HW7

Solutions

Details

Due Date

Please commit and push your submission for this assignment to GitHub by 5:00 PM Friday Nov 15.

Grading

20% of your grade on this assignment is for completion. A quick pass will be made to ensure that you've made a reasonable attempt at all problems.

Some of the problems will be graded more carefully for correctness. In grading these problems, an emphasis will be placed on full explanations of your thought process. You usually won't need to write more than a few sentences for any given problem, but you should write complete sentences! Understanding and explaining the reasons behind your decisions is more important than making the "correct" decision.

Solutions to all problems will be provided.

Collaboration

You are allowed to work with others on this assignment, but you must complete and submit your own write up. You should not copy large blocks of code or written text from another student.

Sources

You may refer to class notes, our textbook, Wikipedia, etc.. All sources you refer to must be cited in the space I have provided at the end of this problem set.

In particular, you may find the following resources to be valuable:

- Courses assigned on DataCamp
- Example R code from class
- Cheat sheets and resources linked from [http://www.evanlray.com/stat340_f2019/resources.html]

Load Packages

The following R code loads packages needed in this assignment.

```r
library(readr)
library(dplyr)
library(ggplot2)
library(caret)
```

Conceptual Problems

Problem 1: Bagging

This problem is related to random forests, which we may not get to until Monday, Nov. 11. You will may want to wait until we've talked about random forests to do this problem.

Suppose we produce ten bootstrapped samples from a data set containing red and green classes as the response. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red|X): 0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and 0.75

There are two common ways to combine these results together into a single class prediction: based on the majority vote, or based on the average probability (most often with a probability cut off of 0.5). Find the final ensemble classification based on each of these two approaches.

Majority vote:

The component model predictions are "Green", "Green", "Green", "Green", "Red", "Red", "Red", "Red", "Red", "Red". Since 6 out of 7 models predict red, the majority vote prediction is "Red".

Average probability:

```r
mean(c(0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75))
```

```
## [1] 0.45
```

The mean of the estimated class probabilities from the 10 component models is 0.45. Since this is less than the cut off of 0.5, the predicted class is "Green".

Problem 2: Stacked Ensembles Intuition

Suppose I'm doing a regression problem. I'm considering two different possible stacked ensembles, each with three component models:

1. An ensemble with a linear regression model, KNN, and regression trees.
2. An ensemble with a linear regression model, LASSO, and Ridge regression

Assuming that all 5 of the component models considered (linear regression, KNN, regression trees, LASSO, and Ridge regression) are roughly similar in performance, which of the above ensembles would you expect to perform better? Why?

I expect the ensemble combining a linear regression model, KNN, and regression trees to be better; in this case, we're told that all 5 component models have similar performance. Any differences between the ensembles will be because of the ensembling process rather than from some of the component models being particularly good. The biggest gains from building an ensemble come when the component models are diverse, so that they make uncorrelated predictions. There is more diversity among the component models in the first ensemble than in the component models in the second ensemble, since three very different model structures are used. a linear regression, LASSO, and ridge regression all build on the structure of a linear model, so their predictions will likely be highly correlated and gains from building an ensemble would be relatively limited.

Applied Problems

Problem 3: College Out-of-State Tuition

The `College` data set that comes with the ISLR package has information about 777 US colleges from the 1995 issue of US News and World Report. Let's predict a college's out of state tuition (`Outstate`) using the other variables in the data set.

```
library(ISLR)
head(College)
```

```
##                              Private Apps Accept Enroll Top10perc
## Abilene Christian University     Yes 1660   1232    721        23
## Adelphi University               Yes 2186   1924    512        16
## Adrian College                   Yes 1428   1097    336        22
## Agnes Scott College              Yes  417    349    137        60
## Alaska Pacific University        Yes  193    146     55        16
## Albertson College                Yes  587    479    158        38
##                              Top25perc F.Undergrad P.Undergrad Outstate
## Abilene Christian University        52        2885         537     7440
## Adelphi University                  29        2683        1227    12280
## Adrian College                      50        1036          99    11250
## Agnes Scott College                 89         510          63    12960
## Alaska Pacific University           44         249         869     7560
## Albertson College                   62         678          41    13500
##                              Room.Board Books Personal PhD Terminal
## Abilene Christian University       3300   450     2200  70       78
## Adelphi University                 6450   750     1500  29       30
## Adrian College                     3750   400     1165  53       66
## Agnes Scott College                5450   450      875  92       97
## Alaska Pacific University          4120   800     1500  76       72
## Albertson College                  3335   500      675  67       73
##                              S.F.Ratio perc.alumni Expend Grad.Rate
## Abilene Christian University      18.1          12   7041        60
## Adelphi University                12.2          16  10527        56
## Adrian College                    12.9          30   8735        54
## Agnes Scott College                7.7          37  19016        59
## Alaska Pacific University         11.9           2  10922        15
## Albertson College                  9.4          11   9727        55
```

