

Analýza dat pro Bakalářskou práci

Michal Lauer

2022-05-01

Funkce a globální nastavení

Nastavení generování bloků kódu a výstupu z print funkcí, načtení knihoven, a načtení funkcí.

Nastavení

```
knitr::opts_chunk$set(message = F)

options("OutDec" = ",")
options("xtable.include.rownames" = F)
options("xtable.table.placement" = "H")
options("xtable.caption.placement" = "top")
options("xtable.format.args" = list(decimal.mark = ","))
options("digits" = 4)
```

Knihovny

```
library(readr)
library(dplyr)
library(stringr)
library(tidyr)
library(ggplot2)
library(GGally)
library(caret)
library(xtable)
library(rsample)
library(paletter)

theme_set(theme_minimal() +
  theme(
    plot.caption = element_text(face = "italic")
  ))
```

Globální proměnné

```

# Seznam map, které jsou potřebné pro analýzy dat
maps <- c(
  "Dust2", "Inferno", "Vertigo", "Overpass", "Nuke", "Mirage",
  "Train", "Cache", "Cobblestone"
)

# Adresář, kam ukládat grafy
PICTURES_DIR <- "../bakalarska-prace/obrazky/"
if (!dir.exists(PICTURES_DIR)) dir.create(PICTURES_DIR)

# Adresář, kam ukládat vytvořené latex tabulky
TABLES_DIR <- "../bakalarska-prace/tabulky/"
if (!dir.exists(TABLES_DIR)) dir.create(TABLES_DIR)

# Adresář, kam ukládat tabulky vytvořených modelů
MODELS_DIR <- "../bakalarska-prace/modely/"
if (!dir.exists(MODELS_DIR)) dir.create(MODELS_DIR)

# Adresář, kam ukládat latex tabulky pro matice záměn
MATRIXES_DIR <- "../bakalarska-prace/matice/"
if (!dir.exists(MATRIXES_DIR)) dir.create(MATRIXES_DIR)

# Adresář, kam ukládat latex tabulky pro výsledné statistiky z
# matice záměn
STATS_DIR <- "../bakalarska-prace/statistiky/"
if (!dir.exists(STATS_DIR)) dir.create(STATS_DIR)

```

Uložení grafu

```

#' Uložení grafu jako png
#'
#' @param .graph Graf z knihovny ggplot
#' @param .file Soubor, do kterého obrázek uložit
save_bc_picture <- function(.graph, .file) {

  # Vytvoření cesty pro soubor
  .file = paste(PICTURES_DIR, .file, sep = "/")

  # Uložení obrázku
  ggsave(
    .file,
    plot = .graph,
    bg = "white",
    width = 150,
    height = 105,
    units = 'mm'
  )
}

```

Uložení tabulky

```
#' Uložení libovolné tabulky do LaTeX souboru
#'
```

```
#' @param .tibble Tabulka
#' @param .file Soubor, do kterého uložit tabulku
#' @param .cap Caption
#' @param .lab Label
#' @param .scale Poměr tabulky
save_bc_table <- function(.tibble, .file, .cap, .lab,
                          .scale = 1) {

  # Pokud není zadáný label, použije se jako label název souboru
  # bez přípony
  if (is.na(.lab)) {
    .lab <- sprintf("tab:%s", str_extract(.file, "^.*(?=\\.tex)"))
  }

  # Protože xtable neumí vypsát datum, převede se na text
  if ("date" %in% colnames(.tibble)) {
    .tibble <- .tibble |>
      mutate(
        across(
          .cols = "date",
          .fns = as.character))
  }

  # Vytvoření pojmenované tabulky
  xtable <- xtable(
    .tibble,
    caption = .cap,
    label = .lab,
    digits = 4
  )

  # V xtable existuje bug, kde při globálním nastavení decimálního
  # oddělovače na ',' se vypíše .scale s čárkou a LaTeX hází error
  options(OutDec = ".")

  # Vytvoření cesty pro soubor
  .file = paste(TABLES_DIR, .file, sep = "/")

  # Uložení tabulky do souboru
  print(xtable,
        file = .file,
        scalebox = .scale
  )

  # Znovunastavení decimálního oddělovače na ','
  options(OutDec = ",")
}
```

