
```

```
p_rf <- predict(rf, test, type = "prob")[, "M"]
table(Predicted = predict(rf, test), Truth = test$diagnosis)
```

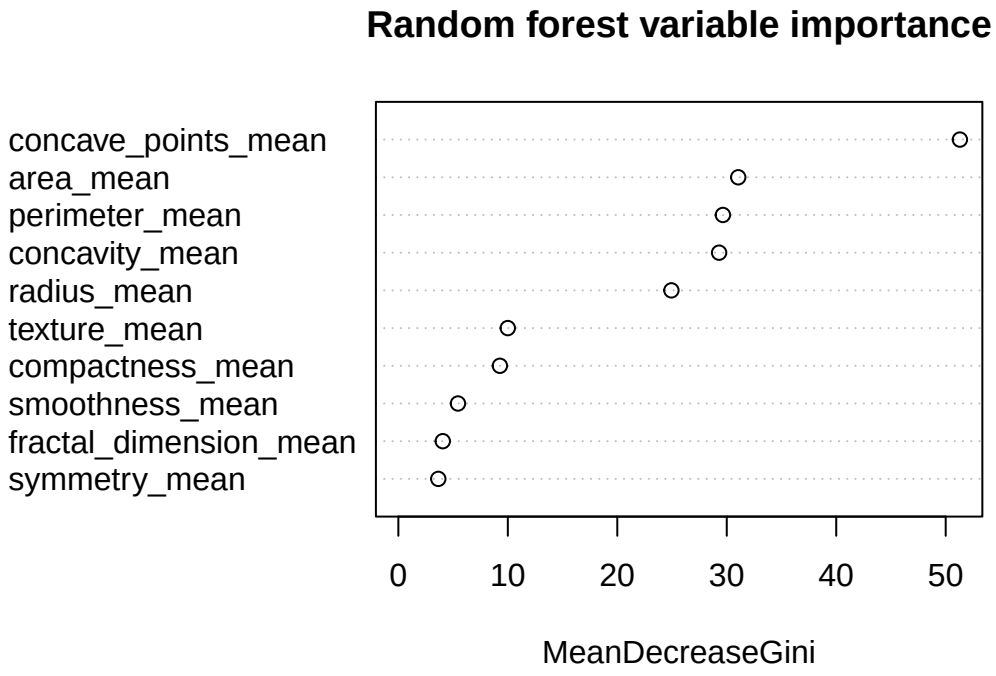
```
##          Truth
## Predicted  B  M
##         B 84  7
##         M  5 46
```

Interpret. Report the OOB error and the test error. Are they close? Why can the OOB error serve as an estimate of test error without a separate validation set? Compare the test error to the single pruned tree in Result 3.

Your interpretation:

Result 5 (6 points). Variable importance (mean decrease in Gini) from the random forest.

```
varImpPlot(rf, main = "Random forest variable importance")
```



```
importance(rf)[order(-importance(rf)), , drop = FALSE]
```

```
##                         MeanDecreaseGini
## concave_points_mean            51.297539
## area_mean                      31.048538
## perimeter_mean                 29.645862
## concavity_mean                 29.300037
## radius_mean                    24.943582
## texture_mean                    9.996259
## compactness_mean                9.275500
## smoothness_mean                 5.444953
## fractal_dimension_mean          4.056726
## symmetry_mean                   3.647379
```

Interpret. Name the top three features. Is the top feature the same as the root split in Result 1? Several top features (for example `radius_mean`, `perimeter_mean`, `area_mean`) measure nearly the same thing. How might this affect how we read the importance of each one separately?

Your interpretation:

Result 6 (6 points). The number of features tried at each split (`mtry`). With `mtry = 10` (all features) the forest is *bagging*. Smaller `mtry` gives a random forest.

