In Class Assignment 5

Lydia Strebe

February 25, 2019

Playoff Predictions

The following three models predict whether or not a team will make it to the playoffs. Each model is evaluated using cross-validation. The outputs are the average percentage of misclassified data (same as the average mean-squared error - the last output), the average false positive rate, and the average false negative rate. These outputs are given for the National League as well as the American League.

Model 1 - Single Logistic Regression

```r
#Creating data frams for all models
all_baseball = read.csv("C:/Users/lydia/Documents/baseball.csv",header=TRUE)
rows_to_keep = all_baseball$Year > 1993
baseball = all_baseball[rows_to_keep,]

NL_rows = baseball$League=="NL"
NL_data = baseball[NL_rows,]

AL_rows = baseball$League=="AL"
AL_data = baseball[AL_rows,]

#Model 1

#Create arrays to hold outputs
NL_error=array(dim = 2011-1995)
AL_error=array(dim = 2011-1995)
NL_fp=array(dim = 2011-1995)
AL_fp=array(dim = 2011-1995)
NL_fn=array(dim = 2011-1995)
AL_fn=array(dim = 2011-1995)
NL_MSE=array(dim = 2011-1995)
AL_MSE=array(dim = 2011-1995)

#This is training and validating models for every year
for(Year in 1996:2011){
  #making the training set
  train_set = (baseball$Year!=Year)
  NL_train_set = (NL_data$Year!=Year)
  AL_train_set = (AL_data$Year!=Year)
  #making the validation set
  NL_data.year = NL_data[!NL_train_set,]
  AL_data.year = AL_data[!AL_train_set,]
  #fitting the training set
  log_fit = glm(Playoffs~RS+RA,data=baseball,subset=train_set,family=binomial)
  #predict prob of playoff on validation set
  NL_prob=predict(log_fit,NL_data.year,type="response")
  AL_prob=predict(log_fit,AL_data.year,type="response")
  #order probabilities
  NL_ordered = order(NL_prob,decreasing=TRUE)
  AL_ordered = order(AL_prob,decreasing=TRUE)
  #index of the 4th best team ("worst best")
  NL_worst_best = NL_ordered[4]
  AL_worst_best = AL_ordered[4]
  #predicted list of playoff teams
  NL_pred = ifelse(NL_prob < NL_prob[NL_worst_best],0,1)
  AL_pred = ifelse(AL_prob < AL_prob[AL_worst_best],0,1)
  #actual list of playoff teams
  NL.Playoffs.year=NL_data$Playoffs[!NL_train_set]
  AL.Playoffs.year=AL_data$Playoffs[!AL_train_set]
  #confusion matrix
  NL_matrix=table(NL_pred,NL.Playoffs.year)
  AL_matrix=table(AL_pred,AL.Playoffs.year)
  #percentage of incorrect predictions (error rate)
  NL_error[Year-1995]=(NL_matrix[1,2]+NL_matrix[2,1])/length(NL_pred)
  AL_error[Year-1995]=(AL_matrix[1,2]+AL_matrix[2,1])/length(AL_pred)
  #false positive rate
  NL_fp[Year-1995]=NL_matrix[2,1]/(NL_matrix[2,1]+NL_matrix[1,1])
  AL_fp[Year-1995]=AL_matrix[2,1]/(AL_matrix[2,1]+AL_matrix[1,1])
  #false negative rate
  NL_fn[Year-1995]=NL_matrix[1,2]/(NL_matrix[2,2]+NL_matrix[1,2])
  AL_fn[Year-1995]=AL_matrix[1,2]/(AL_matrix[2,2]+AL_matrix[1,2])
  #mean squared error
  NL_MSE[Year-1995] = sum((NL.Playoffs.year-NL_pred)^2)/length(NL_pred)
  AL_MSE[Year-1995] = sum((AL.Playoffs.year-AL_pred)^2)/length(AL_pred)
}
NL_avg_error=mean(NL_error)
AL_avg_error=mean(AL_error)
NL_avg_fp=mean(NL_fp)
NL_avg_fn=mean(NL_fn)
AL_avg_fp=mean(AL_fp)
AL_avg_fn=mean(AL_fn)
NL_avg_MSE = mean(NL_MSE)
AL_avg_MSE = mean(AL_MSE)

NL_avg_error
```

```
## [1] 0.1194196
```

```
AL_avg_error
```

```
## [1] 0.09821429
```

```
NL_avg_fp
```

```
## [1] 0.08020833
```

```
NL_avg_fn
```

```
## [1] 0.234375
```

```
AL_avg_fp
```

```
## [1] 0.06875
```

```
AL_avg_fn
```

```
## [1] 0.171875
```

```
NL_avg_MSE
```

```
## [1] 0.1194196
```

```
AL_avg_MSE
```

```
## [1] 0.09821429
```