Uložení modelu do tabulky

```
#' Uložení logistického modelu do LaTeX souboru ve formě tabulky
#'  
#' @param .model Logistický model  
#' @param .file Soubor, do kterého tabulku uložit  
#' @param .cap Caption  
#' @param .lab Label  
save_bc_model <- function(.model, .file, .cap, .lab) {  
  
  # Pokud není zadáný label, použije se jako label název souboru  
  # bez přípony  
  if (is.na(.lab)) {  
    .lab <- sprintf("tab:%s", str_extract(.file, "^.*(?=\\.tex)"))  
  }  
  
  # Vytvoření pojmenované tabulky  
  xtable <- xtable(  
    .model,  
    caption = .cap,  
    label = .lab,  
    digits = 4  
  )  
  
  # Pokud není zadáný label, vytvoří se tabulka bez labelu a  
  # captionu  
  if (is.null(.lab)) {  
    xtable <- xtable(.model,  
                      digits = 4 )  
  }  
  
  # Vytvoření cesty pro soubor  
  .file = paste(MODELS_DIR, .file, sep = "/")  
  
  # Uložení tabulky do souboru  
  print(xtable,  
        file = .file,  
        include.rownames = T  
  )  
}
```

Uložení matice záměn do tabulky

```
#' Uložení matice záměn do LaTeX souboru  
#'  
#' @param .table Tabulka z matice záměn  
#' @param .file Soubor, do tabulku z matice záměn uložit  
#' @param .cap Caption  
#' @param .lab Label  
save_bc_matrix <- function(.table, .file, .cap, .lab) {
```

```

# Přeložen názvů sloupečků
rownames = c("Pozitivní predikce", "Negativní predikce")
colnames = c("Původní pozitivní", "Původní negativní")

dimnames(.table) <- list(
  rownames,
  colnames
)

# Pokud není zadáný label, použije se jako label název souboru
# bez přípony
if (is.na(.lab)) {
  .lab <- sprintf("tab:%s", str_extract(.file, "^.*(?=\\.tex)"))
}

# Vytvoření pojmenované tabulky
xtable <- xtable(
  .table,
  caption = .cap,
  label = .lab,
  digits = 4
)

# Pokud není zadáný label, vytvoří se tabulka bez labelu a
# captionu
if (is.null(.lab)) {
  xtable <- xtable(.table,
    digits = 4)
}

# Vytvoření cesty pro soubor
.file = paste(MATRIXES_DIR, .file, sep = "/")

# Uložení tabulky do souboru
print(xtable,
  file = .file,
  include.rownames = T
)
}

```

Uložení statistik z matice záměn do tabulky

```

#' Uložení statistik z matice záměn do LaTeX souboru
#'
#' @param .matrix Matice záměn
#' @param .file Soubor, do kterého statistiky uložit
#' @param .cap Caption matice
#' @param .lab Label matice
save_bc_matrix_output <- function(.matrix, .file, .cap, .lab) {

  # Spojení statistik

```

```

.matrix <- bind_rows(
  tibble(
    statistika = names(.matrix$overall),
    hodnota = .matrix$overall
  ),
  tibble(
    statistika = names(.matrix$byClass),
    hodnota = .matrix$byClass
  )
) |>
# Získání relevantních statistik a přeložení statistik
filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
mutate(statistika = c("Přesnost", "Senzitivita", "Specifická"))

# Pokud není zadáný label, použije se jako label název souboru
# bez přípony
if (is.na(.lab)) {
  .lab <- sprintf("tab:%s", str_extract(.file, "^.*(?!\\.tex)"))
}

# Vytvoření pojmenované tabulky
xtable <- xtable(
  .matrix,
  caption = .cap,
  label = .lab,
  digits = 4
)

# Pokud není zadáný label, vytvoří se tabulka bez labelu a
# captionu
if (is.null(.lab)) {
  xtable <- xtable(.matrix,
    digits = 4 )
}

# Vytvoření cesty pro soubor
.file = paste(STATS_DIR, .file, sep = "/")

# Uložení tabulky do souboru
print(xtable,
  file = .file
)
}

