GLM CA2

Statistics for Data Analytics

Aayush Sandeep Jain

10518910

20/12/2019

**Question 1**

**Consider a relational dataset and specify your input and output variables**

Dataset:<https://www.kaggle.com/jsphyg/weather-dataset-rattle-package/data>

My parents are on a vacation in Australia and it seems like it doesnt rain so frequently in there. One of the things I detest about Dublin is the unpredicted rainfall. Thus, when I saw this dataset on Kaggle it inspired me to select this dataset and to see what factors affect the rains in Australia.

Here, we select independent variables like temperature, sunshine, windspeed and evaporation rate. The independent variable is the factor RainToday which determines if it rained or not on that particular day.

Initial setup

getwd()

## [1] "C:/Users/hp/Desktop/Stats project/GLM"

setwd("C:/Users/hp/Desktop/Stats project/GLM")  
data<- na.omit(read.csv("weatherAUS.csv"))  
head(data)

## Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine  
## 5940 01/01/2009 Cobar 17.9 35.2 0 12.0 12.3  
## 5941 02/01/2009 Cobar 18.4 28.9 0 14.8 13.0  
## 5943 04/01/2009 Cobar 19.4 37.6 0 10.8 10.6  
## 5944 05/01/2009 Cobar 21.9 38.4 0 11.4 12.2  
## 5945 06/01/2009 Cobar 24.2 41.0 0 11.2 8.4  
## 5946 07/01/2009 Cobar 27.1 36.1 0 13.0 0.0  
## WindGustDir WindGustSpeed WindDir9am WindDir3pm WindSpeed9am  
## 5940 SSW 48 ENE SW 6  
## 5941 S 37 SSE SSE 19  
## 5943 NNE 46 NNE NNW 30  
## 5944 WNW 31 WNW WSW 6  
## 5945 WNW 35 NW WNW 17  
## 5946 N 43 N WNW 7  
## WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am  
## 5940 20 20 13 1006.3 1004.4 2  
## 5941 19 30 8 1012.9 1012.1 1  
## 5943 15 42 22 1012.3 1009.2 1  
## 5944 6 37 22 1012.7 1009.1 1  
## 5945 13 19 15 1010.7 1007.4 1  
## 5946 20 26 19 1007.7 1007.4 8  
## Cloud3pm Temp9am Temp3pm RainToday RISK\_MM RainTomorrow  
## 5940 5 26.6 33.4 0 0 No  
## 5941 1 20.3 27.0 0 0 No  
## 5943 6 28.7 34.9 0 0 No  
## 5944 5 29.1 35.6 0 0 No  
## 5945 6 33.6 37.6 0 0 No  
## 5946 8 30.7 34.3 0 0 No

1. **Train the model using 80% of this dataset and suggest an appropriate GLM to model ouput to input variables.**

Selecting relevant variables

x1=data$MinTemp  
x2=data$Evaporation  
x3=data$Sunshine  
x4=data$WindGustSpeed  
y=data$RainToday  
  
dataset<-data.frame(x1,x2,x3,x4,y)  
trainset.glm<-glm(y ~., data=dataset, family='binomial')  
summary(trainset.glm)

##   
## Call:  
## glm(formula = y ~ ., family = "binomial", data = dataset)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3073 -0.6743 -0.4325 -0.1853 6.3792   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.5913569 0.0464616 -34.25 <2e-16 \*\*\*  
## x1 0.0990292 0.0021956 45.10 <2e-16 \*\*\*  
## x2 -0.2991508 0.0054883 -54.51 <2e-16 \*\*\*  
## x3 -0.1510900 0.0031978 -47.25 <2e-16 \*\*\*  
## x4 0.0350026 0.0008256 42.40 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 59582 on 56419 degrees of freedom  
## Residual deviance: 48751 on 56415 degrees of freedom  
## AIC: 48761  
##   
## Number of Fisher Scoring iterations: 5

1. **Specify the significant variables on the output variable at the level of 𝛼=0.05 and explore the related hypotheses test. Estimate the parameters of your model.**

It’s seen that the selected variables are significant

set.seed(1390)  
n<-nrow(dataset)  
  
index<-sample(n,n\*(80/100))  
train<-dataset[index,]  
test<-dataset[-index,]  
  
trainset.glm<- glm(train$y ~., data = train, family='binomial')  
trainset.glm

##   
## Call: glm(formula = train$y ~ ., family = "binomial", data = train)  
##   
## Coefficients:  
## (Intercept) x1 x2 x3 x4   
## -1.59831 0.09981 -0.29470 -0.15385 0.03459   
##   
## Degrees of Freedom: 45135 Total (i.e. Null); 45131 Residual  
## Null Deviance: 47550   
## Residual Deviance: 38910 AIC: 38920

1. **Predict the output of the test dataset using the trained model. Provide the functional form of the optimal predictive model.**

Predicting the output on the test set

phat\_i=predict(trainset.glm , test, type='response')  
pred=rep(0, length(phat\_i))  
pred[phat\_i>=0.5]=1  
pred

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0  
## [35] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [69] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [103] 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0  
## [137] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [171] 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [205] 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0  
## [239] 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0

1. **Provide the confusion matrix and obtain the probability of correctness of predictions.**

Creating the confusion matrix

actual=test$y  
conf\_mat=table(pred,actual)  
conf\_mat

## actual  
## pred 0 1  
## 0 8340 1881  
## 1 405 658

Display accuracy

accuracy=mean(pred==actual)  
accuracy

## [1] 0.7974123

**Question 2**

![A close up of text on a white background

Description automatically generated]()

![A close up of text on a white background

Description automatically generated]()

![A close up of text on a white background

Description automatically generated]()

![A close up of text on a white background

Description automatically generated]()

**Question 3**

**![A close up of text on a white background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**R-Solution:**

data = data.frame("Yes"=c(200,250),

"No"=c(150,300),

"Can't Say"= c(50,50))

chisq.test(data)

**Output:**

Pearson's Chi-squared test

data: data

X-squared = 16.204, df = 2, p-value = 0.000303