# CAS ADS 2021
# Module 5

# Peer Review Project

**Jana Frangi**
**Maria Mumtaz**

## Learnings:

The most important learnings were the step-by-step repetition and that we were able to clarify misunderstandings as a result. Things discussed include:

1. **Data cleaning:**

We first looked into cleaning the data so that all patient data with missing values are thrown out. This runs the risk of losing too much data. However, this way was chosen also for training purposes and to have fully cleaned data. Missing values in multiple rows can affect the model performance and could lead to over estimation of the model. In general, the clinical relevance of dealing with missing values was discussed and different ways of cleaning of the data defined.

2. **Comparison of different models to reduce RMSE:**

We then looked into different models to find an optimized approach using iterative analysis. We used Logistic Regression and the Random Forest models for our classification problem. RMSE showed that the Random Forest would be a good tool for our data (RMSE of 0.17 and Log. Reg 0.53).

3. **Feature reduction:**

We then decided on the strongest ten variables based on the plot to apply the model again. It turned out that it did not improve the performance of the model (RMSE 0.17).

4. **Functions:**

New was that by defining of a function in advance (see below), you are able to sort of "re-call" this function during the training of the models.

```
def train_model(data, label, algo):
    """
    data: training data
    label: response variable

    return: fitted model and Root Mean Square Error
    """

    model = algo.fit(data, label)
    return model, np.sqrt(mean_squared_error(label, model.predict(data)))
```

```
import sklearn
from sklearn import linear_model
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

```
logistic, logistic_rmse = train_model(x, y, LogisticRegression())
```

5. **Neural Network:**

Lastly, we discussed how we can build a simple fully connected Neural network on the dataset. We looked into train test split, training on the model and finally plotting the training and the validation loss.