How to Interpret Model Fit and Performance Information in R

Let’s consider the last model we ran, which had age, cholesterol and insurance. How well does this model fit the data? We’ll start with what R gives us by default and decide what’s useful and what’s missing. R’s default output for the “glm” command includes the following:

* The call to the algorithm itself, i.e. what the model is
* Deviance residuals
* Coefficients (we’ve covered these already)
* Dispersion parameter
* Null deviance
* Residual deviance
* AIC
* Number of Fisher scoring iterations

The last of these isn’t terribly interesting except to note that it means that the model has converged, i.e. the algorithm has worked and found the best solution. In the others, there’s quite a bit of mention here about “deviance”, which was described in the earlier reading. The null deviance tells us about model fit with just the intercept term in. What’s more important is the deviance when we’ve added our predictors and how much the deviance falls when we do so compared with the null value. For our model:

* The null model had a deviance of **334.54 on 388 degrees of freedom**
* Our model had a residual deviance of **289.28 on 384 degrees of freedom**

That’s a difference of 334.54-289.28 = 45.26 at a “cost” of 388-384 = 4 degrees of freedom. If you recall the concept of degrees of freedom from the previous course, then you’ll see that the four d.f. represent four added parameters, which came from one for age, one for cholesterol and two for insurance. Our model has improved by 45.26 for our “investment” of 4 d.f., but is that a good return on our investment?

To answer this, we need to define “deviance” in this context. This can get very technical, as there are different ways to compute this. First, the bigger the deviance, the worse the model fits the data, so you want to be able to test this. Second, we want our model to be an improvement on the null model – if you have at least one variable with a low p value, then you’ll have an improvement.

**Null Deviance and Residual Deviance**

**To understand residual deviance, we must first think about 3 models: the null model, the proposed model and the saturated model.** The null model, as discussed earlier, is one where we only include the intercept: this model therefore only has one parameter. The proposed model is the model with the variables we included in our logistic regression. The number of parameters in the proposed model is the number of variables plus one (the intercept): this is because each of the variables we have included only needs one parameter, but remember that a categorical variable with three categories, for example, will need two parameters. The saturated model is a model which fits the data perfectly, because it has as many parameters as there are data points.

**The null deviance is a measure of how well the null model explains the data compared with the saturated model.** Having just one parameter, the null model does not usually explain the data very well, and this is indicated by a large null deviance. The point of doing a regression model is that we reckon we can do better at explaining the data with a few variables (age, sex, etc). This brings us on to the residual deviance: how well the proposed model explains the data compared with the saturated model.

**The difference between the null deviance and the residual deviance gives us an idea of how well our model has performed (at the cost of degrees of freedom).** It’s like a return on investment: how much benefit (explanation of the variation in the outcome) do we get for our investment (the variables we’ve added or, more accurately, the degrees of freedom taken up by those variables). If the model is “good”, then the difference between the null deviance and the residual deviance will be large. There are formal ways to see whether the difference is large enough.

Let’s demonstrate this in R:

# design your logistic regression

full\_model <- glm(dm ~ age + chol + insurance, family = binomial(link = logit))

# analyse table of deviance

anova(full\_model, test = "Chisq")

## Analysis of Deviance Table

##

## Model: binomial, link: logit

##

## Response: dm

##

## Terms added sequentially (first to last)

##

##

##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)

## NULL                     388     334.54

## age      1   35.413      387     299.12 2.667e-09 \*\*\*

## chol     1   7.363       386     291.76  0.006658 \*\*

## insurance  2 2.478       384     289.28  0.289613

## ---

## Signif. codes:  0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The fourth column shows the deviances of the models compared with the saturated model. The first (334.54) is the null deviance and each subsequent number is the deviance of the model with each new variable. The final value (289.28) is the deviance of the proposed model (with all three of our variables). This is the residual deviance. As expected, adding each new variable to our model explains the data better, thus reducing the deviance.

**To test whether each added parameter increases the deviance by a significant amount, we asked R to compare it with a chi-square value for the number of degrees of freedom lost.** If the p-value is low, it indicates that the corresponding added variable causes a significant change in deviance, and thus is a better fitting model. It’s not at all essential that you understand why we use the chi-square distribution for this comparison – just that you know how to interpret the resulting p-value.

In our case, adding the variables age and cholesterol significantly reduce the deviance and improve the model fit, as indicated by their low p-values, but including the insurance variable does not improve the model fit enough to justify the loss in degrees of freedom, as indicated by its high p-value of 0.2896.

**AIC**

Lastly, I’ll mention the AIC. **This is short for Akaike Information Criterion and measures the quality of a model in terms of the amount of information lost by that model.** It therefore recognises that all models lose information compared with “reality” but some models lose less than others. It’s of no use by itself but is used for comparing two or more models. Small AIC values are best.