#### Chapter 6 (Exercise 9)

#### a) Split the data set into a training set and a test set.

Ans – I split the data set into a training and test set and started using the linear model unto the training set.

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

Ans – RMSE Rsquared MAE

0.2799768 0.9201765 0.1568743

#### c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

Ans – ## RMSE Rsquared MAE

## 0.1982193 0.9456062 0.1416414

19 x 1 sparse Matrix of class "dgCMatrix"

s1

(Intercept) 0.034871314

Private.No 0.075423210

Private.Yes -0.076037580

Accept 0.665628735

Enroll 0.090243371

Top10perc 0.107160248

Top25perc 0.011628030

F.Undergrad 0.063308800

P.Undergrad 0.017427317

Outstate -0.028995432

Room.Board 0.048720533

Books 0.012799145

Personal -0.002894430

PhD -0.017989250

Terminal -0.010434665

S.F.Ratio 0.006920126

perc.alumni -0.031683867

Expend 0.083525070

Grad.Rate 0.058131023

Graphical user interface, application, table, Excel

Description automatically generated

Chart

Description automatically generated

#### d) Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.

Ans – ## RMSE Rsquared MAE

## 0.1825821 0.9531574 0.1255619

19 x 1 sparse Matrix of class "dgCMatrix"

s1

(Intercept) -3.724310e-02

Private.No 1.370265e-01

Private.Yes -1.240903e-13

Accept 1.041851e+00

Enroll -2.027443e-01

Top10perc 2.015761e-01

Top25perc -4.629400e-02

F.Undergrad 1.250781e-02

P.Undergrad 2.949193e-02

Outstate -8.533313e-02

Room.Board 3.382643e-02

Books 5.116779e-03

Personal 6.295093e-03

PhD -3.701536e-02

Terminal -2.461461e-03

S.F.Ratio 5.385825e-03

perc.alumni -6.575661e-03

Expend 7.703703e-02

Grad.Rate 3.798576e-02

Chart, line chart

Description automatically generated

Chart

Description automatically generated

#### e) Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

Ans – ## RMSE Rsquared MAE

## 0.2478133 0.9, , 10 comps

.outcome

Private.No 0.031985972

Private.Yes -0.031985972

Accept 0.343576750

Enroll 0.305359773

Top10perc 0.042630417

Top25perc 0.027790893

F.Undergrad 0.273818439

P.Undergrad -0.049487667

Outstate 0.038573119

Room.Board 0.070607615

Books 0.016433593

Personal -0.023529455

PhD -0.023992433

Terminal -0.024182230

S.F.Ratio 0.003741623

perc.alumni -0.070567887

Expend 0.090126298

Grad.Rate 0.071302714116039 0.1716418

Chart, line chart

Description automatically generated

Chart, scatter chart

Description automatically generated

#### f) Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

Ans – ## RMSE Rsquared MAE

## 0.1873076 0.9496232 0.1365716

, , 9 comps

.outcome

Private.No 0.070906864

Private.Yes -0.070906864

Accept 1.038336212

Enroll -0.152885477

Top10perc 0.228486518

Top25perc -0.074753822

F.Undergrad -0.037088110

P.Undergrad 0.035256358

Outstate -0.083990748

Room.Board 0.038446928

Books 0.004860183

Personal 0.006223895

PhD -0.042632221

Terminal 0.006801570

S.F.Ratio 0.005333133

perc.alumni -0.007779112

Expend 0.068980219

Grad.Rate 0.037900747

Chart, line chart

Description automatically generated

#### g) 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 five approaches?

Ans – model

<chr>

RMSE

<dbl>

Rsquared

<dbl>

Linear 0.1863952 0.9501783

Ridge 0.1982193 0.9456062

Lasso 0.1825821 0.9531574

PCR 0.2478133 0.9116039

PLS 0.1873076 0.9496232

sd

1 0.9818241

Chart, scatter chart

Description automatically generated

CHAPTER 7

a) Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

Ans – |term | estimate| std.error| statistic| p.value|

|:-------------|--------:|---------:|---------:|-------:|

|(Intercept) | 0.555| 0.003| 201.021| 0|

|poly(dis, 3)1 | -2.003| 0.062| -32.271| 0|

|poly(dis, 3)2 | 0.856| 0.062| 13.796| 0|

|poly(dis, 3)3 | -0.318| 0.062| -5.124| 0|

Graphical user interface, chart, histogram

Description automatically generated

As we can see the model finds each power of the dis coefficient to be statistically significant. On the plot, the fitted line seems to describe the data well without overfitting.

b) Plot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.

