Reproducible codes of the paper: Decomposition of Expected Goal Models: Aggregated SHAP Values for Analyzing Scoring Potential of Player/Team

Adrian Stańdo and Mustafa Cavus

November 25, 2022

Necessary Packages

```
library(tidyverse)      # data manipulation
library(ggplot2)        # data visualization
library(hrbrthemes)     # customization of plot theme
library(ROSE)           # over and under balancing data
library(forester)       # training tree-based models
# (from GitHub: https://github.com/ModelOriented/forester)
# version of forester: 1.0.1 (last available commit of this version:
# https://github.com/ModelOriented/forester/tree/2160324808c77049b4a162801b837d9c17884523 )
library(DALEX)          # using XAI tools
library(ingredients)    # creating CP and AP
library(worldfootballR) # scraping shot data
# (from GitHub: https://github.com/JaseZiv/worldfootballR)
```

Dataset

We focus in our paper on 361,035 shots-related event data (containing 362,207 goals of total shots) from the 14,481 matches in 8 seasons between 2014-15 and 2021-22 from the top-five European football leagues which are Serie A, Bundesliga, La Liga, English Premier League, Ligue 1. The dataset is collected from Understat by using the R-package `worldfootballR` and excluded the 1,172 shots resulting in own goals due to their unrelated pattern from the concept of the model. The following function is used for scraping the data from the leagues over 8 seasons:

(Do not forget that this steps takes a few hours depending on the processing power of your computer!)

```
# Ligue 1
ligue1_2021_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                           season_start_year = 2021)
ligue1_2020_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                           season_start_year = 2020)
ligue1_2019_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                           season_start_year = 2019)
ligue1_2018_shot_location <- understat_league_season_shots(league = "Ligue 1",
```

```
                                                        season_start_year = 2018)
ligue1_2017_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                        season_start_year = 2017)
ligue1_2016_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                        season_start_year = 2016)
ligue1_2015_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                        season_start_year = 2015)
ligue1_2014_shot_location <- understat_league_season_shots(league = "Ligue 1",
                                                        season_start_year = 2014)

# Serie A
seriea_2021_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2021)
seriea_2020_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2020)
seriea_2019_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2019)
seriea_2018_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2018)
seriea_2017_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2017)
seriea_2016_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2016)
seriea_2015_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2015)
seriea_2014_shot_location <- understat_league_season_shots(league = "Serie A",
                                                        season_start_year = 2014)

# Bundesliga
bundesliga_2021_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2021)
bundesliga_2020_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2020)
bundesliga_2019_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2019)
bundesliga_2018_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2018)
bundesliga_2017_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2017)
bundesliga_2016_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2016)
bundesliga_2015_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2015)
bundesliga_2014_shot_location <- understat_league_season_shots(league = "Bundesliga",
                                                           season_start_year = 2014)

# La Liga
laliga_2021_shot_location <- understat_league_season_shots(league = "La liga",
                                                        season_start_year = 2021)
laliga_2020_shot_location <- understat_league_season_shots(league = "La liga",
                                                        season_start_year = 2020)
laliga_2019_shot_location <- understat_league_season_shots(league = "La liga",
                                                        season_start_year = 2019)
```

```r
laliga_2018_shot_location <- understat_league_season_shots(league = "La liga",
                                                           season_start_year = 2018)
laliga_2017_shot_location <- understat_league_season_shots(league = "La liga",
                                                           season_start_year = 2017)
laliga_2016_shot_location <- understat_league_season_shots(league = "La liga",
                                                           season_start_year = 2016)
laliga_2015_shot_location <- understat_league_season_shots(league = "La liga",
                                                           season_start_year = 2015)
laliga_2014_shot_location <- understat_league_season_shots(league = "La liga",
                                                           season_start_year = 2014)

# La Liga
epl_2021_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2021)
epl_2020_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2020)
epl_2019_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2019)
epl_2018_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2018)
epl_2017_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2017)
epl_2016_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2016)
epl_2015_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2015)
epl_2014_shot_location <- understat_league_season_shots(league = "EPL",
                                                        season_start_year = 2014)

# combining data
raw_data <- rbind(ligue1_2021_shot_location,
                  ligue1_2020_shot_location,
                  ligue1_2019_shot_location,
                  ligue1_2018_shot_location,
                  ligue1_2017_shot_location,
                  ligue1_2016_shot_location,
                  ligue1_2015_shot_location,
                  ligue1_2014_shot_location,

                  seriea_2021_shot_location,
                  seriea_2020_shot_location,
                  seriea_2019_shot_location,
                  seriea_2018_shot_location,
                  seriea_2017_shot_location,
                  seriea_2016_shot_location,
                  seriea_2015_shot_location,
                  seriea_2014_shot_location,

                  bundesliga_2021_shot_location,
                  bundesliga_2020_shot_location,
                  bundesliga_2019_shot_location,
                  bundesliga_2018_shot_location,
                  bundesliga_2017_shot_location,
```

```
                  bundesliga_2016_shot_location,
                  bundesliga_2015_shot_location,
                  bundesliga_2014_shot_location,

                  laliga_2021_shot_location,
                  laliga_2020_shot_location,
                  laliga_2019_shot_location,
                  laliga_2018_shot_location,
                  laliga_2017_shot_location,
                  laliga_2016_shot_location,
                  laliga_2015_shot_location,
                  laliga_2014_shot_location,

                  epl_2021_shot_location,
                  epl_2020_shot_location,
                  epl_2019_shot_location,
                  epl_2018_shot_location,
                  epl_2017_shot_location,
                  epl_2016_shot_location,
                  epl_2015_shot_location,
                  epl_2014_shot_location)

# saving data
write.csv(raw_data, "./data/raw_data.csv")
```

