IMDtotheB

Abstract

(Summary of what we wanted to do, and what we've done. To be written later…)

This is essentially Part 6 of the coursework spec.

1. Data exploration

The data are provided in a data frame, `imdb`, which contains information about a subset of 3638 films from the IMDb database.

We are ultimately interested in the relationship between the IMDb score (recorded as `score`) and the rest of the information which is available.

Looking at the distribution of scores we can see the mean rating of the films in our dataset is between 6 and 7; this is a consequence of our tendency to treat 6 or 7 as an "average" film rating, rather than 5 (which would arguably be more logical from a mathematician's point of view!).

It will also be useful to see the relationship between the score and each of the predictors which might be used in our models. *The code used to produce the subsequent plots can be found in Appendix A.*

1.1 Replacing incorrect values

One seemingly erroneous point is immediately noticeable: the aspect value of 16, which lies well out to the right of the other points. We can easily find which film it belongs to.

Consulting the entry for this film in the online IMDb database, it turns out that the aspect should in fact be 16:9, so we shall amend this.

Let's have a look at a couple more plots which might reveal potentially interesting relationships among the predictors.

There is a smattering of outlying values in each plot. Further investigation reveals that one of these, the clear outlier in the bottom-right of the Gross vs Budget plot, is in fact another misentered value.

Again consulting the online database, it seems the budget for this film was actually $85m, not $390m as currently recorded. This is easily changed.

1.2 Duplicated entries

There are also multiple titles which appear more than once in the dataset.

We now have a new data frame, `imdc`, which is identical to `imdb` but with each title now only appearing once. All the following analyses and models will be based on this tidied dataset.

2. Initial model

We are now in a position to begin training linear models on the data. Initially we will train a model on the full set of predictor variables.

Immediately we notice that the factor variables `country` and `rating` are each expanded into many dummy variables by `lm()`, and that many of these new dummy predictors appear to contribute very little. Moreover, the fairly poor adjusted R^2 score indicates that the model is currently not explaining much of the variance in the data.

Before continuing, it would be wise to check the diagnostic plots of the model.

The issues with our model become more obvious. The negative trend in the residuals-fitted plot shows that the model predicts high scores too generously. The spread of variance in residuals is also very uneven. This is largely a consequence of the concentration of film scores at approximately 6: there are a great deal more points there, so we are likely to see a higher variance. Nonetheless we may be able to reduce the severity of the issue.

The Q-Q plot of standardised residuals also reveals that the distribution of residuals is most certainly not normal. Considering the fact that normally-distributed residuals form important underlying assumption of the entire linear modelling process, this is something we must try to remedy. Additionally, the residuals-leverage plot reveals multiple points with unusually high leverage.

Working from this base model we can now begin to take steps to improve the model's performance.

2.1 Transforming numeric predictors

By its very design, a linear model will perform better the closer to linear the relationship between each numeric predictor and the response. However, it appears that the relationship between some of the predictors and the score is **not** linear; and therefore we may be able to improve the performance of the model by somehow transforming the data.

The decision of which transformations to apply is largely an empirical process. Given that the model is consistently overpredicting high scores, we try squaring the response variable to encourage the model to correct this.

We also apply square root transformations to some of the predictors. For count variables such as `uservotes` this choice can be explained, at least in part, by the fact that counts are often approximately Poisson distributed, and square-rooting is an effective method of improving the symmetry of this distribution. For non-count variables such as `duration`, square-rooting simply improves the fit of the resulting model.

The following plots show linear regression lines for the original and transformed predictors and response.

Now we can compare the performance of linear models fitted on the untransformed and transformed numerical predictors.

We can also compare the diagnostic plots of the new model with those of the base model.

The new model shows significant improvements. We still have a slight negative trend in the residuals-fitted plot but the effect is markedly less pronounced (the issue with spread of variance remains, but it is all but impossible to fix this for the reasons described earlier). The Q-Q plot shows that the model's residuals are now distributed much closer to normally.

The high-leverage points have also disappeared, although as we will see shortly that this is due to the omission of factor predictors in this model.

There are two outliers, particularly evident in Q-Q plot: rows 1415 and 3475.

Justin Bieber: Never Say Never makes an appearance, along with the marvellous documentary Winged Migration - at opposite ends of the spectrum in terms of score. In particular, according to our model the former should have a much higher score than it actually does. Given that the featured celebrity tends to divide opinion (one of this report's authors created an IMDb account with the sole intention of giving that particular film a score of 1), it is perhaps not surprising that our model is overly sympathetic in its predicted score.

