Exam 1

Austin Vanderlyn ajl745

7/15/2022

I. True or False

1. In general the more flexible a method is, the lower its RMSE of the test data will be. Answer: False. The test data RMSE has a parabolic shape when plotted against complexity, where the lowest RMSE is at the bias/variance trade off point.

2. When we fit the linear regression model, the collinearity between predictors will improve the coefficient estimates. Answer: False. When predictors are highly correlated, the beta coefficients will be similar or equal for those correlated predictors and it's harder to tell what's meaningful.

3. All types of statistical models discussed in this course are beneficial from data pre-processing. Answer: False. Some do not need pre-processing, like tree based models, and even for ones that require pre-processing, it doesn't do any good if you do the wrong type of pre-processing. Also,

4. One advantage of Principal Component Analysis is that it is a data reduction technique which creates uncorrelated components. Answer: True.

5. The bias-variance trade off means that as a method gets more flexible, the bias will decrease, the variance will increase, but the expected RMSE may go up or down. Answer: True.

6. The trade off between prediction accuracy and interpretability means that a predictive model that is most powerful is usually the least interpretable. Answer: True (I'm a little confused by question's phrasing of "powerful", it's a little vague. If powerful means in reference to the prediction accuracy, then the answer is True, but if powerful means like how useful or meaningful the model is, then I would say False, because it has to be at least somewhat interpretable.)

7. When the sample size n is extremely large, and the number of predictors p is small, we do not expect the performance of a flexible statistical learning method to be better than an inflexible model. Answer: False. It's the other way around, when n » p, the flexible models are better

8. Elastic Net, OLS, Ridge regression, and Lasso regression can all be used and implemented in situations where the number of predictors is larger than the sample size. Answer: False. Elastic Net, ridge and lasso are all good for this situation, but not OLS, it runs into problems when $p > n$

9. The bootstrap is a widely available and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator. Each bootstrap set is created by sampling without replacement, and the size is smaller than our original dataset. Answer: False. Most of that is true, but bootstrap involves sampling WITH, not without, replacement

10. The last name of the instructor of this course is Min. Answer: False. I was a little confused here because I think generally Chinese surnames come first, but in the very first video for this course the instructor jokes about his friends calling him "Minimum" and specifically refers to Min as his first name, not last.

II. Free Response Questions

You think of some real-life applications for statistical learning and predictive modeling.

Problem 1

A. Describe a real life application in which classification might be useful. Describe the response, as well as the predictors. Is the goal of this application inference or prediction? Clearly explain your answer. So, because of the release of the James Webb telescope pictures this week I have astronomy on the brain, so one real life application might be a model that astronomers could use to classify types of stars based on the data being observed by a telescope.

The response would be a categorical variable, either red giant, red dwarf, pulsar, neutron, white dwarf, etc.

The predictors would variables like diameter, wavelength of light (color), brightness of light (luminosity), mass, temperature, etc.

The goal of this application would be inference, because you're not trying to predict what will happen, or what a value will be, you're trying to take observable data and make a guess about what it represents.

B. Describe a real-life application in which regression might be useful. Describe the response, as well as the predictors. Is the goal of this application inference or prediction? Clearly explain your answer. Well I'm also a huge baseball fan, so an application where regression might be useful could be a model to predict how many games the Astros will win next season.

The response would be total number of wins, and the predictors could be things like runs scored last season, games won last season, batting averages of players, on-base-percentage of players, runs allowed, ERA of pitching staff, WHIP of pitching staff, etc.

This would be a prediction goal, because you're using past data to try to predict future performance.

Problem 2

During the class time, we learned k-fold cross validation.

A. Explain how k-fold cross-validation is implemented. K-fold is a resampling technique that selects a subset of data points to remove from a model, uses the remaining data to fit the model, then tests the model on the selected sample. k represents the number of different subsets of the data, and then the results are averaged.

B. What are the advantages and disadvantages of k-fold cross validation relative to the bootstrap sample. The main difference between the bootstrap and k-fold is that bootstrap is sampling with replacement, and k-fold is sampling without replacement.