```

Vytvoření matice záměn

```

#' Získání matice záměn
#'
#' @param .model Model pro matici záměn
#' @param .split Split z knihovny rsample

```

```

#'
#' @return Matice záměn z knihovny caret
get_matrix <- function(.model, .split) {
  confusionMatrix(
    # Predikce na testovací množině dat
    data = predict(object = .model,
                   newdata = testing(.split),
                   type = "response") |>
      round() |>
      as.factor(),
    # Původní hodnoty z testovací množiny dat
    reference = testing(.split)$map_winner |>
      as.factor(),
    positive = "1"
  )
}

```

Získání charakteristik pro hráče na určité mapě

```

#' Funkce pro extrakci charakteristik hráče na mapě v jedno zápase
#'
#' @param tbl datový soubor players.csv
#' @param map číslo mapy pro extrakci charakteristik
#'
#' @return tabulka s charakteristikami hráčů na dané mapě
clear_player_map <- function(.tbl, .map) {
  m_ <- sprintf("m%i_", .map)
  .tbl |>
    select(match_id, player_id, team,
           # Vybrání mapy a statistik podle pro i-tou mapu
           map = sprintf("map_%i", .map),
           matches(sprintf("~m%i_", .map))
    ) |>
    # Získání pouze agregovaných statistik za obě strany
    # e.g. vyloučení m1_kills_ct, m1_kills_t a ponechání pouze m1_kills
    select(!matches("_c?t$")) |>
    drop_na(map) |>
    # Sjedení názvů, e.g. m1_kills -> kills
    rename_with(
      .cols = starts_with(m_),
      .fn = ~ str_remove_all(.x, m_)
    ) |>
    # Odstranění sloupečků, které nejsou kompletní
    # kvůli historii vývoje cs:go a hltv.org
    select(-c(kddiff, flash_assists, kast, adr)) |>
    # konverze HS na relativní jednotky
    mutate(hs = ifelse(kills == 0, 0, hs / kills))
}

```

Sjednocení výsledků pro daný tým

```
#' Získání charakteristik týmu v daném zápase
#'
#' @param .tbl Datový soubor results.csv
#' @param i Číslo týmu pro získání charakteristik
#'
#' @return Tabulka s charakteristikami týmu v zápase
clear_results_team <- function(.tbl, .i) {
  .tbl |>
    # Vybrání relevantních charakteristik pro daný tým
    select(
      date, match_id, team = matches(sprintf("team_%i", .i)),
      map, map_winner, starting_ct,
      team_rank = matches(sprintf("rank_%i", .i))
    ) |>
    # Získání pouze relevantních map
    filter(map %in% maps) |>
    mutate(
      map_winner = as.numeric(map_winner == .i),
      starting_ct = as.numeric(starting_ct == .i)
    )
}
```

Geometrický průměr

```
#' Získání geometrického průměru pomocí log transformace
#'
#' @param x Vektor pro vypočetění geometrického průměru
#'
#' @return Geometrický průměr
geom_mean <- function(x) {
  exp(sum(log(x)) / length(x))
}
```

Datový soubor players.csv

Nahrání dat

```
players.raw <- read_csv("data/players.csv", show_col_types = F)
```

Čištění dat

```
players <-
  # Získání charakteristik pro první mapu
  clear_player_map(players.raw, 1) |>
```



```

# Získání charakteristik pro druhou mapu
rbind(clear_player_map(players.raw, 2)) |>
# Získání charakteristik pro třetí mapu
rbind(clear_player_map(players.raw, 3)) |>
filter(map %in% maps) |>
drop_na() |>
# Odstranění map, kde není právě pět hráčů
group_by(match_id, team, map) |>
filter(n() == 5) |>
ungroup() |>
mutate(
  match_id = as.integer(match_id),
  player_id = as.integer(player_id),
  kills = as.integer(kills),
  assists = as.integer(assists),
  deaths = as.integer(deaths),
  fkdifff = as.integer(fkdifff)
)