Pre-processing of the raw dataset

This section introduces the dataset and how it is pre-processed. First data is imported from a .csv file is `raw_data`, then the features `distanceToGoal` and `angleToGoal` are extracted from the coordinated `X` and `Y`. The features `status`, `distanceToGoal`, `angleToGoal`, `h_a`, `shotType`, `lastAction`, `minute`, `league`, and `season` are prepared for analysis and modeling.

```
# importing the previously scraped data from local to save time
raw_data <- read_csv("./data/raw_data.csv")

raw_data_without_owngoals <- raw_data %>% filter(result != "OwnGoal")
write.csv(raw_data_without_owngoals, "./data/raw_data_without_owngoals.csv") # saving data

shot_stats <- raw_data_without_owngoals %>%
  mutate(status = ifelse(result == "Goal", 1, 0)) %>%
  mutate(distanceToGoal = sqrt((105 - (X * 105)) ^ 2 + (32.5 - (Y * 68)) ^ 2)) %>%
  mutate(angleToGoal = abs(atan((7.32 * (105 - (X * 105))) / ((105 - (X * 105))^2 +
         (32.5 - (Y * 68)) ^ 2 - (7.32 / 2) ^ 2)) * 180 / pi)) %>%
  mutate(h_a = factor(h_a),
         situation = factor(situation),
         shotType = factor(shotType),
         lastAction = factor(lastAction),
         minute = as.numeric(minute)) %>%
  select(status, minute, h_a, situation, shotType, lastAction,
         distanceToGoal, angleToGoal, league, season, match_id, result, player_id)
```

Preparing sets for model training

```
# preparing train set of original dataset
train_data <- shot_stats %>%
   select(status, minute, h_a, situation, shotType, lastAction,
          distanceToGoal, angleToGoal)

# saving the preprocessed dataset
write.csv(train_data, './data/data_preprocessed.csv')

# preparing train set of under-sampled dataset
set.seed(123)
under_train_data <- ovun.sample(status ~ ., data = train_data, method = "under")

# preparing train set of over-sampled dataset
set.seed(123)
over_train_data <- ovun.sample(status ~ ., data = train_data, method = "over")
```

Model training

Modifications on {forester} version 1.0.1

We changed and expanded some functions of the forester package. You can see the reasons for this action below:

- The forester returns the predicted labels, we changed this with predicted probabilities to calculate the performance metrics which are based on probabilities such as log-loss, Brier score and MCC.
- The forester returns only the output of the best performing model in terms of the value of intended metric, we expanded it to return the output of all models for comparing their performance with the additional metrics.
- After under-sample the dataset, the ranger changes the reference class in the model and causes a inconsistency. Thus, we add an argument to the make_ranger and forester functions to control the reference class.

```
setwd("./changes_forester") # calling the modified function from local
source("evaluate.R")
source("forester.R")
source("make_ranger.R")
source("make_xgboost.R")
source("make_lightgbm.R")
source("make_catboost.R")
source("model_performancex.R")
setwd("..")
```