Let's fit our model with these two points omitted, and look at the summary.

This has a better adjusted R^2 than both `tslmod` suggesting it was sensible to remove the outlying points. It also has a better adjusted R^2 than `fullmod`, despite the greater amount of predictors and therefore information `fullmod` was trained on. This is no doubt due to the huge expansion of the factor variables. We will address this issue in Section 3.

2.3 Interactions

Given the large number of predictors, there are an almighty number of possible interactions whc=ich could be added to the model. First let's check for strongly-correlated predictors. The upper triangle of the following figure is the correlation matrix of the numeric predictors; the lower half is scatterplots of each pair; and the diagonal shows histograms of each predictor.

We will add the five strongest correlations as interactions in our model. Three of these are obvious linear relationships between user and critic votes and reviews. The others can be easily interpreted in the context of film: high-budget films are often well-advertised, leading to large gross takings at the box office and concurrently many votes from filmgoers.

2.4 Re-fitting the full model

We now re-fit `fullmod` by returning the factor variables to `tsdf`, adding the interaction predictors and removing the two outlying points we identified earlier.

The adjusted R^2 score is still low, but is significantly better than the first time we fitted the model. Of course, the score a given film receives from a user is highly subjective; for such a difficult task, the adjusted R^2 is not terrible. In the following section we will aim to further improve and simplify the model.

3. Simplifying the initial model

Due to the inclusion of many-levelled factor predictors, the current model contains a very large number of predictors. Many of these are likely to be statistically insignificant.

Moreover, on closer inspection many levels contain only one data point: `country` is perhaps the most extreme example.

Hence many predictors only apply to a single point. This is virtually the textbook definition of overfitting, which will lead to poor results when the model is used to predict unseen scores. Certainly we must ensure that these one-point predictors at the very least are removed from any simplified models.

3.1 Selecting smaller models with AIC

Given that we currently have over 50 predictors in the model, a “top-down” `step()`-style approach is unlikely to reduce our model to a satisfactory size. Moreover, `step()` keeps staunchly to its numeric-assessment-based approach, and therefore does a poor job of recognising the endemic overfitting within our model and removing the offending predictors.

Therefore we adopt a somewhat more natural approach, founded on the principle of Occam's Razor, whereby we will add predictors to an initially empty model. The `regsubsets()` function in the `leaps` package allows us to find the “best” model (according to Akaike's An Information Criterion, or AIC) of each size. We find such models up to an arbitrary limit of 25 predictors - a generous limit given that we wish to find a “simple” model.

We can now calculate the AIC of each of these models according to the formula

$$-2\{\text{maximised log-likelihood}\} + 2\{\text{number of parameters, including intercept}\} + (\text{constant})$$

which for linear models is equivalent to

$$n \log \left(\frac{RSS}{n} \right) + 2\{\text{number of parameters, including intercept}\} + (\text{constant})$$

We can subsequently plot these AIC scores against model size.

Notice that over this range of sizes, adding predictors consistently reduces (improves) the AIC score of the model - `step()` would have faltered long before reaching this stage. Therefore this is not a simple case of choosing the model with the lowest AIC and we shall have to be slightly more subjective in our selection of a smaller model.

We take the approach that two distinct "elbows" can be seen in the plot, at 3 and (less markedly) 9 predictors. After each of these points the rate of improvement of the model with the addition of each extra predictor becomes smaller. Again abiding by Occam's Razor we select these model sizes to study in more detail.

Firstly we consider the "reduced" model with 9 predictors.

Notice that this model performs almost as well as `fullmod` did earlier, according to adjusted R^2. Aditionally the troublesome one-point predictors have all been eliminated.

Let's also inspect the "tiny" model with only 3 predictors.

Observe that `rmod` has an AIC score very close to `fullmod`, but uses approximately six times fewer predictors and avoids the overfitting issues we saw earlier. In addition `tmod` performs almost as well using only 3 predictors.

3.2 Manually "collapsing" factor predictors

We can take a different approach to reduce the number of levels in factor predictors which involves manually grouping similar levels - for example, grouping ratings into "kid-friendly" and "not kid-friendly". Details can be found in Appendix B, but seeing as we ultimately did not manage to make significant improvements to the models from Section 3.1 we do not include the results here.

4. Interpreting the reduced model

5. Further analysis: [Witty and amusing question to be inserted here]

Appendix A: Plots of score vs predictors

Appendix B: Grouping similar factor levels