Ch5_EOC_RandallPlyler

Randall Plyler

1/24/2022

Consider Figure 5.12, the decile-wise lift chart for the transaction data model, applied to new data.

a. Interpret the meaning of the first and second bars from the left.

The left two bars indicate that the model correctly classified the model performance. The chart aggregates the lift information and classifies that the best two mean responses are classified in the 1 and 2 percntile blocks.

b. Explain how you might use this information in practice.

When wanting to best predict a business problem or scenario, looking at a decile bar chart can assist in determining the correct amount to predict.

c. Another analyst comments that you could improve the accuracy of the model by classifying everything as nonfraudulent. If you do that, what is the error rate?

This would indicate that the decile-wise lift chart with the buckets that are smaller would indicate the nonfraudulent activity. This would be approximately the 5% range.

d. Comment on the usefulness, in this situation, of these two metrics of model performance (error rate and lift).

Error rate does a better job at determining what the misclassification rate is, while lift does a better job at providing effectiveness of the classification model. Lift ultimately does a better job at evaluating performance of the classification models.

```
#The overall goal is to use a subset of records to determine the highest cummulative predicted values o

#In practice, this can be used like the example the book provides where in the toyota cars dataset, the
```

A large number of insurance records are to be examined to develop a model for predicting fraudulent claims. Of the claims in the historical database, 1% were judged 148 EVALUATING PREDICTIVE PERFORMANCE to be fraudulent. A sample is taken to develop a model, and oversampling is used to provide a balanced sample in light of the very low response rate. When applied to this sample (n = 800), the model ends up correctly classifying 310 frauds, and 270 nonfrauds. It missed 90 frauds, and classified 130 records incorrectly as frauds when they were not.

a. Produce the confusion matrix for the sample as it stands.

b. Find the adjusted misclassification rate (adjusting for the oversampling).

Missclasification Rate = 27.5%

c. What percentage of new records would you expect to be classified as fraudulent?

Fraudlent Rate = 55%

```
DataFrameA <- matrix(c(310,90,400,130,270,400,440,360,800), ncol=3, byrow=TRUE)
DataFrameA <- confusionMatrix(DataFrameA)
DataFrameA
```

```
## Confusion Matrix and Statistics
##
##     A   B   C
## A 310  90 400
## B 130 270 400
## C 440 360 800
##
## Overall Statistics
##
##                Accuracy : 0.4312
##                  95% CI : (0.414, 0.4486)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 1.00000
##
##                   Kappa : 0.09
##
##  Mcnemar's Test P-Value : 0.01029
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C
## Sensitivity           0.35227  0.37500     0.50
## Specificity           0.78879  0.78629     0.50
## Pos Pred Value        0.38750  0.33750     0.50
## Neg Pred Value        0.76250  0.81250     0.50
## Prevalence            0.27500  0.22500     0.50
## Detection Rate        0.09688  0.08438     0.25
## Detection Prevalence  0.25000  0.25000     0.50
## Balanced Accuracy     0.57053  0.58065     0.50
```

```
DataFrameB <- matrix(c(310,90,400,130,270,400,440,360,800), ncol=3, byrow=TRUE)
colnames(DataFrameB)<-c("Fraud","NonFraud","Total" )
rownames(DataFrameB)<-c("Fraud","NonFraud","Total" )

DataFrameB
```

```
##          Fraud NonFraud Total
## Fraud      310       90   400
## NonFraud   130      270   400
## Total      440      360   800
```

```
#Actual percentage of fraud ->50%
#Predicted Fraud ->55%
#
```

Table 5.7 shows a small set of predictive model validation results for a classification model, with both actual values and propensities. a. Calculate error rates, sensitivity, and specificity using cutoffs of 0.25, 0.5, and 0.75. b. Create a decile-wise lift chart in R.

```
DataFrame1 <- matrix(c(.03,0,.52,0,.38,0,.82,1,.33,0,.42,0,.55,1,.59,0,.09,0,.21,0,.43,0,.04,0,.08,0,.1
colnames(DataFrame1)<-c("Propensity of 1","Actual")
DataFrame1<- as.data.frame(DataFrame1)
confusionMatrix(as.factor(ifelse(DataFrame1$`Propensity of 1`>0.25, '1', '0')),as.factor(DataFrame1$Actu
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 9 0
##          1 8 3
##
##                Accuracy : 0.6
##                  95% CI : (0.3605, 0.8088)
##     No Information Rate : 0.85
##     P-Value [Acc > NIR] : 0.99867
##
##                   Kappa : 0.2523
##
##  Mcnemar's Test P-Value : 0.01333
##
##             Sensitivity : 0.5294
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.2727
##              Prevalence : 0.8500
##          Detection Rate : 0.4500
##    Detection Prevalence : 0.4500
##       Balanced Accuracy : 0.7647
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(as.factor(ifelse(DataFrame1$`Propensity of 1`>0.5, '1', '0')),as.factor(DataFrame1$Actua
```