Ans – Chart

Description automatically generated

We see that when fitted and tested on the same data the model with the highest polynomial degree has the lowest RSS.

Chart

Description automatically generated

c) Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results

Ans- Chart, bar chart, histogram

Description automatically generated

When tested on out-of-sample data the model with polynomial degree 4 is chosen. If we look at the plot of various polynomial power we can see that this is the highest degree that does not show clear signs of overfitting.

d) Use the bs() function to fit a regression spline to predict nox using dis. Report the output for the fit using four degrees of freedom. How did you choose the knots? Plot the resulting fit

Ans- |term | estimate| std.error| statistic| p.value|

|:----------------|--------:|---------:|---------:|-------:|

|(Intercept) | 0.734| 0.015| 50.306| 0.000|

|bs(dis, df = 4)1 | -0.058| 0.022| -2.658| 0.008|

|bs(dis, df = 4)2 | -0.464| 0.024| -19.596| 0.000|

|bs(dis, df = 4)3 | -0.200| 0.043| -4.634| 0.000|

|bs(dis, df = 4)4 | -0.389| 0.046| -8.544| 0.000|

Chart

Description automatically generated

The model finds all the different bases to be statistically significant. The prediction line seems to fit the data well without overfitting.

e) Now fit a regression spline for a range of degrees of freedom and plot the resulting fits and report the resulting RSS. Describe the results obtained.

Ans – Chart, bar chart

Description automatically generated

Again, when trained and tested on the same data, a model with very high complexity is deemed the best.

Graphical user interface

Description automatically generated with low confidence

f) Perform cross-validation or another approach in order to select the best degrees of freedom for a regression spline on this data. Describe your results.

Ans – Chart, bar chart

Description automatically generated

When validated on out-of-sample data a simpler model is chosen. Just like with polynomial validation we can see that this is the most complex model that does not show obvious signs of overfitting.

CHAPTER 8

#### a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

Ans – Created a training data set containing a random sample of 800 observations and a set containing the rest.

#### b) Fit a tree to the training data, with Purchase as the response and the other variables as predictors. Use the summary () function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?

Ans – ## Call:

## rpart(formula = Purchase ~ ., data = training, method = "class",

## control = rpart.control(cp = 0))

## n= 801

##

## CP nsplit rel error xerror xstd

## 1 0.522435897 0 1.0000000 1.0000000 0.04423447

## 2 0.023504274 1 0.4775641 0.5128205 0.03626754

## 3 0.016025641 4 0.4070513 0.4583333 0.03473842

## 4 0.009615385 5 0.3910256 0.4583333 0.03473842

## 5 0.006410256 9 0.3493590 0.4358974 0.03405724

## 6 0.001068376 10 0.3429487 0.4679487 0.03502081

## 7 0.000000000 13 0.3397436 0.4903846 0.03565844

##

## Variable importance

## LoyalCH PriceDiff SalePriceMM ListPriceDiff PriceMM

## 53 8 7 7 4

## PriceCH DiscMM PctDiscMM STORE StoreID

## 3 3 3 2 2

## WeekofPurchase DiscCH SalePriceCH PctDiscCH

## 2 2 2 1

Node number 1: 801 observations

predicted class=CH expected loss=0.3895131 P(node) =1

class counts: 489 312

probabilities: 0.610 0.390

The summary shows us that the variable LoyalCH is by far the most important for determining which orange juice a customer will buy. It also tells us that if we were to only guess the majority class we would get a baseline accuracy of 61.8%.

| | x|

|:--------|---------:|

|Accuracy | 0.8676654|

|Kappa | 0.7217437|

The simple tree model beats the baseline with an accuracy of 85.6%.

