Logit Model

Gus Vu

2023-05-22

# Testing / Training Split.

The next step is to split our data into an 80/20 training testing split. In other words we will train a model on 80% of our data and test how accurate our model is on the remaining 20%. We do this because we want to avoid over fitting our model. If we trained and tested on the same data set we would inflate our accuracy. We have a split beacause we want to test our model on new data. We want to test our model on data that it has not seen before.

library(caTools)

## Warning: package 'caTools' was built under R version 4.2.3

set.seed(123)  
data\_sample = sample.split(NewData$Class,SplitRatio=0.80)  
train\_data = subset(NewData,data\_sample==TRUE)  
test\_data = subset(NewData,data\_sample==FALSE)

# Fitting Logistic Regression Model

A logistic regression model is used to model 2 outcomes success (fraud) or fail (not fraud).

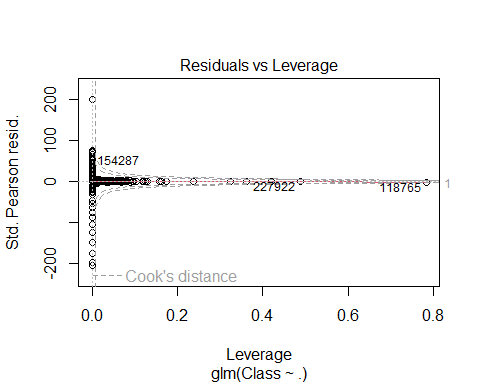
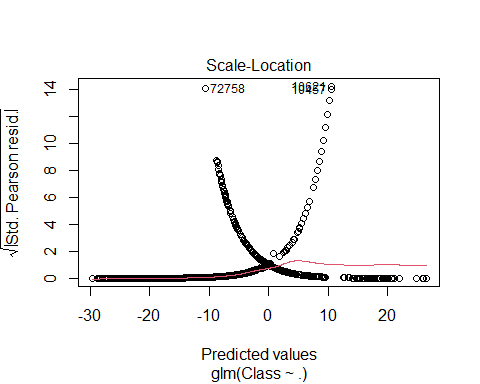
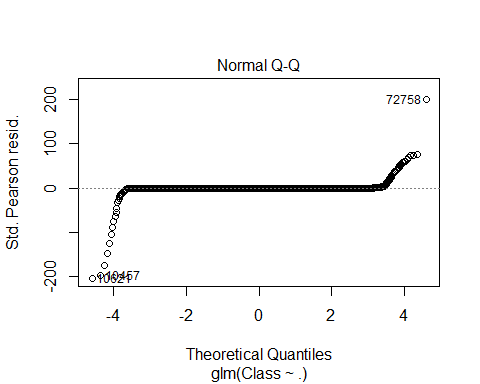
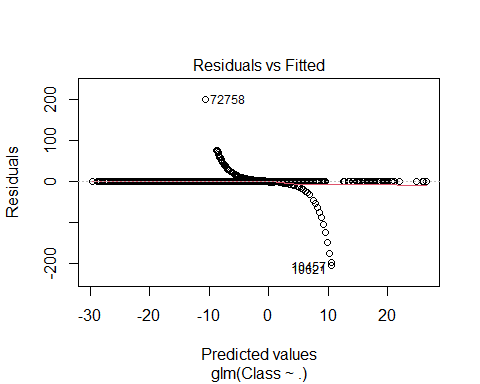
* Note we must be careful when interpreting this model as a logit model predicts log odds of success. We need to do a transformation if we want to get probabilities.

