Logistic Regression: Speed Dating

Signal Data Science

We'll introduce logistic regression by returning to the Columbia speed dating dataset.

Unregularized logistic regression

First, we'll see how to use unregularized logistic regression.

Loading the data

The dataset you'll be using in this assignment is an aggregated form of the full speed dating dataset; you've worked with a simplified form of this dataset before (with fewer variables). Refer to the documentation in `speeddating-documentation.txt` for a description of the new variables.

- Use `read.csv()` to load `speeddating-aggregated.csv` in the `speed-dating` dataset.
- Use `complete.cases()` to determine the number and proportion of rows in the data with `NA`s. Clean the data by using `na.omit()` to remove all rows with `NA`s.

Using `glm()`

You can run logistic regression with `glm()`. It can be used in the same fashion as `lm()`, except for logistic regression you must pass in the additional parameter `family="binomial"`. Additionally, the column representing the binary class which you want to predict must either be (1) a numeric column taking on values 0 and 1 or (2) a factor.

The `pROC` package provides a function, `roc()`, which plots the receiver operating characteristic (ROC) curve given the results of a logistic regression fit. The output of `roc()` can be passed into `plot()` or directly printed to display the area under the ROC. Note that `roc()` accepts *probabilities* as inputs, but the

predictions made with a logistic regression model will be in the form of *log-odds ratios*, which must be converted into probabilities with

$$P = \frac{\exp L}{1 + \exp L}$$

where L is a log-odds ratio and P is the corresponding probability.

When working through the following questions, examine and interpret the coefficients of each logistic regression model. In addition, examine the area under the ROC curve as well as the shape of the ROC curve itself.

- Predict gender in terms of the 17 self-rated activity participation variables.
- Restrict to the subset of participants who indicated career code 2 (academia) or 7 (business / finance). Predict membership in either class in terms of the 17 activities.
- Restrict to the subset of participants who indicated being Caucasian (`race == 2`) or Asian (`race == 4`). Predict membership in either class in terms of the 17 activities.

Regularized linear regression

We can also use *regularization* with logistic regression via the `glmnet` package. Like `glm()`, the only difference with linear regression is that we need to pass in the `family="binomial"` parameter. Unlike `glm()`, `glmnet()` and `cv.glmnet()` will only accept a binary variable (taking on values 0 and 1) and *not* a factor for the target variable.

With regularization, we have the freedom to throw a lot of features into our model, because the regularization parameter will be chosen such that only the important ones remain. In particular, L^1 regularization will produce a very interpretable model, because the coefficients of the less important variables will be driven to 0.

Our goal will be to distinguish between Caucasians and Asians by using regularized logistic regression. As predictors, we'll include the 17 activites *and* every possible 2nd-order interaction term between the 17 activities.

- Restrict to the subset of Caucasian or Asian participants. From that subset, create a new data frame, `df_activities`, with just the 17 activity variables.
- To form the 2nd-order interaction terms, pass in `df_activities` to `model.matrix()` along with the formula `~ .*. + 0`. Store the output in a variable `cross_terms`. (In the formula, `.*.` indicates that every possible

interaction term should be formed and `+ 0` indicates that no intercept column should be created.)

- Scale `cross_terms` and use it with `cv.glmnet()` to train a L^1 regularized logistic regression model distinguishing between Caucasians and Asians. Access the coefficients of the optimal model with `coef()` (you'll have to pass in the value of λ to use into the `s` parameter) and print out the nonzero entries. Interpret the results.
- Compare the AUC of the regularized model with cross terms against the AUC of the unregularized model trained only against the 17 activities.