Universität Salzburg
Ferndinand Ferber, BSc.
Univ.-Prof. Dr. Wolfgang Trutschnig

Salzburg, May 2024
SS 23/24

Exercise Sheet 08 @'Applied AI Using R'

We are working with logistic regressions. As mentioned last time, exercises like these do prepare you for participating in projects and for internships in companies.

Exercise 33.
In this exercise we want to gain a better feeling for odds and logistic classification. In one study (Guéguen and Jacob 2014, see below), researchers measured the effect of waitresses wearing red clothing on the amount of tips received by male guests in five restaurants.

1. The article is open access. Get the following information: How many men have tipped a waitress with a red top? How many men have not tipped a waitress with a red top? How many men have tipped/not tipped a waitress wearing some other color?

2. What is the probability of a man tipping a waitress with a red top? What are the odds of a men tipping a waitress with a non-red top?

3. Write down the formula for the logistic regression model. We want to predict if a man tipped (encoded as a one) or not (encoded as a zero), based on whether the waitress had a red top (encoded as a one) or not (encoded as a zero).

4. Calculate the model coefficients (intercept and slope) for the given dataset.

5. Re-arrange the formula so that it outputs the odds (and not the logits) and plug-in the coefficients

6. Are the model coefficients aligned with the result of the study?

References

[1] Guéguen, Nicolas, and Céline Jacob. 2014. "Clothing Color and Tipping: Gentlemen Patrons Give More Tips to Waitresses with Red Clothes." *Journal of Hospitality & Tourism Research* 38 (2): 275–80. `https://doi.org/10.1177/1096348012442546`.

Exercise 34.
Remember how in Aufgabe 27 we checked if `lm` really detects the `true` coefficients. Perform the same analysis for the logistic regression, i.e., fix parameters, choose a simple logistic regression model with these parameters, generate data from the model, estimate the parameters, and check, if the estimations are close to the true values; repeat at least a 100 times and check if the estimates get better with increasing sample size.

Exercise 35.
We consider weather data from Australia and want to predict the `RainTomorrow` variable with a logistic classification model.

1. Download the *rain in Australia* dataset from `https://ufile.io/noxjbosy` and parse it into `R`
2. Drop all rows that contain at least one `NA` value. Transform the `RainTomorrow` column into a factorial column. Filter the dataset on one location of your liking (e.g. `Mildura` or `Woomera`). Remove the `Location` and `Date` columns afterwards.
3. Define a `{tidymodels}` preprocessor that transforms all character columns into factorial columns and uses a one-hot-encoding for all factorial columns.
4. Train a logistic classification model to predict `RainTomorrow` from the other (preprocessed) variables.
5. Either use `extract_fit_engine()` and `summary()` or use the `tidy()` function on the fitted workflow to extract the model coefficients.
6. Interpret the coefficient for the `Pressure3pm` variable in terms of log-odds and odds

Exercise 36.
After solving the previous exercise we might wonder if different variables are important for different locations. Maybe to predict rainfall in one location we need the wind direction, while this is irrelevant for other locations. In this exercise we therefore build a model for each location and explore how model coefficients and p-values are changing. This is also a good way to utilize our functional programming skills.

1. Use the same dataset as in the previous exercise, but train a logistic classification model on every location separately. You should end up with one model per location. Hint: Use `{dpylr}` verbs like `nest()` and `map()`
2. For how many models is the coefficient for the variable `Pressure3pm` significantly non-zero? Hint: Use `map()` and `tidy()`.
3. Compute the variance for each estimated coefficient (across all models)
4. Find a way to visualize the relationship between the amount of sunshine and whether it will rain tomorrow using `{ggplot2}`

Exercise 37 (This exercise is voluntary - you can score extra points)**.**
This exercise explores the intruder detection problem, which is an interesting combination of data analysis and behavioral psychology. Yandex for example solves the mailbox intruder detection problem based on behavior patterns of the users, like the time when mails get deleted (right after read or once in a while in bulk) or even different cursor movements. In this dataset we use at the browsing history of Alice to detect if she or

some intruder used her computer. The variable `target` will be 0 if Alice was operating and 1 if some intruder was controlling the browser. This exercise requires some advanced data wrangling skills to compute the actual features for the classifier.

1. Download the *Alice and the Intruder dataset* from `https://ufile.io/613owlwx` and parse it into `R`

2. Bring the dataframe into the following format:

```
# A tibble: 19,486 x 4
  session_id target index time
       <dbl>  <dbl> <chr> <dttm>
1        149      0 1     2013-11-19 15:23:04
2        149      0 2     2013-11-19 15:23:05
3        149      0 3     2013-11-19 15:23:05
4        149      0 4     2013-11-19 15:23:06
5        149      0 5     2013-11-19 15:23:06
# i 19,481 more rows
```

3. Add a column that contains the end time of each site visit. Hint: Use `lead()`. Calculate the time spent on each site in each session.

4. Bring the resulting dataframe into the following format:

```
# A tibble: 2,000 x 12
# Groups:   session_id [2,000]
  session_id target time_spent_1 time_spent_2 time_spent_3 time_spent_4
       <dbl>  <dbl> <drtn>       <drtn>       <drtn>       <drtn>
1        149      0  1 secs      0 secs       1 secs       0 secs
2        201      0 52 secs      9 secs       3 secs       4 secs
3        287      1  0 secs      0 secs       0 secs       1 secs
4        351      0  1 secs      1 secs       0 secs       2 secs
5        430      1 85 secs      3 secs       1 secs       0 secs
# i 1,995 more rows
# i 6 more variables: time_spent_5 <drtn>, time_spent_6 <drtn>,
#   time_spent_7 <drtn>, time_spent_8 <drtn>, time_spent_9 <drtn>,
#   time_spent_10 <drtn>
```

5. Remove the `session_id` column and use only sessions with at least five visited sites. Train a logistic classification model on the column `target` with predictor variables `time_spent_1` to `time_spent_5`.

6. Compute the `roc_auc()` measure on the same dataset. Would you trust this intrusion detector?