Model 2 - Logistic Regression with Dummy Variable for League

```
#Model 2

#Create arrays to hold outputs
NL_error2=array(dim = 2011-1995)
AL_error2=array(dim = 2011-1995)
NL_fp2=array(dim = 2011-1995)
AL_fp2=array(dim = 2011-1995)
NL_fn2=array(dim = 2011-1995)
AL_fn2=array(dim = 2011-1995)
NL_MSE2=array(dim = 2011-1995)
AL_MSE2=array(dim = 2011-1995)

#This is training and validating models for every year
for(Year in 1996:2011){
  #making the training set
  train_set = (baseball$Year!=Year)
  NL_train_set = (NL_data$Year!=Year)
  AL_train_set = (AL_data$Year!=Year)
  #making the validation set
  NL_data.year = NL_data[!NL_train_set,]
  AL_data.year = AL_data[!AL_train_set,]
  #fitting the training set
  log_fit = glm(Playoffs~RS+RA+League,data=baseball,subset=train_set,family=binomial)
  #predict prob of playoff on validation set
  NL_prob=predict(log_fit,NL_data.year,type="response")
  AL_prob=predict(log_fit,AL_data.year,type="response")
  #order probabilities
  NL_ordered = order(NL_prob,decreasing=TRUE)
  AL_ordered = order(AL_prob,decreasing=TRUE)
  #index of the 4th best team ("worst best")
  NL_worst_best = NL_ordered[4]
  AL_worst_best = AL_ordered[4]
  #predicted list of playoff teams
  NL_pred = ifelse(NL_prob < NL_prob[NL_worst_best],0,1)
  AL_pred = ifelse(AL_prob < AL_prob[AL_worst_best],0,1)
  #actual list of playoff teams
  NL.Playoffs.year=NL_data$Playoffs[!NL_train_set]
  AL.Playoffs.year=AL_data$Playoffs[!AL_train_set]
  #confusion matrix
  NL_matrix=table(NL_pred,NL.Playoffs.year)
  AL_matrix=table(AL_pred,AL.Playoffs.year)
  #percentage of incorrect predictions (error rate)
  NL_error2[Year-1995]=(NL_matrix[1,2]+NL_matrix[2,1])/length(NL_pred)
  AL_error2[Year-1995]=(AL_matrix[1,2]+AL_matrix[2,1])/length(AL_pred)
  #false positive rate
  NL_fp2[Year-1995]=NL_matrix[2,1]/(NL_matrix[2,1]+NL_matrix[1,1])
  AL_fp2[Year-1995]=AL_matrix[2,1]/(AL_matrix[2,1]+AL_matrix[1,1])
  #false negative rate
  NL_fn2[Year-1995]=NL_matrix[1,2]/(NL_matrix[2,2]+NL_matrix[1,2])
  AL_fn2[Year-1995]=AL_matrix[1,2]/(AL_matrix[2,2]+AL_matrix[1,2])
  #mean squared error
  NL_MSE2[Year-1995] = sum((NL.Playoffs.year-NL_pred)^2)/length(NL_pred)
  AL_MSE2[Year-1995] = sum((AL.Playoffs.year-AL_pred)^2)/length(AL_pred)
}

NL_avg_error2=mean(NL_error2)
AL_avg_error2=mean(AL_error2)
NL_avg_fp2=mean(NL_fp2)
NL_avg_fn2=mean(NL_fn2)
AL_avg_fp2=mean(AL_fp2)
AL_avg_fn2=mean(AL_fn2)
NL_avg_MSE2 = mean(NL_MSE2)
AL_avg_MSE2 = mean(AL_MSE2)

NL_avg_error2
```

```
## [1] 0.1194196
```

```
AL_avg_error2
```

```
## [1] 0.1071429
```

```
NL_avg_fp2
```

```
## [1] 0.08020833
```

```
NL_avg_fn2
```

```
## [1] 0.234375
```

```
AL_avg_fp2
```

```
## [1] 0.075
```

```
AL_avg_fn2
```

```
## [1] 0.1875
```

```
NL_avg_MSE2
```

```
## [1] 0.1194196
```

```
AL_avg_MSE2
```

```
## [1] 0.1071429
```

