Forestfires model - probability

Matthew Hurworth

29/01/2022

```
#Tells Rstudio where to find data and loads data
data<-read.csv("D:/Data analytics year/forestfires.csv")
summary(data)
#Transform data to binomial
data$area<-ifelse(data$area> 1,1,0)

install.packages("pRoc")
```

STR summary shows that months and days need factorising

```
#str summary
str(data)
```

```
#Turning months and days into factors
data$month<- as.factor(data$month)
data$day<- as.factor(data$day)
#data summary
summary(data)
```

Boxplots demonstrate the data contains many out liars in data set

```
#Area must be factorised for boxplots due to the binomial nature of the data

library(ggplot2)

areafactor<- as.factor(data$area)

boxplots<- function(x,y){ggplot(data) +aes_string(x,y)+geom_boxplot()}

library(purrr)

xvar<- c("areafactor")
yvar<- c("FFMC", "DMC", "DC", "ISI", "temp", "RH", "wind", "rain")


map2(xvar, yvar, boxplots)
```

Housekeeping of data

```
library(tidyverse)
#Set seed so methods can be reproduced
set.seed(42)
#Split data into test and train
x<-data[sample(1:nrow(data)),]
train<-x[1:380,]
test<-x[381:517,]
```

First model

```
#Produces first glm model with data
glm<- glm(area ~ 1, data=train, family=binomial)
summary(glm)
```

Use add1 function to determine first significant covariate with lowest AIC

```
#add1 similar to step model command compares covariates and shows significant ones. Comparison between
add1(glm, scope=train, test="Chisq")
```

Wind AIC = 522.08 so is added to model

```
glm2<- glm(area ~ wind , data=train, family=binomial)
summary(glm2)
```

Add1 repeated for second model

```
add1(glm2, scope=train, test = "Chisq")
```

DC is next lowest AIC that is significant (AIC - 518.47) so added to model

```
glm3<- glm(area ~ DC + wind, data=train, family=binomial)
summary(glm3)
```

Third model (glm3) is best - no more covariates to add

```
add1(glm3, scope=train, test="Chisq")
```

Model validation

D sq value is low - nearer 1 is best.

```
1-(glm3$deviance / glm3$null.deviance)
```

Prediction values show a weak model

```
#Using predictive function for train data to produce yes/no results over 0.5
trn_pr <- ifelse(predict(glm3, type="response", data = "train")> 0.5, "Yes", "No")
trn_pr
```

Confusion matrices show only ~50% accuracy of model

```
#Tabulating data for comparison of predicted vs actual - train data set
trn_tab <- table(predicted =trn_pr, actual=train$area)
trn_tab

#167 of 210 correctly predicted small, 57 of 170 correctly predicted large
```

```
#Tabulating data for comparison of predicted vs actual - test data set
tst_pred <- ifelse(predict(glm3, newdata=test, type="response")>0.5, "Yes","No")
tst_tab<- table(predicted = tst_pred, actual= test$area)
tst_tab

# 44 of 64 are correctly predicted small, 20 of 73 are correctly predicted large
```

MOdels are similar between test/train shows model works, but it is wrong nearly 50% of time

```
#Producing prediction probabilities for comparison of predicted vs actual visually via graphs.
pr <- predict(glm3, x, type="response")
head(round(pr,2))
```

Histogram adds evidence glm3 is not a great model - values 0 should all be on left side and 1 all on right, overlapping shows model is often incorrect.

```
hist(pr, breaks=20)

hist(pr[data$area==TRUE],col="red", breaks = 20, add=TRUE)
```

AUC value is 0.542 - values less than 0.75 demonstrate poor model accuracy.

```
#Test model Sensitivity vs specificity - value closer to 1 is better but its relative so we use to comp
#ROC showing AUC for glm 3

library("pROC")
test_prob<- predict(glm3, newdata=test, type= "response")
test_roc <- roc(test$area~test_prob, plot= TRUE, print.auc= TRUE)
```

glm3 is best model although it is a poor one.