```

Uložení LaTeX tabulky

```

set.seed(12345)
players |>
  slice_sample(n = 6) |>
  save_bc_table(
    .tibble = _,
    .file = "players_csv_transformovano.tex",
    .cap = "Záznam z transformovaného datového souboru players.csv",
    .lab = NA,
    .scale = 0.90
  )

```

Datový soubor results.csv

Nahrání dat

```

results.raw <- read_csv("data/results.csv", show_col_types = F)

```

Čištění dat

```

results <- results.raw |>
  select(date, team_1, team_2,
         map = `_map`, map_winner,
         starting_ct, rank_1, rank_2, match_id) |>
  # Smazání záznamu ke tým hrál sám proti sobě
  # Chybná extrakce dat

```

```

filter(match_id != 2314674)

results <-
  # Získání charakteristik pro první tým
  clear_results_team(.tbl = results, 1) |>
  # Získání charakteristik pro druhý tým
  rbind(clear_results_team(.tbl = results, 2)) |>
  mutate(
    match_id = as.integer(match_id),
    map_winner = as.integer(map_winner),
    starting_ct = as.integer(starting_ct),
    team_rank = as.integer(team_rank)
  ) |>
  mutate(team = ifelse(team == "?", "Astralis", team))

```

Uložení LaTeX tabulky

```

set.seed(9574)
results |>
  slice_sample(n = 6) |>
  save_bc_table(
    .tibble = _,
    .file = "results_csv_transformovano.tex",
    .cap = "Příklad záznamu z transformovaného datového souboru results.csv",
    .lab = NA,
    .scale = 1
  )

```

Spojený datový soubor

Spojení dat

```

data <- results |>
  inner_join(players,
    by = c("match_id", "team", "map"))

```

Uložení LaTeX tabulky

```

set.seed(9574)
data |>
  slice_sample(n = 6) |>
  save_bc_table(
    .tibble = _,
    .file = "data_spojena.tex",
    .cap = "Příklad záznamu z transformovaného datového souboru results.csv",
    .lab = NA,
  )

```

```
.scale = 1  
)
```

Agregovaný datový soubor

Agregace

```
data.aggregated <- data |>  
  group_by(match_id, map, team) |>  
  summarise(  
    mean_kills = mean(kills),  
    mean_assists = mean(assists),  
    mean_deaths = mean(deaths),  
    mean_hs = geom_mean(hs),  
    mean_fkdiff = mean(fkdiff),  
    # Průměr z vektoru stejných čísel c je c  
    map_winner = mean(map_winner),  
    starting_ct = mean(starting_ct),  
    team_rank = mean(team_rank),  
    .groups = "drop"  
  )
```

Uložení LaTeX tabulky

```
set.seed(6587)  
data.aggregated |>  
  slice_sample(n = 6) |>  
  save_bc_table(  
    .tibble = _,  
    .file = "data_agregovana.tex",  
    .cap = "Agregovaná data pro týmy za zápas a mapu",  
    .lab = NA,  
    .scale = 0.5  
  )
```

Průzkumová analýza dat

Kvantitativní proměnné

```
g <- data |>  
  select(kills, assists, deaths, hs, fkdiff, rating, team_rank) |>  
  ggcorr(hjust = 0.8, size = 4, label = T) +  
  labs(  
    title = "Korelační matice kvantitativních prediktorů",  
    caption = "Spojený datový soubor"
```

```
)

save_bc_picture(.graph = g,
                .file = "prediktory_corr_matice.png")
```

Histogram prediktorů

```
bin_count <- round(1 + 3.3*log10(nrow(data)/5))
g <- data |>
  pivot_longer(
    cols = c(kills, assists, deaths, hs, fkdifff),
    names_to = "stats",
    values_to = "vals"
  ) |>
  select(stats, vals) |>
  mutate(bins = 1 + 3.3*log10(n()/6),
         bins = 1) |>
  ggplot(aes(x = vals, fill = stats)) +
  facet_wrap(vars(stats), ncol = 2, scales = "free") +
  geom_histogram(bins = bin_count, show.legend = F) +
  scale_y_continuous(labels = scales::number) +
  theme(
    axis.title.x = element_blank(),
    strip.background = element_rect(fill = "grey")
  ) +
  labs(
    title = "Rozdělení kvantitativních statistik",
    subtitle = "Osa Y je přizpůsobená každé statistice",
    y = "Absolutní počet",
    caption = "Spojení souboru players.csv a results.csv"
  )

save_bc_picture(.graph = g,
                .file = "histogram_prediktoru.png")
```