The advantage of k-fold in this sense is that all the data points are used to train and test across the multiple folds, whereas bootstrap sampling can oversample some points but not sample others, but the disadvantage is that the higher the k value, the more computation time it takes to run the model. Also, k-fold tends to have more uncertainty than bootstrap.

Problem 3

What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred? Some of the advantages of a highly flexible model are that there will be a lower training MSE and lower bias, and a better fit for non-linear data, but the disadvantages are that while the training MSE might be low, the testing MSE might be high, and generally, the more flexible the method is, the harder it is to interpret.

A more flexible approach would be preferred when there is a non-linear relationship in the datapoints or when the number of samples is very high.

A less flexible approach would be preferred when the number of samples is low, or there is clearly a linear trend in the data.

Problem 4

In this class, we discussed the bias-variance trade off. Answer the following questions.

a. Provide a sketch of typical (squared) bias, variance, training error, test error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be four curves. I'm attaching a scan of the sketch I did and submitting it separately.

b. Briefly explain why each of the four curves has the shape displayed in part a. The variance and bias curves have an inverse parabolic relationship where the more complex the model is, the higher the variance and lower the bias, and vice versa.

As for the MSE, the general trend for the training MSE is that it will continue to decrease as the model grows more flexible. The testing MSE, however, will decrease for a while as the model grows more flexible, until it reaches the point of overfitting and the testing MSE will start to increase.

III. Coding Questions

Problem 5

Suppose we are interested in examining the relationship between the response variable sales and the amount of money spent advertising on the TV, radio, and newspapers. We fit a multiple linear regression with four predictors, and obtain the following results.

a. Provide an appropriate interpretation for the coefficient 1.91e-02 That is the beta coefficient for the predictor TV, which means that it is the constant coefficient for TV in the linear regression equation: y = beta0 + beta1TV + *beta2*radio + beta3*newspaper* + *beta4(radio*TV).

It means, if all other predictors are held constant, that for every 1 unit increase in TV there will be a 1.91e-02 increase in the response variable sales.

b. True or false: Since the coefficient for the TV and radio interaction term "TV: radio" is quite small, there is very little evidence that this interaction term is important in predicting the response variable "sales". Justify your answer. False. First, the coefficient itself isn't relevant to deciding whether or not the term is important; the P value is. The P value for the interaction term is highly significant (<.05), so the interaction term is important, the only relevance of the magnitude of the coefficient is on how important it is relative to the other terms. While the coefficient may be small, it's not much smaller than any of the other coefficients, so this is probably just a case where any kind of changes have small impacts on sales, but the interaction term is just as important to predicting sales as TV and more important than radio.

c. Suppose that the company has two options to split $12,000 for the three types of advertising: (i) invest equally $4,000 for each type of advertising, (ii) invest $6,000 for TV, $3,000 for radio, and $3,000 for newspapers. Which option should be recommended for the company? Justify your answer. In order to answer this I'm going to set up a little code with the values from the call;

```
b0 = 6.73e+00
b1 = 1.91e-02
b2 = 2.80e-02
b3 = 1.44e-03
b4 = 1.09e-03
eq1 = b0 + b1*4000 + b2*4000 + b3*4000 + b4*4000*4000
eq2 = b0 + b1*6000 + b2*3000 + b3*3000 + b4*6000*3000
eq1
```

```
## [1] 17640.89
```

```
eq2
```

```
## [1] 19829.65
```

Based on this, the company should go with the second option, investing more in TV.

d. Based on this model fit, which predictors are important in predicting the sales? In other words, explain what conclusions you can draw based on the p-values. Your explanation should be phrased in terms of sales, TV, radio, newspaper and TV:radio, rather than in terms of the coefficients of the linear model. Well I already addressed this a little bit in part A, but we can see that three of the terms from the linear model are significant in terms of predicting sales. TV (2e-16), radio (2e-16), and the interaction term between TV and radio (0.0025) all have p-values below the standard alpha level of .05. The newspaper term (2e-16) is not significant.