Model 3 - Two Logistic Regressions - One for Each League

```
#Model 3

#Create arrays to hold outputs
NL_error3=array(dim = 2011-1995)
AL_error3=array(dim = 2011-1995)
NL_fp3=array(dim = 2011-1995)
AL_fp3=array(dim = 2011-1995)
NL_fn3=array(dim = 2011-1995)
AL_fn3=array(dim = 2011-1995)
NL_MSE3=array(dim = 2011-1995)
AL_MSE3=array(dim = 2011-1995)

#This is training and validating models for every year
for(Year in 1996:2011){
  #making the training set
  NL_train_set = (NL_data$Year!=Year)
  AL_train_set = (AL_data$Year!=Year)
  #making the validation set
  NL_data.year = NL_data[!NL_train_set,]
  AL_data.year = AL_data[!AL_train_set,]
  #fitting the training set
  NL_log_fit = glm(Playoffs~RS+RA,data=NL_data,subset=NL_train_set,family=binomial)
  AL_log_fit = glm(Playoffs~RS+RA,data=AL_data,subset=AL_train_set,family=binomial)
  #predict prob of playoff on validation set
  NL_playoff_prob=predict(NL_log_fit,NL_data.year,type="response")
  AL_playoff_prob=predict(AL_log_fit,AL_data.year,type="response")
  #order probabilities
  NL_prob_ordered = order(NL_playoff_prob,decreasing=TRUE)
  AL_prob_ordered = order(AL_playoff_prob,decreasing=TRUE)
  #index of the 4th best team ("worst best")
  NL_threshold = NL_prob_ordered[4]
  AL_threshold = AL_prob_ordered[4]
  #predicted list of playoff teams
  NL_playoff_pred = ifelse(NL_playoff_prob < NL_playoff_prob[NL_threshold],0,1)
  AL_playoff_pred = ifelse(AL_playoff_prob < AL_playoff_prob[AL_threshold],0,1)
  #actual list of playoff teams
  NL.Playoffs.year=NL_data$Playoffs[!NL_train_set]
  AL.Playoffs.year=AL_data$Playoffs[!AL_train_set]
  #confusion matrix
  NL_matrix=table(NL_playoff_pred,NL.Playoffs.year)
  AL_matrix=table(AL_playoff_pred,AL.Playoffs.year)
  #percentage of incorrect predictions (error rate)
  NL_error3[Year-1995]=(NL_matrix[1,2]+NL_matrix[2,1])/length(NL_playoff_pred)
  AL_error3[Year-1995]=(AL_matrix[1,2]+AL_matrix[2,1])/length(AL_playoff_pred)
  #false positive rate
  NL_fp3[Year-1995]=NL_matrix[2,1]/(NL_matrix[2,1]+NL_matrix[1,1])
  AL_fp3[Year-1995]=AL_matrix[2,1]/(AL_matrix[2,1]+AL_matrix[1,1])
  #false negative rate
  NL_fn3[Year-1995]=NL_matrix[1,2]/(NL_matrix[2,2]+NL_matrix[1,2])
  AL_fn3[Year-1995]=AL_matrix[1,2]/(AL_matrix[2,2]+AL_matrix[1,2])
  #mean squared error
  NL_MSE3[Year-1995] = sum((NL.Playoffs.year-NL_playoff_pred)^2)/length(NL_playoff_pred)
  AL_MSE3[Year-1995] = sum((AL.Playoffs.year-AL_playoff_pred)^2)/length(AL_playoff_pred)
}

NL_avg_error3=mean(NL_error3)
AL_avg_error3=mean(AL_error3)
NL_avg_fp3=mean(NL_fp3)
NL_avg_fn3=mean(NL_fn3)
AL_avg_fp3=mean(AL_fp3)
AL_avg_fn3=mean(AL_fn3)
NL_avg_MSE3 = mean(NL_MSE3)
AL_avg_MSE3 = mean(AL_MSE3)

NL_avg_error3
```

```
## [1] 0.1194196
```

```
AL_avg_error3
```

```
## [1] 0.1160714
```

```
NL_avg_fp3
```

```
## [1] 0.08020833
```

```
NL_avg_fn3
```

```
## [1] 0.234375
```

```
AL_avg_fp3
```

```
## [1] 0.08125
```

```
AL_avg_fn3
```

```
## [1] 0.203125
```

```
NL_avg_MSE3
```

```
## [1] 0.1194196
```

```
AL_avg_MSE3
```

```
## [1] 0.1160714
```

Outcomes

We can see that splitting the data by league does not perform better out-of-sample. The model that performed best out-of-sample for the American League was the single logistic regression, then the single logistic regression with a dummy variable, and finally the separate logistic regressions. For the National League, all the models had the same out-of-sample performance.