Míra výhry proměnných

```
data.plot <- data.aggregated |>
  semi_join(
    data.aggregated |>
      group_by(team) |>
      filter(n() > 100) |>
      summarise(mtr = mean(team_rank)) |>
      slice_min(order_by = mtr, n = 10),
    by = "team"
  ) |>
  group_by(team) |>
  summarise(mmw = mean(map_winner)) |>
  arrange(desc(mmw))
```

```

g <- ggplot(data = data.plot,
  aes(x = reorder(team, mmw), y = mmw, fill = team)) +
  geom_col(show.legend = F) +
  geom_text(data = filter(data.plot, mmw > 0.6),
    aes(label = paste0(round(mmw * 100, 2), " %"),
      y = mmw + 0.08)) +
  scale_fill_manual(values = paletteer_d("basetheme::deepblue")) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 0.8)) +
  coord_flip() +
  labs(
    title = "Porovnání míry výhry pro nejlepších 10 týmů",
    subtitle = "Pouze pro týmy s alespoň 100 zápasy\n0sa x je omezená na interval <0; 0,8>",
    x = "Tým",
    y = "Míra výhry",
    caption = "Agregovaný datový soubor"
  )

save_bc_picture(.graph = g,
  .file = "mira_vyhry_tymu.png")

```

Modely pro hráče - Mirage

Optimalizovaný model

```

set.seed(547)
mirage.data <- data |>
  filter(map == "Mirage")

mirage.split <- initial_split(mirage.data, prop = 0.8)

mirage.model <- glm(
  formula = map_winner ~ kills + assists + deaths + hs +
    fkdif + starting_ct,
  data = training(mirage.split),
  family = binomial
)

```

Uložení LaTeX tabulky

```

save_bc_model(
  .model = mirage.model,
  .file = "Mirage_model.tex",
  .cap = "Výstup z programu R pro model hráčů na mapě Mirage",
  .lab = NA
)

```

Matice záměn

```
mirage.matrix <- get_matrix(  
  .model = mirage.model,  
  .split = mirage.split  
)
```

Uložení LaTeX tabulky

```
save_bc_matrix(  
  .table = mirage.matrix$table,  
  .file = "Mirage_matice.tex",  
  .cap = " Matice záměn pro model hráčů na mapě Mirage",  
  .lab = NA  
)  
  
save_bc_matrix_output(  
  .matrix = mirage.matrix,  
  .file = "Mirage_stats.tex",  
  .cap = "Vybrané statistiky z matice záměn pro model hráčů na mapě Mirage",  
  .lab = NA  
)
```

Modely pro hráče - Vertigo

Neoptimalizovaný model

```
set.seed(547)  
vertigo.data <- data |>  
  filter(map == "Vertigo")  
  
vertigo.split <- initial_split(vertigo.data, prop = 0.8)  
  
vertigo.model <- glm(  
  formula = map_winner ~ kills + assists + deaths + hs +  
    fkdif + starting_ct,  
  data = training(vertigo.split),  
  family = binomial  
)
```

Uložení LaTeX tabulky

```
save_bc_model(  
  .model = vertigo.model,  
  .file = "Vertigo_model.tex",
```

```

    .cap = "Výstup z programu R pro model hráčů na mapě Vertigo",
    .lab = NA
)

```

Optimalizovaný model

```

set.seed(547)
vertigo.data <- data |>
  filter(map == "Vertigo")

vertigo.split <- initial_split(vertigo.data, prop = 0.8)

vertigo.model <- glm(
  formula = map_winner ~ kills + assists + deaths,
  data = training(vertigo.split),
  family = binomial
)

```

Uložení LaTeX tabulky

```

save_bc_model(
  .model = vertigo.model,
  .file = "Vertigo_model_opt.tex",
  .cap = "Výstup z programu R pro optimalizovaný model hráčů na mapě Vertigo",
  .lab = NA
)