(a) Construct a stacked ensemble for this task. Use the following four component models to build your ensemble: linear regression, LASSO, KNN, and a regression tree. Obtain a measure of test set performance for each of the four component models as well as the ensemble. Please also add a plot of cross-validated RMSE vs. the tuning parameters for LASSO, KNN, and the regression tree to ensure you have explored a large enough range of values for the tuning parameters of those models.

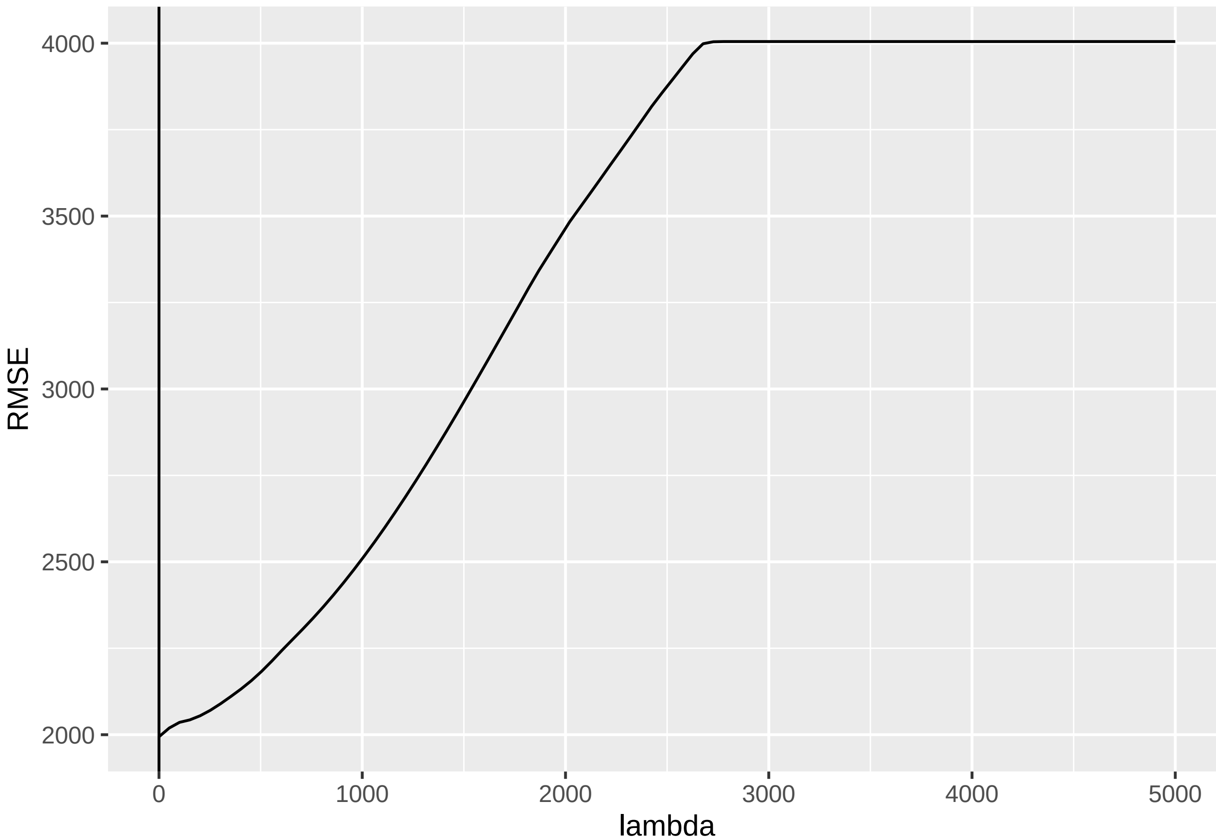
To save you some typing, I've pasted in below the code from the example of a stacked ensemble for regression from the example in class. You will need to modify this code as appropriate to work with this data set and include all four component models.

```
library(readr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(purrr)
library(glmnet)
library(caret)

# Initial train/test split ("estimation"/test) and cross-validation folds
set.seed(63770)
```

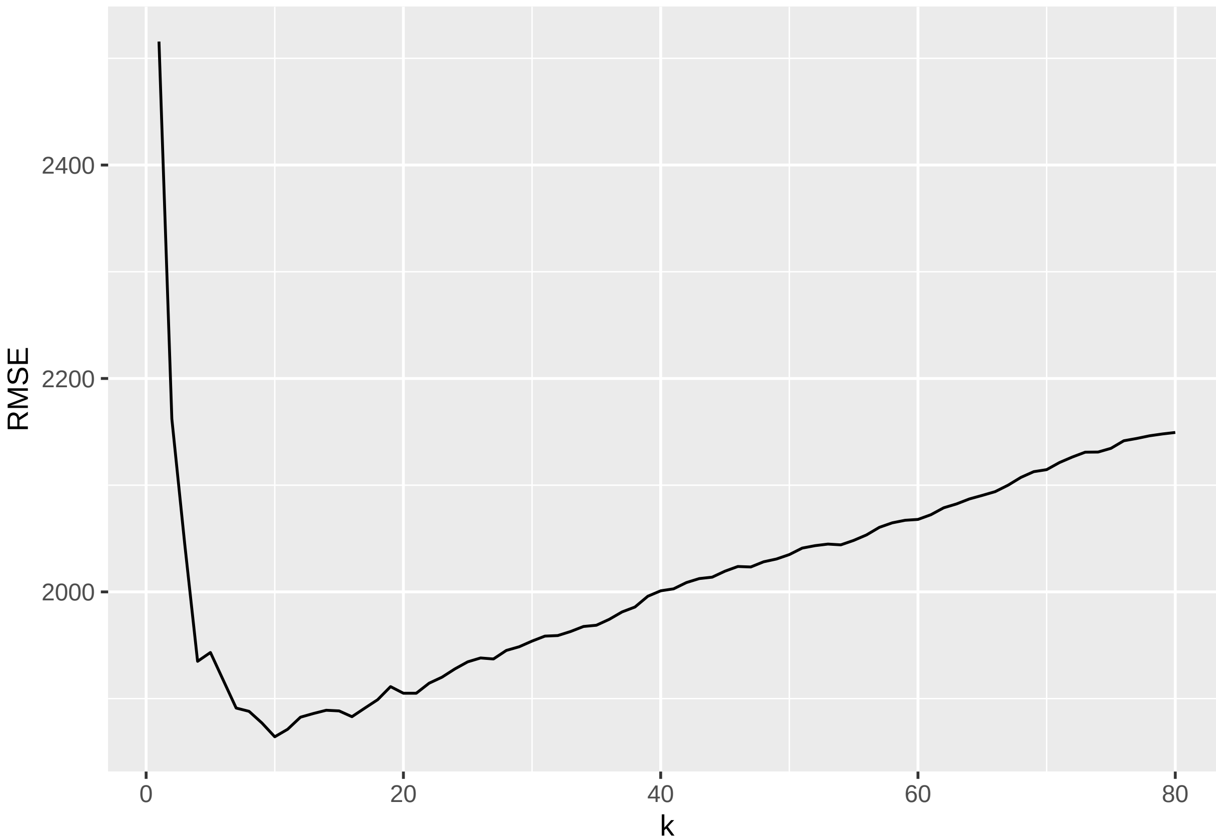
```r
tt_inds <- caret::createDataPartition(College$Outstate, p = 0.8)
train_set <- College %>% slice(tt_inds[[1]])
test_set <- College %>% slice(-tt_inds[[1]])

crossval_val_fold_inds <- caret::createFolds(
  y = train_set$Outstate, # response variable as a vector
  k = 10 # number of folds for cross-validation
)

get_complementary_inds <- function(x) {
  return(seq_len(nrow(train_set))[-x])
}
crossval_train_fold_inds <- map(crossval_val_fold_inds, get_complementary_inds)


lm_fit <- train(
  form = Outstate ~ .,
  data = train_set,
  method = "lm", # method for fit
  trControl = trainControl(method = "cv", # evaluate method performance via cross-validation
    number = 10, # number of folds for cross-validation
    index = crossval_train_fold_inds, # I'm specifying which folds to use, for consistency across metho
    indexOut = crossval_val_fold_inds, # I'm specifying which folds to use, for consistency across meth
    returnResamp = "all", # return information from cross-validation
    savePredictions = TRUE) # return validation set predictions from cross-validation
)


lasso_fit <- train(
  form = Outstate ~ .,
  data = train_set,
  method = "glmnet", # method for fit
  trControl = trainControl(method = "cv", # evaluate method performance via cross-validation
    number = 10, # number of folds for cross-validation
    index = crossval_train_fold_inds, # I'm specifying which folds to use, for consistency across metho
    indexOut = crossval_val_fold_inds, # I'm specifying which folds to use, for consistency across meth
    returnResamp = "all", # return information from cross-validation
    savePredictions = TRUE), # return validation set predictions from cross-validation
  tuneGrid = data.frame(
    alpha = 1,
    lambda = seq(from = 0, to = 5000, length = 100)
  )
)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```r
ggplot(data = lasso_fit$results, mapping = aes(x = lambda, y = RMSE)) +
  geom_line() +
  geom_vline(xintercept = lasso_fit$bestTune$lambda)
```



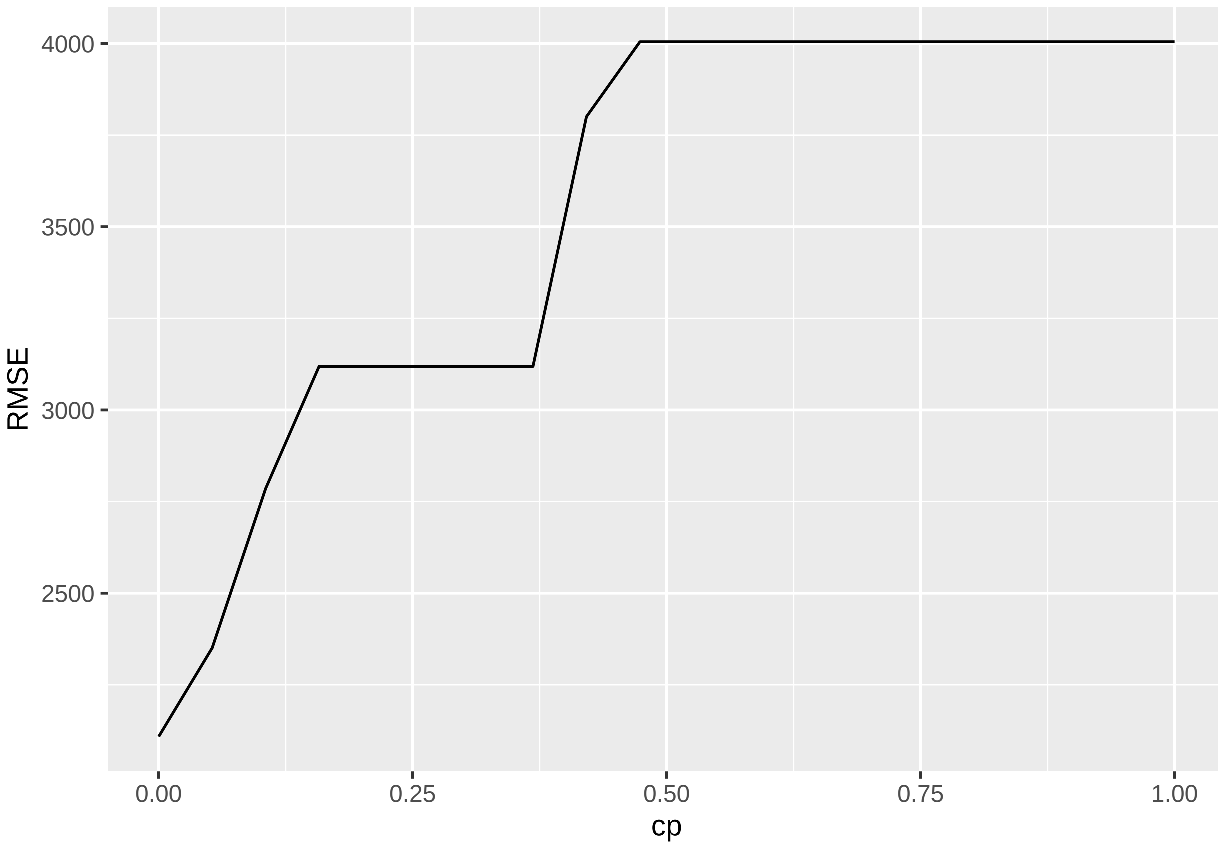

```
knn_fit <- train(
  form = Outstate ~ .,
  data = train_set,
  method = "knn",
  preProcess = "scale",
  trControl = trainControl(method = "cv",
    number = 10,
    index = crossval_train_fold_inds, # I'm specifying which folds to use, for consistency across metho
    indexOut = crossval_val_fold_inds, # I'm specifying which folds to use, for consistency across meth
    returnResamp = "all",
    savePredictions = TRUE),
  tuneGrid = data.frame(k = 1:80)
)

ggplot(data = knn_fit$results, mapping = aes(x = k, y = RMSE)) +
  geom_line() +
  geom_vline(xintercept = knn_fit$bestTune$lambda)
```




```
rpart_fit <- train(
  form = Outstate ~ .,
  data = train_set,
  method = "rpart",
  trControl = trainControl(method = "cv",
    number = 10,
    index = crossval_train_fold_inds, # I'm specifying which folds to use, for consistency across metho
    indexOut = crossval_val_fold_inds, # I'm specifying which folds to use, for consistency across meth
    returnResamp = "all",
    savePredictions = TRUE),
  tuneGrid = data.frame(cp = seq(from = 0, to = 1, length = 20))
)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```
ggplot(data = rpart_fit$results, mapping = aes(x = cp, y = RMSE)) +
  geom_line() +
  geom_vline(xintercept = knn_fit$bestTune$lambda)
```




```
# Step 1: Validation-fold predictions from component models
lm_val_pred <- lm_fit$pred %>%
  arrange(rowIndex) %>%
  pull(pred)

lasso_val_pred <- lasso_fit$pred %>%
  filter(lambda == lasso_fit$bestTune$lambda) %>%
  arrange(rowIndex) %>%
  pull(pred)

knn_val_pred <- knn_fit$pred %>%
  filter(k == knn_fit$bestTune$k) %>%
  arrange(rowIndex) %>%
  pull(pred)

rpart_val_pred <- rpart_fit$pred %>%
  filter(cp == rpart_fit$bestTune$cp) %>%
  arrange(rowIndex) %>%
  pull(pred)

# Step 2: data set with validation-set component model predictions as explanatory variables
train_set <- train_set %>%
  mutate(
    lm_pred = lm_val_pred,
    lasso_pred = lasso_val_pred,
    knn_pred = knn_val_pred,
    rpart_pred = rpart_val_pred
```

```r
  )

# Step 3: fit model using component model predictions as explanatory variables
# Here, a linear model without intercept (via lm directly because caret::train
# doesn't let you fit a model without intercept without more work).
stacking_fit <- lm(Outstate ~ 0 + lm_pred + lasso_pred + knn_pred + rpart_pred, data = train_set)
coef(stacking_fit)
```

```
##    lm_pred lasso_pred   knn_pred rpart_pred
##  5.1470321 -4.9555384  0.5098111  0.2964101
```

```r
# Step 4 (both cross-validation and refitting to the full training set were already done
# as part of obtaining lm_fit, knn_fit, and rpart_fit above)
lm_test_pred <- predict(lm_fit, newdata = test_set)
lasso_test_pred <- predict(lasso_fit, newdata = test_set)
knn_test_pred <- predict(knn_fit, newdata = test_set)
rpart_test_pred <- predict(rpart_fit, newdata = test_set)

# Step 5: Assemble data frame of test set predictions from each component model
stacking_test_x <- data.frame(
  lm_pred = lm_test_pred,
  lasso_pred = lasso_test_pred,
  knn_pred = knn_test_pred,
  rpart_pred = rpart_test_pred
)

# Step 6: Stacked model predictions
stacking_preds <- predict(stacking_fit, stacking_test_x)

# Calculate test set RMSE
sqrt(mean((test_set$Outstate - lm_test_pred)^2))
```

```
## [1] 1967.841
```

```r
sqrt(mean((test_set$Outstate - lasso_test_pred)^2))
```

```
## [1] 1969.559
```

```r
sqrt(mean((test_set$Outstate - knn_test_pred)^2))
```

```
## [1] 2076.162
```

```r
sqrt(mean((test_set$Outstate - rpart_test_pred)^2))
```

```
## [1] 2100.62
```

```r
sqrt(mean((test_set$Outstate - stacking_preds)^2))
```

```
## [1] 1861.186
```

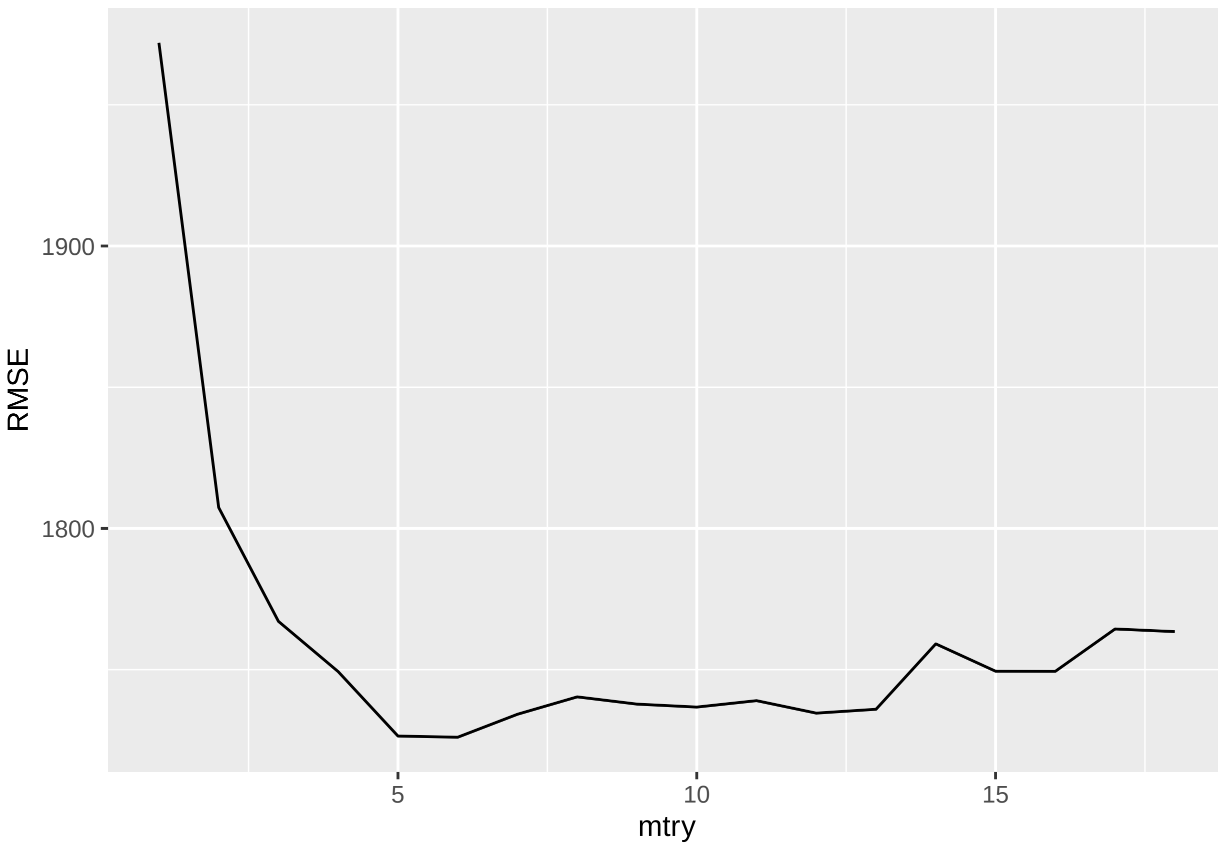
(b) Fit a random forest model to the training data. Use cross-validation to select the value of mtry; make a plot of mtry vs. RMSE to ensure you've explored a wide enough range of values for mtry. Find the test set performance for the random forest.

```r
train_set <- College %>% slice(tt_inds[[1]])
test_set <- College %>% slice(-tt_inds[[1]])
```

```r
rf_fit <- train(
  form = Outstate ~ .,
  data = train_set,
  method = "rf",
  trControl = trainControl(method = "oob"),
  tuneGrid = data.frame(mtry = seq(from = 1, to = 18))
)
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```r
ggplot(data = rf_fit$results, mapping = aes(x = mtry, y = RMSE)) +
  geom_line() +
  geom_vline(xintercept = knn_fit$bestTune$lambda)
```




```r
rf_test_pred <- predict(rf_fit, newdata = test_set)
sqrt(mean((test_set$Outstate - rf_test_pred)^2))
```

```
## [1] 1709.77
```

Collaboration and Sources

If you worked with any other students on this assignment, please list their names here.

If you referred to any sources (including our text book), please list them here. No need to get into formal citation formats, just list the name of the book(s) you used or provide a link to any online resources you used.