We use the forester `forester` AutoML tool to train various tree-based classification models from `XGBoost`, `randomForest`, `LightGBM`, and `CatBoost` libraries.

```
# training tree-based models on original dataset
set.seed(123)
original_model <- forester(data   = train_data,
                           target = "status",
                           type   = "classification")

# training tree-based models on under-sampled dataset
set.seed(123)
under_model <- forester(data    = under_train_data$data,
                        target  = "status",
                        type    = "classification",
                        refclass = "")

# training tree-based models on over-sampled dataset
set.seed(123)
over_model <- forester(data   = over_train_data$data,
                       target = "status",
                       type   = "classification")
```

Performance of trained xG models

```
# performance of random forest model
# on over-sampled data
model_performancex(over_model$model3)
```

```
# on under-sampled data
model_performancex(under_model$model3)
```

```
# on original data
model_performancex(original_model$model3)
```

```
# performance of catboost model
# on over-sampled data
model_performancex(over_model$model1)
```

```
# on under-sampled data
model_performancex(under_model$model1)
```

```
# on original data
model_performancex(original_model$model1)
```

```
# performance of xgboost model
# on over-sampled data
model_performancex(over_model$model2)
```

```
# on under-sampled data
model_performancex(under_model$model2)
```

```
# on original data
model_performancex(original_model$model2)
```

```
# performance of lightgbm model
# on over-sampled data
model_performancex(over_model$model4)
```

```
# on under-sampled data
model_performancex(under_model$model4)
```

```
# on original data
model_performancex(original_model$model4)
```

The random forest model trained on oversampled dataset turned out to be the best, so that is why it is used in further analysis.

```
# saving the best model
model <- over_model$model3
saveRDS(model, file = "./model/model.RDS")
```

aSHAP

The aggregated SHAP plots and calculations were created by extending an existing {DALEX} library.

```
setwd("./changes_dalex")
source("aSHAP.R")
setwd("..")

# importing the previously trained model from local to save time
model <- readRDS("./model/model.RDS")
# importing the previously preprocessed data from local to save time
train_data <- read.csv('./data/data_preprocessed.csv')
data <- train_data
```

```
explainer <- DALEX::explain(model = model,
                            data = data[names(data) != 'status'],
                            y = data$status
                            )
```

Figures

The variable order will be the same for all the plots.

```
order_variables <- c('minute', 'h_a', 'situation', 'shotType', 'lastAction',
                     'distanceToGoal', 'angleToGoal')
```

Robert Lewandowski

Season 2019/2020

```
set.seed(42)
df_rl_19_20 <- data[data$player == 'Robert Lewandowski' && data$season == 2019,]
rl_19_20 <- predict_parts_shap_aggregated(explainer, df_rl_19_20,
                                          B=15, order_variables = order_variables)
plot(rl_19_20, add_boxplots = FALSE,
     subtitle = "lewandowski-season2019", max_features = 10)
```

Season 2020/2021

```
set.seed(42)
df_rl_20_21 <- data[data$player == 'Robert Lewandowski' && data$season == 2020,]
rl_20_21 <- predict_parts_shap_aggregated(explainer, df_rl_20_21,
                                          B=15, order_variables = order_variables)
plot(rl_20_21, add_boxplots = FALSE,
     subtitle = "lewandowski-season2020", max_features = 10)
```

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ Robert Lewandowski is 1st top scorer.

```
set.seed(42)
df_rl_21_22 <- data[data$player == 'Robert Lewandowski' && data$season == 2021,]
rl_21_22 <- predict_parts_shap_aggregated(explainer, df_rl_21_22,
                                          B=15, order_variables = order_variables)
plot(rl_21_22, add_boxplots = FALSE,
     subtitle = "lewandowski-season2021", max_features = 10)
```

Cristiano Ronaldo

Season 2021/2022

```
set.seed(42)
df_cr_21_22 <- data[data$player == 'Cristiano Ronaldo' && data$season == 2021,]
cr_21_22 <- predict_parts_shap_aggregated(explainer, df_cr_21_22,
                                          B=15, order_variables = order_variables)
plot(cr_21_22, add_boxplots = FALSE,
     subtitle = "ronaldo-season2021", max_features = 10)
```

Patrik Schick

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ Patrik Schick is 2nd top scorer.

```
set.seed(42)
df_ps_21_22 <- data[data$player == 'Patrik Schick' && data$season == 2021,]
ps_21_22 <- predict_parts_shap_aggregated(explainer, df_ps_21_22,
                                          B=15, order_variables = order_variables)
plot(ps_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-schick-season2021", max_features = 10)
```

Max Kruse

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ Max Kruse is 10th top scorer.

```
set.seed(42)
df_mk_21_22 <- data[data$player == 'Max Kruse' && data$season == 2021,]
mk_21_22 <- predict_parts_shap_aggregated(explainer, df_mk_21_22,
                                          B=15, order_variables = order_variables)
plot(mk_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-schick-season2021", max_features = 10)
```

