Prediction of length-of-stay in an ICU bed

Note: for now I used 100% random information of length-of-stay

Background

At admission of a patient to the ICU (an ICU bed) we want to predict how long they will stay in the bed. The outcome we will look is therefore length-of-stay in an ICU bed. We want to base our predictions on some baseline characteristics of the patient (at admission), hospital-level-data, and bed-level-data.

Data

Mock data structure explained

Python was used to build a synthetic data set. The sample size is 9773.

Predictors

After a literature review the following potential predictors were retrieved:

- tba
- tba

Training and Testing

To evaluate the models and in the long-term maybe send alerts when the model is not performing anymore, the data is split into a train and a testing sample. The training data is composed of 75% of the sample. Note that in the long term this should be changed to cross-validation (*e.g* splitting the data several times).

```
# Split into train and test
set.seed(1)
train_rows <- sample(1:n_sample, size = .75*n_sample)
train_patients <- patients[train_rows, ]
test_patients <- patients[-train_rows, ]
n_train <- nrow(train_patients)
n_test <- nrow(test_patients)
```

Description of potential models

Once the product is running, data is collected and lessons have been learned, more sophisticated models can be looked at (*e.g* machine learning). For the time-being and for a first prototype simple parametric regression models should be used.

Simple poisson regression model

Since we want to predict the number of days a patient stays in the bed, this can be resolved with a simple count-data problem.

```
poisson_regression <-
  glm(randomly_allocate_a_length_of_stay ~ Gender + Age + Ort +
           zertifizierte..Betten,
      family = "poisson",data = train_patients)
```

Negative-binomial regression model

An alternative to poisson regression are negative-binomial models.

```
negbin_regression <-
  glm.nb(randomly_allocate_a_length_of_stay ~ Gender + Age + Ort +
           zertifizierte..Betten,
         data = train_patients)
```

Tree-based solutions

More non-parametric models should be looked at, once more data has been collected.

Model performance and variable selection on the mock-data

To compare the model performances, let us use the root-mean-square error (RMSE). The RMSE is commonly used to measure differences between values predicted by a model and the values observed, where a lower score implies better accuracy.

This model validation can also be used to update the models, add or drop variables, by judging whether they have had any predictive power.

Let's see how well that first poisson model performs:

```
predict_los <- predict(poisson_regression, newdata = test_patients,
                       type = "response")
observed_los <- test_patients$randomly_allocate_a_length_of_stay

RMSE <- sqrt(sum((observed_los - predict_los)**2)/n_test)
RMSE
```

```
## [1] 2.614144
```