Logistic\_Model=glm(Class~.,train\_data,family=binomial())  
summary(Logistic\_Model)

##   
## Call:  
## glm(formula = Class ~ ., family = binomial(), data = train\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.6108 -0.0292 -0.0194 -0.0125 4.6021   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.651305 0.160212 -53.999 < 2e-16 \*\*\*  
## V1 0.072540 0.044144 1.643 0.100332   
## V2 0.014818 0.059777 0.248 0.804220   
## V3 0.026109 0.049776 0.525 0.599906   
## V4 0.681286 0.078071 8.726 < 2e-16 \*\*\*  
## V5 0.087938 0.071553 1.229 0.219079   
## V6 -0.148083 0.085192 -1.738 0.082170 .   
## V7 -0.117344 0.068940 -1.702 0.088731 .   
## V8 -0.146045 0.035667 -4.095 4.23e-05 \*\*\*  
## V9 -0.339828 0.117595 -2.890 0.003855 \*\*   
## V10 -0.785462 0.098486 -7.975 1.52e-15 \*\*\*  
## V11 0.001492 0.085147 0.018 0.986017   
## V12 0.087106 0.094869 0.918 0.358532   
## V13 -0.343792 0.092381 -3.721 0.000198 \*\*\*  
## V14 -0.526828 0.067084 -7.853 4.05e-15 \*\*\*  
## V15 -0.095471 0.094037 -1.015 0.309991   
## V16 -0.130225 0.138629 -0.939 0.347537   
## V17 0.032463 0.074471 0.436 0.662900   
## V18 -0.100964 0.140985 -0.716 0.473909   
## V19 0.083711 0.105134 0.796 0.425897   
## V20 -0.463946 0.081871 -5.667 1.46e-08 \*\*\*  
## V21 0.381206 0.065880 5.786 7.19e-09 \*\*\*  
## V22 0.610874 0.142086 4.299 1.71e-05 \*\*\*  
## V23 -0.071406 0.058799 -1.214 0.224589   
## V24 0.255791 0.170568 1.500 0.133706   
## V25 -0.073956 0.142634 -0.519 0.604109   
## V26 0.120841 0.202553 0.597 0.550782   
## V27 -0.852018 0.118391 -7.197 6.17e-13 \*\*\*  
## V28 -0.323854 0.090075 -3.595 0.000324 \*\*\*  
## Amount 0.292477 0.092075 3.177 0.001491 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5799.1 on 227845 degrees of freedom  
## Residual deviance: 1790.9 on 227816 degrees of freedom  
## AIC: 1850.9  
##   
## Number of Fisher Scoring iterations: 12

Lets visualize our model

plot(Logistic\_Model)



Our residual vs Fitted plot looks good. There is a little deviation but the residuals are more or less distributed around 0. We have a few outliers but this is ok since our data is large. Our normal QQ plot looks good also, our data more or less follows a straight line but has small deviation at the end points. Our scale location plot shows a relatively horizontal red line, this means that we have more or less equal variance.

Lastly Our cooks plot shows that we have only a few highly influential points / outliers. This is more or less neglagable since our data is so large.

Lets look at an ROC curve. Also known as a Receiver Optimistic Characteristics. This plots the sensitivity and the specificity . The red line represents a random choince. Since there is only 2 possible outcomes heads or tails. We want the area under our ROC curve (the blue line) to be greater then the area under the red line.

library(pROC)

## Warning: package 'pROC' was built under R version 4.2.2

## Type 'citation("pROC")' for a citation.

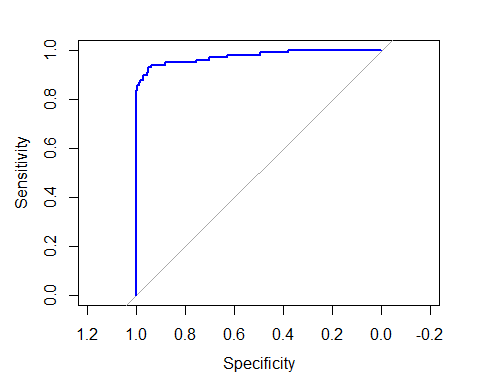
##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

lr.predict <- predict(Logistic\_Model,test\_data, probability = TRUE)  
auc.gbm = roc(test\_data$Class, lr.predict, plot = TRUE, col = "blue")

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases



auc.gbm$auc

## Area under the curve: 0.9748

We observe that the area under the curve is .9748, this is much better than random choice of 0.5.