Problem 6

```
library(ISLR)
```

We will predict the number of applications received using the other variables in the College data set available in the R package ISLR, which can be accessed as follows;

```
## Warning: package 'ISLR' was built under R version 4.1.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(College)
head(College)
```

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

```
dim(College)
```

```
## [1] 777  18
```

a. Appropriately split the data set into a training set (80%) and a test set (20%). Split into training and test set;

```
set.seed(123)
trainsplit = sample(1:nrow(College), 0.8 * nrow(College))
collegeTrain = College[trainsplit,]
collegeTest = College[-trainsplit,]
```

b. Fit a linear model using least squares on the training set, and report the test error obtained. Fit model;

```
ctrl = trainControl(method = "repeatedcv",
                    repeats = 10)

set.seed(123)
lm.college = train(Apps ~ .,
                   method = "lm",
                   preProc = c("center", "scale"),
                   data = collegeTrain)
summary(lm.college)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3257.7  -431.1   -57.5   318.8  6581.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2846.911     38.989  73.018  < 2e-16 ***
## PrivateYes  -267.113     65.873  -4.055 5.67e-05 ***
## Accept      2724.387    118.145  23.060  < 2e-16 ***
## Enroll      -247.023    168.861  -1.463  0.14402
## Top10perc    799.829    103.211   7.749 3.93e-14 ***
## Top25perc   -271.872     94.101  -2.889  0.00400 **
## F.Undergrad  418.028    156.851   2.665  0.00790 **
## P.Undergrad    7.631     51.172   0.149  0.88150
## Outstate    -213.830     78.904  -2.710  0.00692 **
## Room.Board   177.521     54.165   3.277  0.00111 **
## Books          8.985     41.175   0.218  0.82734
## Personal      -5.816     44.326  -0.131  0.89565
## PhD          -94.201     78.596  -1.199  0.23118
## Terminal     -74.297     77.076  -0.964  0.33546
## S.F.Ratio     15.331     53.758   0.285  0.77560
## perc.alumni  -77.188     53.540  -1.442  0.14991
```

```
## Expend       412.489     66.174   6.233 8.58e-10 ***
## Grad.Rate    184.197     53.028   3.474  0.00055 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 971.6 on 603 degrees of freedom
## Multiple R-squared:  0.9192, Adjusted R-squared:  0.9169
## F-statistic: 403.6 on 17 and 603 DF,  p-value: < 2.2e-16
```

Predictions;

```
preds.lm = predict(lm.college, collegeTest)
RMSE(preds.lm, collegeTest$Apps)
```

```
## [1] 1449.199
```

The RMSE test error from the OLS model is 1449.199.

c. fit a ridge regression model on the training set, with lambda chosen by cross validation. Report the test error obtained. Fit ridge regression model;

```
ridgeGrid = expand.grid(lambda = seq(0, 0.1,
                                     length = 10))
set.seed(123)
ridgeTune = train(Apps ~ .,
              data = collegeTrain,
              method = "ridge",
              tuneGrid = ridgeGrid,
              trControl = ctrl,
              preProc = c("center", "scale"))

ridgeTune
```

```
## Ridge Regression
##
## 621 samples
##  17 predictor
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 559, 558, 558, 560, 559, 559, ...
## Resampling results across tuning parameters:
##
##   lambda      RMSE       Rsquared   MAE
##   0.00000000   995.6235  0.9132453  617.1481
##   0.01111111   996.9740  0.9131927  624.2628
##   0.02222222  1003.2855  0.9122905  634.0911
##   0.03333333  1011.2824  0.9111416  644.3893
##   0.04444444  1019.8609  0.9099502  654.6002
##   0.05555556  1028.6210  0.9087922  664.4592
##   0.06666667  1037.4267  0.9076941  674.2790
##   0.07777778  1046.2475  0.9066627  684.1637
```

```
##   0.08888889  1055.0944  0.9056962  694.1071
##   0.10000000  1063.9934  0.9047895  704.1503
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 0.
```

Predictions;

```
ridge.preds = predict(ridgeTune, collegeTest)
RMSE(ridge.preds, collegeTest$Apps)
```

```
## [1] 1449.199
```