#### c) Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.

Ans – n= 801

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 801 312 CH (0.61048689 0.38951311)

## 2) LoyalCH>=0.48285 500 80 CH (0.84000000 0.16000000)

## 4) LoyalCH>=0.7645725 255 10 CH (0.96078431 0.03921569) \*

## 5) LoyalCH< 0.7645725 245 70 CH (0.71428571 0.28571429)

## 10) ListPriceDiff>=0.235 141 17 CH (0.87943262 0.12056738) \*

## 11) ListPriceDiff< 0.235 104 51 MM (0.49038462 0.50961538)

## 22) PriceDiff>=0.085 42 11 CH (0.73809524 0.26190476) \*

## 23) PriceDiff< 0.085 62 20 MM (0.32258065 0.67741935)

## 46) STORE>=3.5 11 3 CH (0.72727273 0.27272727) \*

## 47) STORE< 3.5 51 12 MM (0.23529412 0.76470588) \*

## 3) LoyalCH< 0.48285 301 69 MM (0.22923588 0.77076412)

## 6) LoyalCH>=0.282272 126 49 MM (0.38888889 0.61111111)

## 12) PriceDiff>=0.195 68 31 CH (0.54411765 0.45588235)

## 24) LoyalCH< 0.3084325 7 0 CH (1.00000000 0.00000000) \*

## 25) LoyalCH>=0.3084325 61 30 MM (0.49180328 0.50819672)

## 50) WeekofPurchase>=248.5 40 17 CH (0.57500000 0.42500000)

## 100) STORE< 1.5 26 9 CH (0.65384615 0.34615385) \*

## 101) STORE>=1.5 14 6 MM (0.42857143 0.57142857) \*

## 51) WeekofPurchase< 248.5 21 7 MM (0.33333333 0.66666667) \*

## 13) PriceDiff< 0.195 58 12 MM (0.20689655 0.79310345) \*

## 7) LoyalCH< 0.282272 175 20 MM (0.11428571 0.88571429)

## 14) LoyalCH>=0.051325 110 19 MM (0.17272727 0.82727273)

## 28) LoyalCH< 0.203377 65 15 MM (0.23076923 0.76923077)

## 56) LoyalCH>=0.180654 7 3 CH (0.57142857 0.42857143) \*

## 57) LoyalCH< 0.180654 58 11 MM (0.18965517 0.81034483) \*

## 29) LoyalCH>=0.203377 45 4 MM (0.08888889 0.91111111) \*

## 15) LoyalCH< 0.051325 65 1 MM (0.01538462 0.98461538) \*

The root is split into nodes using the varable LoyalCH which was also determined to be the most important variable for the model in the summary. If a customer scored LoyalCH≥0.51 they are predicted to be in the class CH. That means we expect them to buy Citrus Hill instead of Minute Maid.

We can see from the output that the accuracy of that node is 81.9%

#### d) Create a plot of the tree and interpret the results.

Ans – Timeline

Description automatically generated

The plot shows us the same results as the output above. If we have information about a particular customer, we can use the plot to predict which brand of orange juice they will buy.

#### e) Predict the response on the test data and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

Ans – ## Confusion Matrix and Statistics

##

## Reference

## Prediction CH MM

## CH 140 25

## MM 24 80

##

## Accuracy : 0.8178

## 95% CI : (0.7664, 0.8621)

## No Information Rate : 0.6097

## P-Value [Acc > NIR] : 1.354e-13

##

## Kappa : 0.6166

## Mcnemar's Test P-Value : 1

##

## Sensitivity : 0.8537

## Specificity : 0.7619

## Pos Pred Value : 0.8485

## Neg Pred Value : 0.7692

## Prevalence : 0.6097

## Detection Rate : 0.5204

## Detection Prevalence : 0.6134

## Balanced Accuracy : 0.8078

##

## 'Positive' Class : CH

The test accuracy is found to be 78.2%.