```

Matice záměn

```

vertigo.matrix <- get_matrix(
  .model = vertigo.model,
  .split = vertigo.split
)

```

Uložení LaTeX tabulky

```

save_bc_matrix(
  .table = vertigo.matrix$table,
  .file = "Vertigo_matice_opt.tex",
  .cap = "Matice záměn pro model hráčů na mapě Vertigo",
  .lab = NA
)

save_bc_matrix_output(
  .matrix = vertigo.matrix,

```

```

.file = "Vertigo_stats_opt.tex",
.cap = "Vybrané statistiky z matice záměn pro modelu hráčů na mapě Vertigo",
.lab = NA
)

```

Porovnání statistik obou modelů

```

# Vytvoření statistik z matice záměn pro mapu Mirage
m1 <-
  bind_rows(
    tibble(
      statistika = names(mirage.matrix$overall),
      `Mapa Mirage` = mirage.matrix$overall
    ),
    tibble(
      statistika = names(mirage.matrix$byClass),
      `Mapa Mirage` = mirage.matrix$byClass
    )
  ) |>
  filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
  mutate(statistika = c("Přesnost", "Senzitivita", "Specifická"))

# Vytvoření statistik z matice záměn pro mapu Vertigo
m2 <-
  bind_rows(
    tibble(
      statistika = names(vertigo.matrix$overall),
      `Mapa Vertigo` = vertigo.matrix$overall
    ),
    tibble(
      statistika = names(vertigo.matrix$byClass),
      `Mapa Vertigo` = vertigo.matrix$byClass
    )
  ) |>
  filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
  mutate(statistika = c("Přesnost", "Senzitivita", "Specifická"))

m12 <-
  cbind(
    m1,
    m2 |> select(-statistika)
  )

save_bc_table(
  .tibble = m12,
  .file = "Mirage_Vertigo_porovnani.text",
  .cap = "Porovnání statistik pro model hráčů na mapě Mirage a Vertigo",
  .lab = NA
)

```


Uložení ostatních map

```
set.seed(962147)
# Uložení modelů, matic záměn a statistik pro zbývající mapy
for (.map in setdiff(maps, c("Mirage", "Vertigo"))) {

  .data <- data |>
    filter(map == .map)

  .split <- initial_split(.data, prop = 0.8)

  .model <- glm(
    formula = map_winner ~ kills + assists + deaths + hs +
      fkdif + starting_ct,
    data = training(.split),
    family = binomial
  )

  save_bc_model(
    .model = .model,
    .file = sprintf("%s_model.tex",
      .map),
    .cap = sprintf(
      "Výstup z programu R pro model hráčů na mapě %s",
      .map),
    .lab = NA
  )

  .matrix <- get_matrix(
    .model = .model,
    .split = .split
  )

  save_bc_matrix(
    .table = .matrix$table,
    .file = sprintf(
      "%s_matice.tex",
      .map),
    .cap = sprintf(
      "Matice záměn pro model hráčů na mapě %s",
      .map),
    .lab = NA
  )

  save_bc_matrix_output(
    .matrix = .matrix,
    .file = sprintf("%s_stats.tex",
      .map),
    .cap = sprintf("Vybrané statistiky z matice záměn pro model hráčů na mapě %s",
      .map),
    .lab = NA
  )
}
```

Model pro všechny týmy

Optimalizovaný model

```
set.seed(4872)
global.split <- initial_split(data.aggregated, prop = 0.8,
                              strata = map)

global.model <-
  glm(
    formula = map_winner ~ mean_kills + mean_assists + mean_deaths +
      mean_hs + mean_fkdiff + team_rank + starting_ct:map,
    data = training(global.split),
    family = binomial
  )
```

Uložení LaTeX tabulky

```
save_bc_model(
  .model = global.model,
  .file = "Global_model.tex",
  .cap = "Výstup z programu R pro model týmu pro všechny týmy",
  .lab = NA
)
```

Matice záměn

```
global.matrix <- get_matrix(
  .model = global.model,
  .split = global.split
)
```