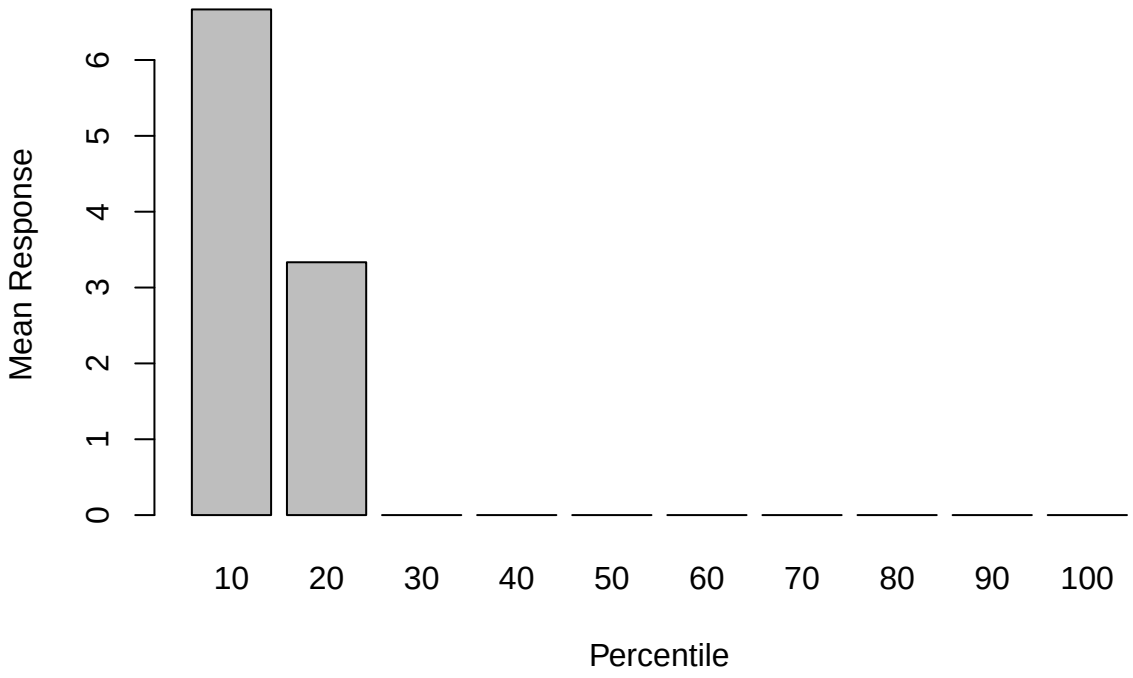
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 15  0
##          1  2  3
##
##                Accuracy : 0.9
##                  95% CI : (0.683, 0.9877)
```

```
##     No Information Rate : 0.85
##     P-Value [Acc > NIR] : 0.4049
##
##                   Kappa : 0.6923
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.8824
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6000
##              Prevalence : 0.8500
##          Detection Rate : 0.7500
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.9412
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(as.factor(ifelse(DataFrame1$`Propensity of 1`>0.75, '1', '0')),as.factor(DataFrame1$Actu
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 17  1
##          1  0  2
##
##                Accuracy : 0.95
##                  95% CI : (0.7513, 0.9987)
##     No Information Rate : 0.85
##     P-Value [Acc > NIR] : 0.1756
##
##                   Kappa : 0.7727
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 1.0000
##             Specificity : 0.6667
##          Pos Pred Value : 0.9444
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8500
##          Detection Rate : 0.8500
##    Detection Prevalence : 0.9000
##       Balanced Accuracy : 0.8333
##
##        'Positive' Class : 0
##
```

```
gain <- gains(DataFrame1$Actual, DataFrame1$`Propensity of 1`)
barplot(gain$mean.resp/mean(DataFrame1$Actual), names.arg=gain$depth, xlab="Percentile",ylab ="Mean Resp
```


Mean Response
0
1
2
3
4
5
6
10
20
30
40
50
60
70
80
90
100
Percentile