#### f) Apply the cv.tree() function to the training set in order to determine the optimal tree size.

Ans – ## CART

##

## 801 samples

## 17 predictor

## 2 classes: 'CH', 'MM'

##

## No pre-processing

## Resampling: Cross-Validated (10 fold)

## Summary of sample sizes: 721, 721, 720, 722, 721, 721, ...

## Resampling results across tuning parameters:

##

## cp Accuracy Kappa

## 0.00000000 0.8077877 0.5948491

## 0.05555556 0.8065227 0.5922651

## 0.11111111 0.8065227 0.5922651

## 0.16666667 0.8065227 0.5922651

## 0.22222222 0.8065227 0.5922651

## 0.27777778 0.8065227 0.5922651

## 0.33333333 0.8065227 0.5922651

## 0.38888889 0.8065227 0.5922651

## 0.44444444 0.8065227 0.5922651

## 0.50000000 0.7902727 0.5375197

##

## Accuracy was used to select the optimal model using the largest value.

## The final value used for the model was cp = 0.

#### g) Produce a plot with tree size on the x-axis and cross-validated classification error rate on the y-axis.

Ans - Chart, line chart

Description automatically generated

#### h) Which tree size corresponds to the lowest cross-validated classification error rate?

Ans – cp Accuracy Kappa AccuracySD KappaSD

0.0000000 0.8077877 0.5948491 0.0320064 0.0700690

0.0555556 0.8065227 0.5922651 0.0431349 0.0914759

0.1111111 0.8065227 0.5922651 0.0431349 0.0914759

0.1666667 0.8065227 0.5922651 0.0431349 0.0914759

0.2222222 0.8065227 0.5922651 0.0431349 0.0914759

0.2777778 0.8065227 0.5922651 0.0431349 0.0914759

0.3333333 0.8065227 0.5922651 0.0431349 0.0914759

0.3888889 0.8065227 0.5922651 0.0431349 0.0914759

0.4444444 0.8065227 0.5922651 0.0431349 0.0914759

0.5000000 0.7902727 0.5375197 0.0750970 0.2092606

#### i) Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.

Ans – ## n= 801

##

## node), split, n, loss, yval, (yprob)

## \* denotes terminal node

##

## 1) root 801 312 CH (0.61048689 0.38951311)

## 2) LoyalCH>=0.48285 500 80 CH (0.84000000 0.16000000)

## 4) LoyalCH>=0.7645725 255 10 CH (0.96078431 0.03921569) \*

## 5) LoyalCH< 0.7645725 245 70 CH (0.71428571 0.28571429)

## 10) ListPriceDiff>=0.235 141 17 CH (0.87943262 0.12056738) \*

## 11) ListPriceDiff< 0.235 104 51 MM (0.49038462 0.50961538)

## 22) PriceDiff>=0.085 42 11 CH (0.73809524 0.26190476) \*

## 23) PriceDiff< 0.085 62 20 MM (0.32258065 0.67741935) \*

## 3) LoyalCH< 0.48285 301 69 MM (0.22923588 0.77076412) \*

We now have a pruned tree modelled on the whole training set by using the value of Cp obtained from cross-validation.

Timeline

Description automatically generated

#### j) Compare the training error rates between the pruned and unpruned trees. Which is higher?

Ans – Accuracy 0.8676654

Kappa 0.7217437

Accuracy 0.8414482

Kappa 0.6761968

The unpruned model has higher training accuracy. This does not mean we should never prune. It just means that the training set is well representative of the testing set.

#### k) Compare the test error rates between the pruned and unpruned trees. Which is higher?

Ans – Accuracy 0.8178439

Kappa 0.6166196

Accuracy 0.8104089

Kappa 0.6090228

But again…as we can see the unpruned model has higher accuracy and again this does not mean we should never prune.