I'm not sure I quite understand this result but there seems to be the same test RMSE for the ridge model that there was for the basic lm, 1449.199

d. Fit an ENET model on the training set with tuning parameters chosen by cross validation. Report the test error obtained, along with the number of non-zero coefficient estimates. Fit model;

```
set.seed(123)

enetGrid = expand.grid(lambda = c(0, 0.01, .1),
                       fraction = seq(.05, 1,
                                      length = 20))

enetTune = train(Apps ~ .,
                 data = collegeTrain,
                 method = "enet",
                 tuneGrid = enetGrid,
                 trControl = ctrl,
                 preProcess = c("center", "scale"))
enetTune$bestTune
```

```
##    fraction lambda
## 16      0.8      0
```

Get coefficient estimates from the 16th model;

```
enetcoefs = data.frame(beta = predict(enetTune$finalModel,
                                      type = "coefficients",
                                      s = 16)$coefficients)
enetcoefs
```

```
##                     beta
## PrivateYes  -262.1473481
## Accept      2624.4106439
## Enroll         0.0000000
## Top10perc    714.2204932
## Top25perc   -199.3511138
## F.Undergrad  270.4935495
```

```
## P.Undergrad    0.9826091
## Outstate    -180.9253201
## Room.Board   174.8411680
## Books          3.0894872
## Personal       0.0000000
## PhD          -83.6286951
## Terminal     -68.8936576
## S.F.Ratio      3.8758099
## perc.alumni  -85.9318603
## Expend       398.9261831
## Grad.Rate    166.8197952
```

```
nonzerobeta = enetcoefs %>%
  filter(enetcoefs$beta != 0)
length(nonzerobeta$beta)
```

```
## [1] 15
```

Predictions;

```
enet.preds = predict(enetTune, collegeTest)
RMSE(enet.preds, collegeTest$Apps)
```

```
## [1] 1501.315
```

The testing RMSE for the ENET model is 1501.315, and the number of nonzero beta coefficients is 15.

e. Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these three approaches? Going to create a table of the test errors from these three models;

```
# predictions
testResults = data.frame(OBS = collegeTest)
testResults$OLS = predict(lm.college, collegeTest)
testResults$Ridge = predict(ridgeTune, collegeTest)
testResults$ENET = predict(enetTune, collegeTest)

R2 = RMSE = MAE = Accuracy = numeric(0)
R2[1] = cor(testResults$OLS, collegeTest$Apps)^2
RMSE[1] = sqrt(mean((testResults$OLS - collegeTest$Apps)^2))
MAE[1] = mean(abs(testResults$OLS - collegeTest$Apps))
Accuracy[1] = 1 - sum(abs(testResults$OLS - collegeTest$Apps)) / sum(collegeTest$Apps)

R2[2] = cor(testResults$Ridge, collegeTest$Apps)^2
RMSE[2] = sqrt(mean((testResults$Ridge - collegeTest$Apps)^2))
MAE[2] = mean(abs(testResults$Ridge - collegeTest$Apps))
Accuracy[2] = 1 - sum(abs(testResults$Ridge - collegeTest$Apps)) / sum(collegeTest$Apps)

R2[3] = cor(testResults$ENET, collegeTest$Apps)^2
RMSE[3] = sqrt(mean((testResults$ENET - collegeTest$Apps)^2))
MAE[3] = mean(abs(testResults$ENET - collegeTest$Apps))
```

```
Accuracy[3] = 1 - sum(abs(testResults$ENET - collegeTest$Apps)) / sum(collegeTest$Apps)

errors = cbind(R2, RMSE, MAE, Accuracy)
row.names(errors) = c("OLS", "Ridge", "ENET")
errors
```

```
##              R2     RMSE      MAE  Accuracy
## OLS   0.9360662 1449.199 666.8622 0.8156602
## Ridge 0.9360662 1449.199 666.8622 0.8156602
## ENET  0.9320520 1501.315 670.0832 0.8147698
```

The differences between these three models is pretty minimal. We can predict the number of applications received with about 81% accuracy. The OLS and Ridge perform virtually identically, and just slightly better than the ENET.