Modelling

library(tidyverse)

library(tidymodels)

library(rsample)

library(parsnip)

library(xgboost)

# Prepare Data

check for NAs

is.na(data)

split

basketball\_split = initial\_split(basketball, prop = 0.8, strata = 'shot\_made\_flag')

basketball\_train = training(basketball\_split)

basketball\_test = testing(basketball\_split)

# Recipe

In- & output

model\_recipe = recipe(shot\_made\_flag ~., data = basketball\_train)

model\_recipe %>%

Kategorische String-Variablen

step\_string2factor() %>%

Kategorische, Numerische-Variablen

step\_num2factor() %>%

Nicht-kontinuierliche Variablen (num & chr)

step\_dummy() %>%

step\_scale(all\_numeric()) %>%

num Variablen skalieren & zentrieren

step\_center(all\_numeric()) -> model\_recipe\_steps

# Prepare & Bake Recipe



prepare

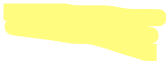
prepped\_recipe = prep(model\_recipe\_steps, training = basketball\_train)



basketball\_train\_preprocessed = bake(prepped\_recipe, basketball\_train)

bake

basketball\_test\_preprocessed = bake(prepped\_recipe, basketball\_test)



train Modells

# Train Models

classification: TRUE or FALSE?

🡪 errechnet: pred\_class

regression: z.B. Preis?

🡪 errechnet: .pred



## 4.1 Logistic Regression

logistic\_reg(mode = "classification") %>%



set\_engine("glm") %>%

fit(shot\_made\_flag ~ ., data = basketball\_train\_preprocessed) -> glm\_model



## Random Forest

rand\_forest(mode = "classification", trees = 250) %>%



set\_engine("ranger") %>%

fit(shot\_made\_flag ~ ., data = basketball\_train\_preprocessed) -> rf\_model



## 4.3 Boosting

boost\_tree(mode = "classification",



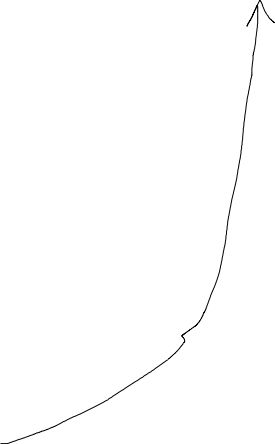
trees = 1500,

mtry = 3,

learn\_rate = 0.03,

sample\_size = 1,

tree\_depth = 5) %>%



set\_engine("xgboost") %>%

fit(shot\_made\_flag ~ ., data = basketball\_train\_preprocessed) -> boost\_model



# Evaluation

## 5.1 Prediction

model %>%



predict(new\_data = basketball\_test\_preprocessed) %>%

bind\_cols(basketball\_test\_preprocessed %>%

dplyr::select(shot\_made\_flag)) -> predictions\_model



## Confidence Matrix

predictions\_model %>%



conf\_mat(shot\_made\_flag, .pred\_class) %>%



summary() %>%

dplyr::select(-.estimator) %>%

filter(.metric %in% c("accuracy", "mcc", "f\_meas"))