```
mtry_vals <- c(1, 3, 5, 10)
res <- sapply(mtry_vals, function(m) {
    set.seed(42)
    f <- randomForest(diagnosis ~ ., data = train, ntree = 500,
        mtry = m)
    c(mtry = m, OOB_error = unname(f$err.rate[500, "OOB"]),
        test_error = mean(predict(f, test) != test$diagnosis))
})
round(t(res), 3)
```

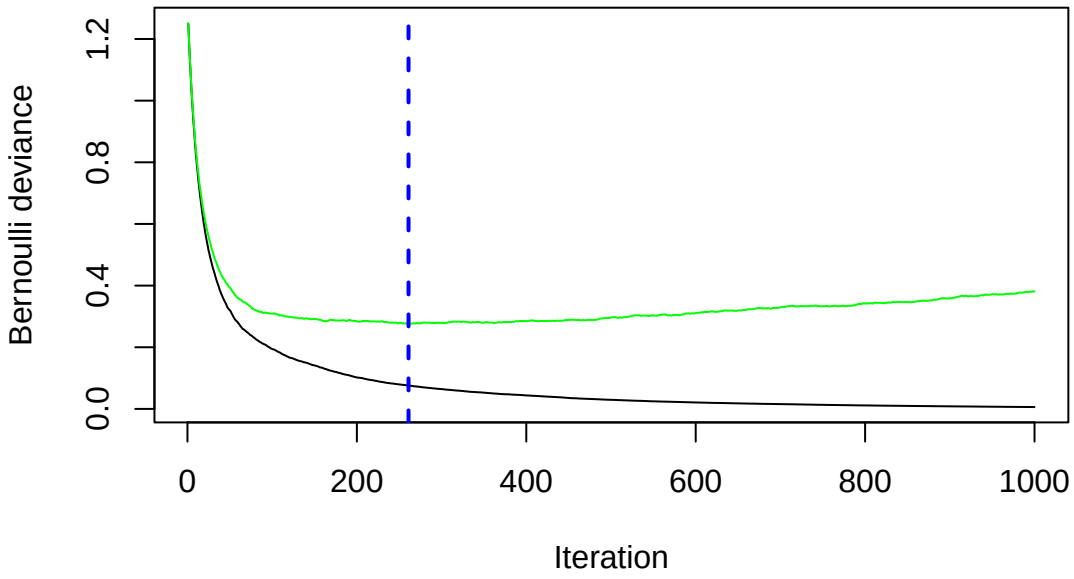
```
##      mtry OOB_error test_error
## [1,]    1     0.054      0.092
## [2,]    3     0.054      0.085
## [3,]    5     0.052      0.070
## [4,]   10     0.061      0.070
```

Interpret. Which `mtry` has the lowest OOB error? Is bagging (`mtry = 10`) better or worse than a random forest here? Explain what `mtry` changes about the trees, and why that can help. Why do the OOB and test errors not always pick the same `mtry`? (There are only 142 test observations: how many errors is one percentage point?)

Your interpretation:

Result 7 (6 points). Boosting with `gbm`: 1000 trees of depth 2 and `shrinkage = 0.05`. The plot shows the training deviance (black) and the 5-fold cross-validated deviance (green).

```r
set.seed(42)
tr <- train |>
    mutate(y = as.numeric(diagnosis == "M")) |>
    dplyr::select(-diagnosis)
gb <- gbm(y ~ ., data = tr, distribution = "bernoulli",
    n.trees = 1000, interaction.depth = 2, shrinkage = 0.05,
    cv.folds = 5)
best_iter <- gbm.perf(gb, method = "cv")
```




```r
best_iter
```

```
## [1] 261
```

```r
p_gb <- predict(gb, test, n.trees = best_iter, type = "response")
table(Predicted = ifelse(p_gb > 0.5, "M", "B"), Truth = test$diagnosis)
```

```
##          Truth
## Predicted  B  M
##         B 86  4
##         M  3 49
```

Interpret. Describe the two curves. What happens to the training deviance as trees are added, and what happens to the cross-validated deviance? What number of trees did cross-validation pick? Why does boosting risk overfitting when random forests (adding more trees) do not? What does `shrinkage` do, and what would happen to the best number of trees if we made it smaller?

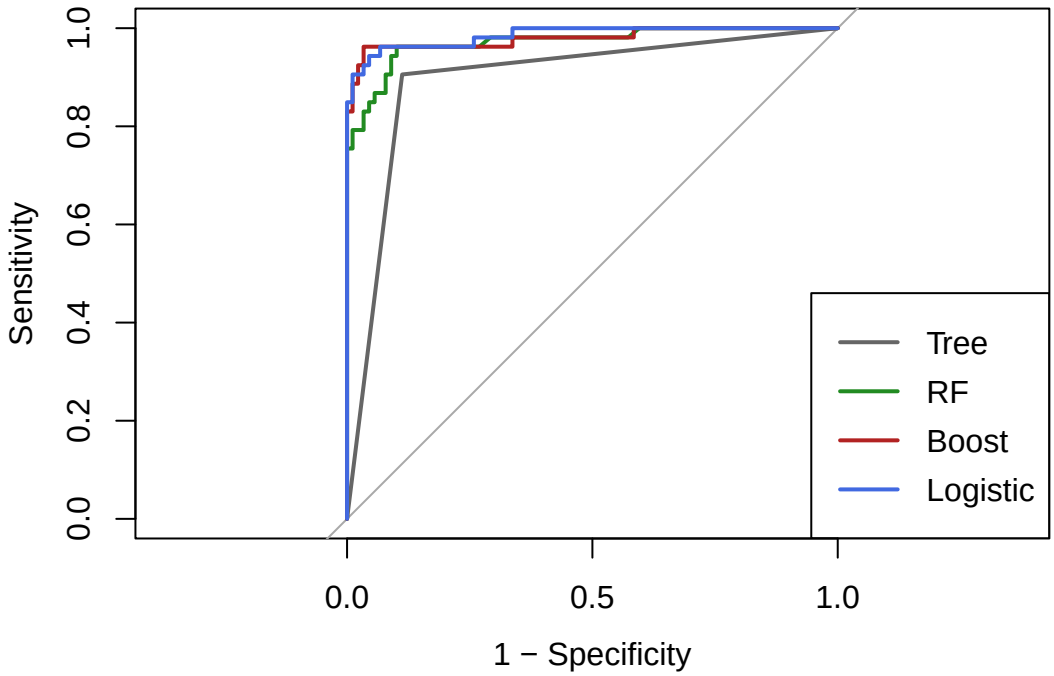
Your interpretation:

Part 3. Comparing and averaging models

Result 8 (6 points). ROC curves and AUC on the test set for four models. The logistic regression uses the standardized features.