Bayern Munich

```
raw_data_without_owngoals <- read_csv("./data/raw_data_without_owngoals.csv")
```

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ Bayern Munich is the first best team in season 2021.

```
set.seed(42)

df_bm_21_22 <- data[
  (raw_data_without_owngoals$home_team == 'Bayern Munich' &&
     data$h_a == 'h' && data$season == 2021) ||
  (raw_data_without_owngoals$away_team == 'Bayern Munich' &&
     data$h_a == 'a' && data$season == 2021),]

bm_21_22 <- predict_parts_shap_aggregated(explainer, df_bm_21_22,
                                          B=15, order_variables = order_variables)
plot(bm_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-bayern_minuch-season2021", max_features = 10)
```

Borussia Dortmund

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ Borussia Dortmund is the second best team in season 2021.

```
set.seed(42)

df_bd_21_22 <- data[
  (raw_data_without_owngoals$home_team == 'Borussia Dortmund' &&
     data$h_a == 'h' && data$season == 2021) ||
  (raw_data_without_owngoals$away_team == 'Borussia Dortmund' &&
     data$h_a == 'a' && data$season == 2021),]

bd_21_22 <- predict_parts_shap_aggregated(explainer, df_bd_21_22,
                                          B=15, order_variables = order_variables)
plot(bd_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-borussia_dortmund-season2021", max_features = 10)
```

VfB Stuttgart

Season 2021/2022

According to https://www.flashscore.com/football/germany/bundesliga-2021-2022/ VfB Stuttgart is 15th team in season 2021.

```
set.seed(42)

df_vs_21_22 <- data[
  (raw_data_without_owngoals$home_team == 'VfB Stuttgart' &&
     data$h_a == 'h' && data$season == 2021) ||
  (raw_data_without_owngoals$away_team == 'VfB Stuttgart' &&
     data$h_a == 'a' && data$season == 2021),]

vs_21_22 <- predict_parts_shap_aggregated(explainer, df_vs_21_22,
                                          B=15, order_variables = order_variables)
plot(vs_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-vfb_stuttgart-season2021", max_features = 10)
```

The whole Bundesliga

Season 2021/2022

```
set.seed(42)

df_b_21_22 <- data[raw_data_without_owngoals$league == 'Bundesliga' &&
                     data$season == 2021,]

b_21_22 <- predict_parts_shap_aggregated(explainer, df_b_21_22,
```

```
                                        B=15, order_variables = order_variables)
plot(b_21_22, add_boxplots = FALSE,
     subtitle = "bundesliga-all_teams-season2021", max_features = 10)
```

SHAP Dependence Scatter Plots

distanceToGoal

```
column <- 'distanceToGoal'
column_vals <- train_data[,column]
B <- 15 # parameter from previous calculations

shaps_raw <- b_21_22$raw
shaps_raw <- shaps_raw[shaps_raw$variable_name == column, 'variable_name']

shaps <- c()

for i in 1:(as.integer(length(shaps_raw / B))){
  shap_val <- mean(shaps_raw[(B*(i-1) + 1):(B*i)])
  shaps <- c(shaps, shap_val)
}

vals <- data.frame(
          c1 = shaps,
          c2 = column_vals
        )
colnames(vals) <- c('SHAP', column)

ggplot(vals, aes_string(x=column, y='SHAP')) +
  geom_point(col='#4c72b0') +
  theme_bw() +
  ggtitle(paste0("SHAP Dependence Scatter Plot for ", column, " variable"))
```

angleToGoal

```
column <- 'angleToGoal'
column_vals <- train_data[,column]
B <- 15 # parameter from previous calculations

shaps_raw <- b_21_22$raw
shaps_raw <- shaps_raw[shaps_raw$variable_name == column, 'variable_name']

shaps <- c()

for i in 1:(as.integer(length(shaps_raw / B))){
  shap_val <- mean(shaps_raw[(B*(i-1) + 1):(B*i)])
  shaps <- c(shaps, shap_val)
}
```

```
vals <- data.frame(
          c1 = shaps,
          c2 = column_vals
        )
colnames(vals) <- c('SHAP', column)

ggplot(vals, aes_string(x=column, y='SHAP')) +
  geom_point(col='#4c72b0') +
  theme_bw() +
  ggtitle(paste0("SHAP Dependence Scatter Plot for ", column, " variable"))
```

Session info

```
sessionInfo()
```