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

Salzburg, June 2024
SS 23/24

Exercise Sheet 10 @'Applied AI Using R'

This week's sheet is almost a hackathon. We will use decision trees and random forecasts with increasing complexity to forecast withdrawn daily amounts at an ATM.
In the last step you should come up for our own optimal forecasting approach (tree/random forest/lm, etc.) - the performance of your approach will then be tested on a new ATM next week; the one of you whose forecasts is the best (in terms of mean absolute error MAE) scores an extra 5 points and qualifies for a (paid) IDA-Lab internship[i].

Exercise 40.
We work with the ATM dataset already discussed in `07_Sprout` (Slides and R-Snippets), proceed as sketched on page 44 in the slides, and only use the following features as predictors: `weekday, day.of.month, month, festivity`[ii], `pre-festivity`[iii]

1. Use the data from 2008-01-01 till 2009-01-11 to train a model for predicting $d = 1$ day ahead.
2. Predict the withdrawn amount on 2009-01-12.
3. Use the data from 2008-01-01 till 2009-01-12 to train a model.
4. Predict the withdrawn amount on 2009-01-13.
5. ...proceed analogously till forecasting the value for 2009-12-31.
6. Calculate the MAE (mean absolute error) of all 353 forecasts.

Exercise 41.
Use the same predictors as in the previous exercise but this time forecast $d = 7$ days ahead:

1. Use the data from 2008-01-01 till 2009-01-11 to train a model for predicting $d = 7$ day ahead.
2. Predict the withdrawn amount on 2009-01-18.
3. Use the data from 2008-01-01 till 2009-01-12 to train a model.
4. Predict the withdrawn amount on 2009-01-19.
5. ...proceed analogously forecasting the value for 2009-12-31.
6. Calculate the MAE (mean absolute error) of all 346 forecasts.

[i] if desired
[ii] yes/no; 1 means 'yes', all other values 'no'
[iii] yes/no; 1.5 means 'yes', all other values 'no'

Exercise 42.
Notice that in the previous two exercises we completely ignored the withdrawn amounts of the past couple of days. Obviously the forecasts should improve if we include this information. Build a model using the following extended set of features: `weekday`, `day.of.month`, `month`, `festivity`, `pre-festivity`, `sum.out.m0`, `sum.out.m1`, `sum.out.m2`, `sum.out.m3`[iv], `sum.out.m4`, `sum.out.m5`,`sum.out.m6`,`sum.out.m7` whereby, e.g., the feature `sum.out.m6` is sum_out 6 days ago.

1. Use the data from 2008-01-01 till 2009-01-11 to train the model for predicting $d = 1$ day ahead. Notice that in this case `sum.out.m0` is the withdrawn amount on day 2009-01-11, `sum.out.m1` is the withdrawn amount on day 2009-01-10, `sum.out.m2` is the withdrawn amount on day 2009-01-09, and so on. Use the learned dplyr tools to do the required data wrangling.
2. Predict the withdrawn amount on 2009-01-12.
3. Use the data from 2008-01-01 till 2009-01-12 to train the model.
4. Predict the withdrawn amount on 2009-01-13.
5. ...proceed analogously till forecasting the value for 2009-12-31.
6. Calculate the MAE (mean absolute error) of all 353 forecasts.

Exercise 43.
Use the same predictors as in the previous exercise but this time forecast $d = 7$ days ahead:

1. Use the data from 2008-01-01 till 2009-01-11 to train a model for predicting $d = 7$ day ahead.
2. Predict the withdrawn amount on 2009-01-18
3. Use the data from 2008-01-01 till 2009-01-12 to train a model.
4. Predict the withdrawn amount on 2009-01-19
5. ...proceed analogously forecasting the value for 2009-12-31.
6. Calculate the MAE (mean absolute error) of all 346 forecasts.

Exercise 44.
We finally arrive at the hackathon exercise. Think through what other features might improve the forecast quality for the situation $d = 7$ (including more days in the past? aggregating prior days to weeks and using the weekly withdrawn amounts as predictor, etc.). Give free rein to your creativity and calculate the MAE for your improved model.

[iv] the 'm' stands for minus, i.e., in this case minus 3 days