```
pp <- preProcess(train |>
    dplyr::select(-diagnosis), method = c("center", "scale"))
trs <- predict(pp, train)
tes <- predict(pp, test)
lg <- glm(diagnosis ~ ., data = trs, family = binomial)
p_lg <- predict(lg, tes, type = "response")

probs <- list(Tree = p_tree, RF = p_rf, Boost = p_gb, Logistic = p_lg)
rocs <- lapply(probs, function(p) roc(y_test, p, quiet = TRUE))
plot(rocs$Tree, col = "gray40", legacy.axes = TRUE)
plot(rocs$RF, col = "forestgreen", add = TRUE)
plot(rocs$Boost, col = "firebrick", add = TRUE)
plot(rocs$Logistic, col = "royalblue", add = TRUE)
legend("bottomright", legend = names(rocs), lwd = 2, col = c("gray40",
    "forestgreen", "firebrick", "royalblue"))
```




```
round(sapply(rocs, auc), 3)
```

```
##     Tree       RF    Boost Logistic
##    0.897    0.972    0.980    0.985
```

Interpret. Rank the four models by AUC. How big are the gaps between the three best? With 142 test observations, would you claim that one of them is clearly the best? Why does the single tree rank last? Does the ranking match what you would expect from "ensembles beat single trees"?

Your interpretation:

Result 9 (6 points). How similar are the models? The matrix shows the correlation between the predicted malignancy probabilities on the test set.

```
P <- cbind(Tree = p_tree, RF = p_rf, Boost = p_gb, Logistic = p_lg)
round(cor(P), 3)
```

```
##           Tree    RF Boost Logistic
## Tree     1.000 0.883 0.864    0.839
## RF       0.883 1.000 0.967    0.955
## Boost    0.864 0.967 1.000    0.961
## Logistic 0.839 0.955 0.961    1.000
```

Interpret. Which pair of models is most alike, and which pair is least alike? Based on Lecture 11, which models would you combine to get the most benefit from averaging? Which pair would you expect to gain least from averaging, and why? (Think about the diversity term V from Theory T2.)

Your interpretation:

Result 10 (6 points). Averaging the random forest, boosting, and logistic regression with equal weights. The table checks the identity from Theory T2 on the test set, using squared error ($(y-\hat{p})^2$, the Brier score).

```
members <- P[, c("RF", "Boost", "Logistic")]
ens <- rowMeans(members)  # equal-weight average
mse <- function(p) mean((y_test - p)^2)

avg_member_risk <- mean(apply(members, 2, mse))
diversity <- mean(apply(members, 2, function(p) mean((p -
    ens)^2)))

member_risk <- apply(members, 2, mse)
round(c(member_risk, Avg_member = avg_member_risk, Diversity_V = diversity,
    Ensemble = mse(ens), Avg_minus_V = avg_member_risk -
        diversity), 4)
```

```
##          RF       Boost    Logistic  Avg_member Diversity_V    Ensemble
##      0.0566      0.0351      0.0395      0.0437      0.0050      0.0387
## Avg_minus_V
##      0.0387
```

```
cat("Ensemble AUC:", round(as.numeric(auc(roc(y_test, ens,
    quiet = TRUE))), 3), "\n")
```

```
## Ensemble AUC: 0.982
```

```r
all_p <- cbind(members, Ensemble = ens)
apply(all_p, 2, function(p) sum((p > 0.5) != y_test))  # test errors
```

```
##       RF    Boost Logistic Ensemble
##       12        7        7        7
```

Interpret. Does the output agree with the identity from T2(a) (compare the last two rows of the risk table)? Is the ensemble better than the average member? Is it better than the best member? Connect your answer to T2(b) and T2(c). Finally, does the ensemble beat the best member on AUC and on the number of test errors? Would you report the ensemble as "clearly better" in a paper?

Your interpretation: