Homework 7

STAT 430, Spring 2017

Due: Friday, March 31 by 11:59 PM

Exercise 1

[25 points] For this assignment, we will use the `College` data from the `ISLR` package. Familiarize yourself with this dataset before performing analyses. We will attempt to predict the `Outstate` variable.

Test-train split the data using this code.

```
set.seed(42)
library(caret)
library(ISLR)
index = createDataPartition(College$Outstate, p = 0.80, list = FALSE)
college_trn = College[index, ]
college_tst = College[-index, ]
```

Train a total of **six** models using five-fold cross validation.

- An additive linear model.
- An elastic net model using additive predictors. Use a `tuneLength` of `10`.
- An elastic net model that also uses all two-way interactions. Use a `tuneLength` of `10`.
- A well-tuned KNN model.
- A well-tuned KNN model that also uses all two-way interactions. (Should this work?)
- A well-tuned GAM, trained using `method = gamSpline` with `caret`.

Before training the models, set a seed equal to your UIN.

```
uin = 123456789
set.seed(uin)
```

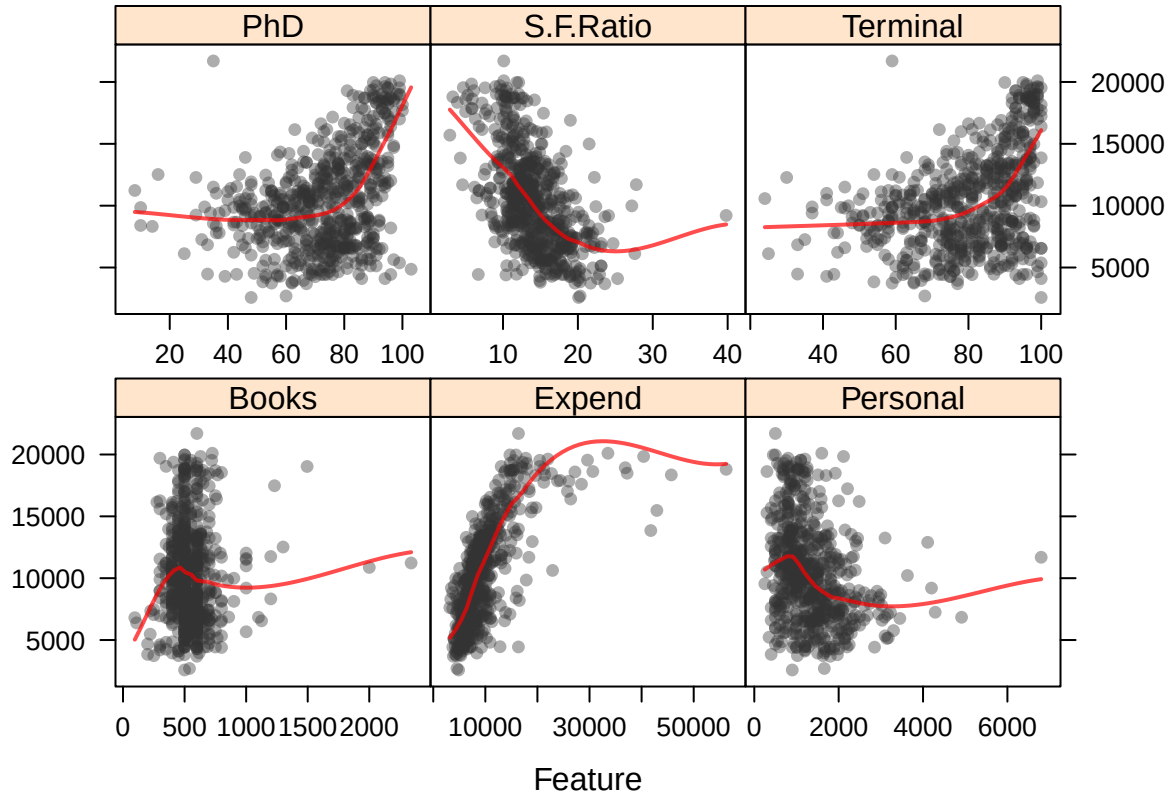
Also answer the following:

- Create a table which reports CV and Test RMSE for each.
- Based on the table, which model do you prefer? Justify your answer.
- For both of the elastic net models, report the best tuning parameters from `caret`. For each, is this ridge, lasso, or somewhere in between? If in between, closer to which?
- Did you scale the predictors when you used KNN? Should you have scaled the predictors when you used KNN?
- Of the two KNN models which works better? Can you explain why?
- For both of the KNN models, plot the CV results against the tuning parameters. Does this plot verify that you used an appropriate tuning grid?
- For the GAM, plot the CV results against the tuning parameters. Does this plot verify that you used an appropriate tuning grid?
- What was the best tuning parameter for the GAMs? Does this suggest non-linearity?
- What year is this dataset from? What was out-of-state tuition at UIUC at that time?

Solution:

Note that some code, for plotting and summarizing, is hidden. See the .Rmd file for code.

```
library(glmnet)
library(gam)
```




```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
```

```
cv_5 = trainControl(method = "cv", number = 5)
```

```
set.seed(uin)
fit_lm          = train(Outstate ~ ., data = college_trn, method = "lm",
                        trControl = cv_5)
fit_glmnet      = train(Outstate ~ ., data = college_trn, method = "glmnet",
                        trControl = cv_5, tuneLength = 10)
fit_glmnet_int = train(Outstate ~ . ^ 2, data = college_trn, method = "glmnet",
                        trControl = cv_5, tuneLength = 10)
fit_knn         = train(Outstate ~ ., data = college_trn, method = "knn",
                        trControl = cv_5, tuneLength = 25, preProcess = c("center", "scale"))
fit_knn_int     = train(Outstate ~ . ^ 2, data = college_trn, method = "knn",
                        trControl = cv_5, tuneLength = 25, preProcess = c("center", "scale"))
fit_gam         = train(Outstate ~ ., data = college_trn, method = "gamSpline",
                        trControl = cv_5, tuneGrid = expand.grid(df = 1:5))
```

```
get_best_result = function(caret_fit) {
  best_result = caret_fit$results[as.numeric(rownames(caret_fit$bestTune)), ]
  rownames(best_result) = NULL
  best_result
}
```

Method	CV RMSE	Test RMSE
Linear Model	1986.812	2010.547
Elastic Net	2013.242	2019.921
Elastic Net with Interactions	1849.239	1763.225
KNN	1939.199	1906.311
KNN with Interactions	1981.523	1963.830
GAM	1901.187	1913.001

- Standard Deviation of CV-RMSE for the "Best" Model:

```
get_best_result(fit_glmnet_int)$RMSESD
```

```
## [1] 218.2388
```

- Tuning parameters for `glmnet` models:

```
fit_glmnet$bestTune
```

```
##   alpha   lambda
## 5   0.1 34.91758
```

```
fit_glmnet_int$bestTune
```

```
##   alpha   lambda
## 5   0.1 39.90903
```

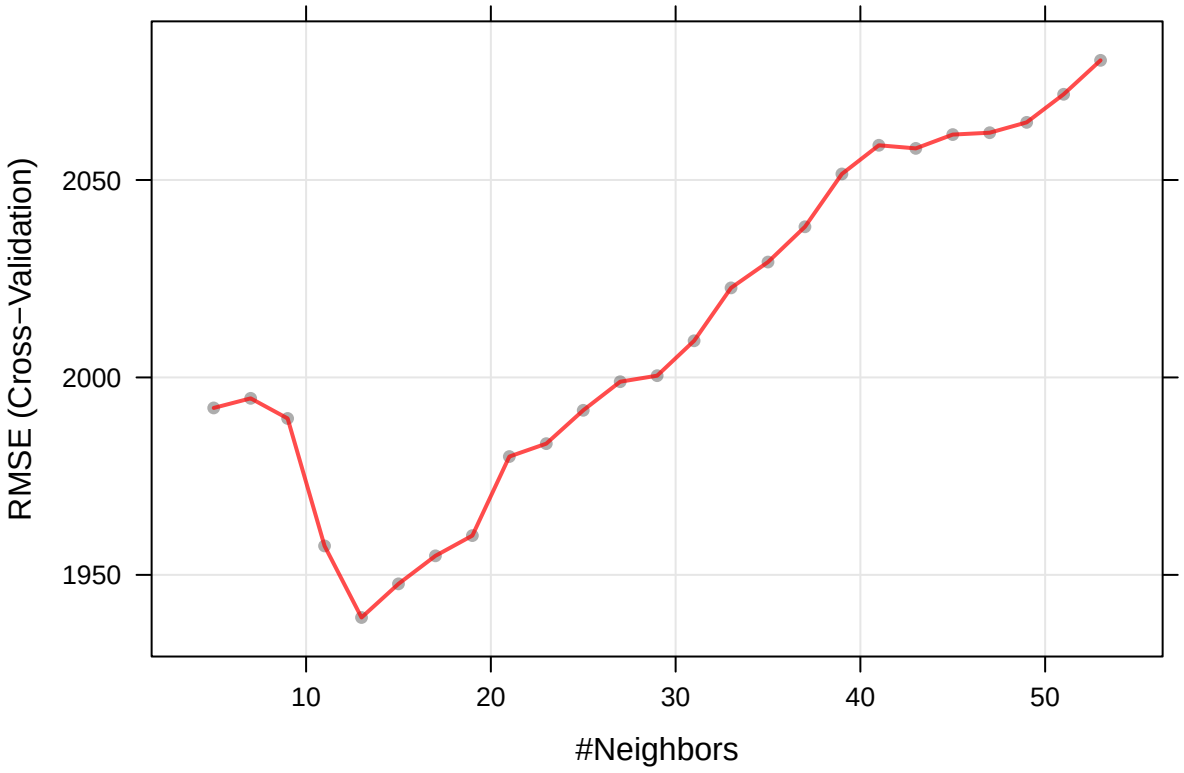
- Justification of scaled KNN:

```
fit_knn_unscaled = train(Outstate ~ ., data = college_trn, method = "knn",
                         trControl = cv_5, tuneLength = 25)
get_best_result(fit_knn_unscaled)$RMSE
```

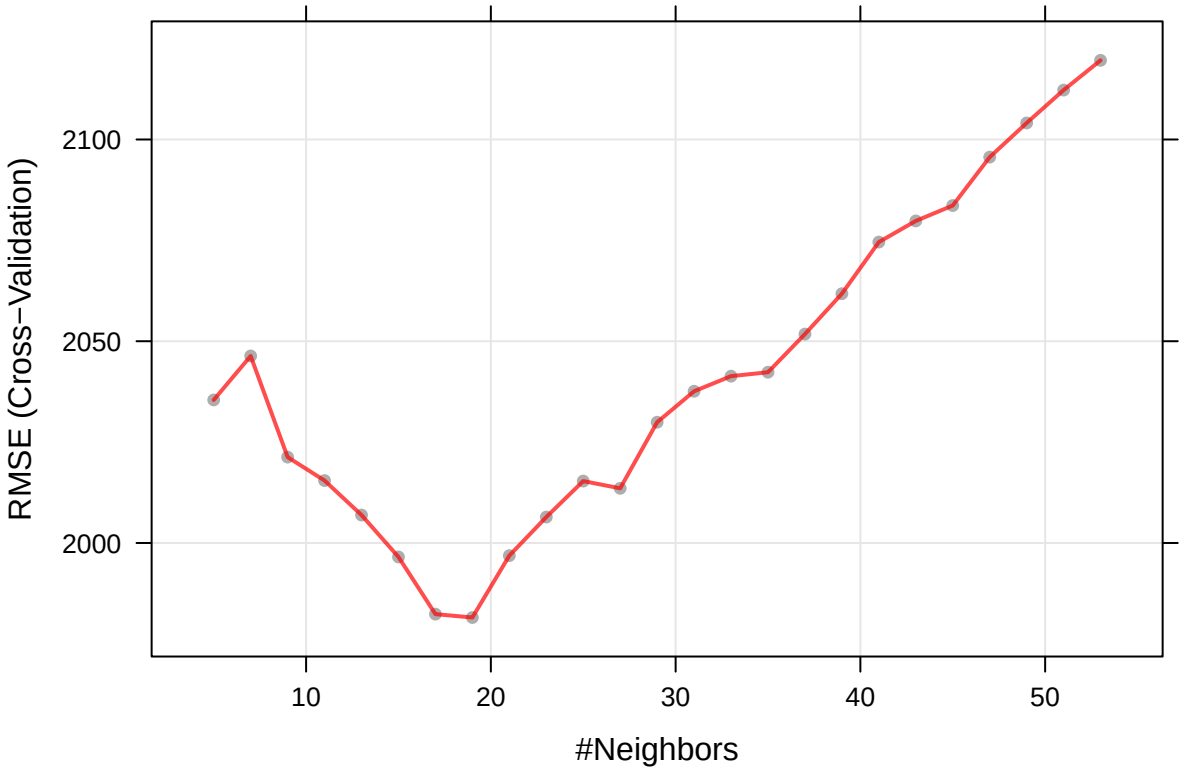
```
## [1] 2039.111
```

- KNN plots:

```
plot(fit_knn)
```

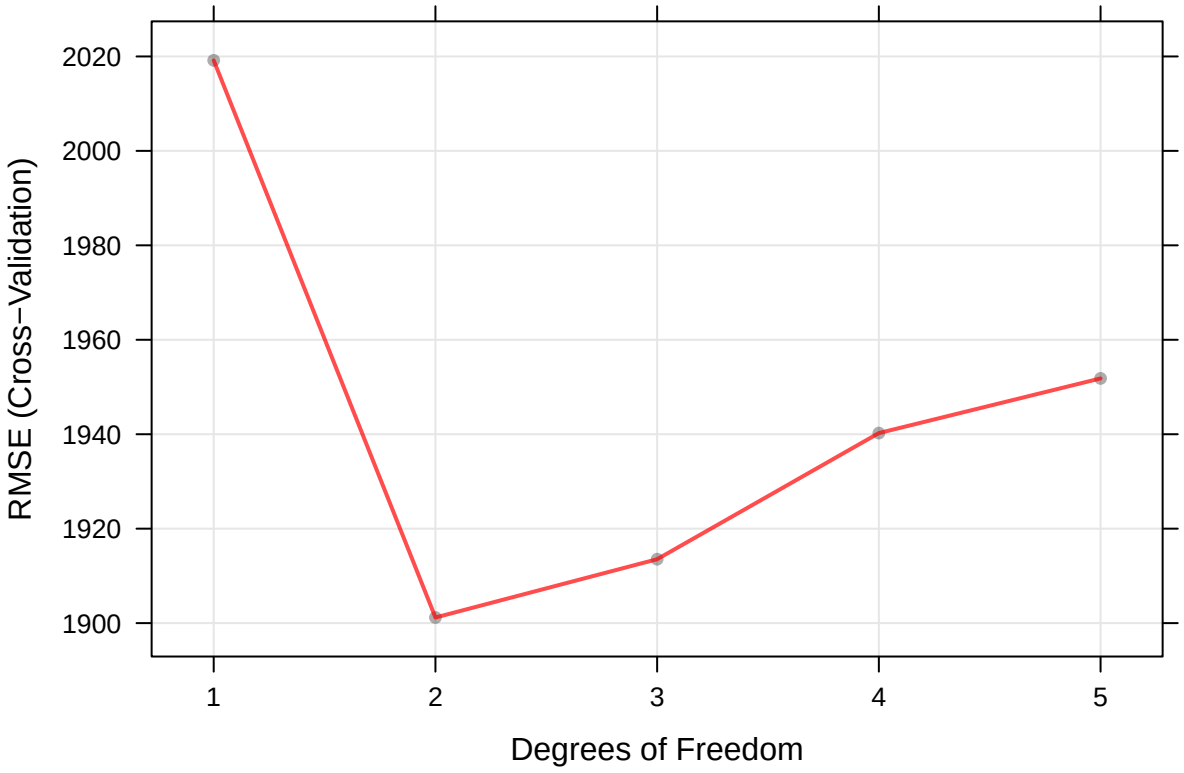



```
plot(fit_knn_int)
```



- GAM plots:

```
plot(fit_gam)
```



- Tuning parameters for `gam` models:

```
fit_gam$bestTune
```

```
##   df
## 2  2
```

Answering the questions:

- Create a table which reports CV and Test RMSE for each.
  - See above.
- Based on the table, which model do you prefer? Justify your answer.
  - The elastic-net with all interactions appears to perform the best since it obtains the lowest CV-RMSE as well as test RMSE.
- For both of the elastic net models, report the best tuning parameters from `caret`. For each, is this ridge, lasso, or somewhere in between? If in between, closer to which?
  - Seen above, they both use an α of 0.1 which is between lasso and ridge, but closer to ridge.
- Did you scale the predictors when you used KNN? Should you have scaled the predictors when you used KNN?
  - Yes! Notice the unscaled results are worse.
- Of the two KNN models which works better? Can you explain why?
  - The model without the interactions. This is probably a result of the curse of dimensionality.
- For both of the KNN models, plot the CV results against the tuning parameters. Does this plot verify that you used an appropriate tuning grid?
  - Notice that both form the expected U-shape.
- For the GAM, plot the CV results against the tuning parameters. Does this plot verify that you used an appropriate tuning grid?
  - Notice that this plot forms the expected U-shape.

- What was the best tuning parameter for the GAMs? Does this suggest non-linearity?
  - 2. Yes! This suggests non-linearity.
- What year is this dataset from? What was out-of-state tuition at UIUC at that time?
  - 1995! $7560. We're not sure if this is semester or year, but either way, wow!

Exercise 2

[5 points] Continue using the `College` data. Now use `Private` as the response variable. Fit Regularized Discriminant Analysis trained using five-fold cross-validation and a tuning length of `5` with `train()`. Use the seed below.

```
set.seed(42)
```

Report the tuning parameters and CV-Accuracy of the chosen model. Is this LDA, QDA, or something else? Also report test accuracy

```
fit_rda = train(Private ~ ., data = college_trn, method = "rda", trControl = cv_5, tuneLength = 5)
```

```
get_best_result(fit_rda)
```

```
##   gamma lambda  Accuracy     Kappa AccuracySD    KappaSD
## 1     0      1 0.9422452 0.8494863 0.02283959 0.05970799
```

```
# test accuracy
mean(predict(fit_rda, college_tst) == college_tst$Private)
```

```
## [1] 0.9285714
```

Based on these results, we see that this is LDA since $\gamma = 0$ and $\lambda = 1$.