Uložení LaTeX tabulky

```
save_bc_matrix(
  .table = global.matrix$table,
  .file = "Global_matice.tex",
  .cap = "Matice záměn pro model týmu pro všechny týmy",
  .lab = NA
)

save_bc_matrix_output(
  .matrix = global.matrix,
  .file = "Global_stats.tex",
  .cap = "Vybrané statistiky z matice záměn pro model týmu pro všechny týmy",
  .lab = NA
)
```

Model pro tým Astralis

Neotimalizovaný model

```
set.seed(417)
astralis.split <- initial_split(data.aggregated |>
  filter(team == "Astralis"),
  prop = 0.8,
  strata = "map")

astralis.model <-
  glm(
    formula = map_winner ~ mean_kills + mean_assists + mean_deaths +
      mean_hs + mean_fkdiff + team_rank + starting_ct:map,
    data = training(astralis.split),
    family = binomial
  )
```

Optimalizovaný model

```
set.seed(487)
astralis.split <- initial_split(data.aggregated |>
  filter(team == "Astralis"),
  prop = 0.8,
  strata = "map")

astralis.model <- glm(
  formula = map_winner ~ mean_kills + mean_deaths,
  data = training(astralis.split),
  family = binomial
)
```

Uložení LaTeX tabulky

```
save_bc_model(
  .model = astralis.model,
  .file = "Astralis_model_opt.tex",
  .cap = "Výstup z programu R pro optimalizovaný pro model týmu pro tým Astralis",
  .lab = NA
)
```

Matice záměn

```
astralis.matrix <- get_matrix(
  .model = astralis.model,
  .split = astralis.split
)
```

Uložení LaTeX tabulky

```
save_bc_matrix(
  .table = astralis.matrix$table,
  .file = "Astralis_matice_opt.tex",
  .cap = "Matice záměn pro model týmu pro tým Astralis",
  .lab = NA
)

save_bc_matrix_output(
  .matrix = astralis.matrix,
  .file = "Astralis_stats_opt.tex",
  .cap = "Vybrané statistiky z matice záměn pro model týmu pro tým Astralis",
  .lab = NA
)
```

Model pro tým Sprout

Neotimalizovaný model

```
set.seed(8741)
sprout.split <- initial_split(data.aggregated |>
  filter(team == "Sprout"),
  prop = 0.8,
  strata = "map")

sprout.model <-
  glm(
    formula = map_winner ~ mean_kills + mean_assists + mean_deaths +
      mean_hs + mean_fkdif + team_rank + starting_ct:map,
    data = training(sprout.split),
    family = binomial
  )
```

Optimalizovaný model

```
set.seed(326598)
sprout.split <- initial_split(data.aggregated |>
  filter(team == "Sprout"),
  prop = 0.8,
```

```

                                strata = "map")

sprout.model <- glm(
  formula = map_winner ~ mean_kills + mean_deaths,
  data = training(sprout.split),
  family = binomial
)

```

Uložení LaTeX tabulky

```

save_bc_model(
  .model = sprout.model,
  .file = "Sprout_model_opt.tex",
  .cap = "Výstup z programu R pro optimalizovaný pro model týmu pro tým Sprout",
  .lab = NA
)

```

Matice záměn

```

sprout.matrix <- get_matrix(
  .model = sprout.model,
  .split = sprout.split
)

```

Uložení LaTeX tabulky

```

save_bc_matrix(
  .table = sprout.matrix$table,
  .file = "Sprout_matice_opt.tex",
  .cap = "Matice záměn pro model týmu pro tým Sprout",
  .lab = NA
)

save_bc_matrix_output(
  .matrix = sprout.matrix,
  .file = "Sprout_stats_opt.tex",
  .cap = "Vybrané statistiky z matice záměn pro model týmu pro tým Sprout",
  .lab = NA
)

```

Porovnání statistik týmových modelů

```

# Vytvoření statistik z matice záměn pro všechny týmy
m1 <-
  bind_rows(
    tibble(
      statistika = names(global.matrix$overall),
      `Celkový model` = global.matrix$overall
    ),
    tibble(
      statistika = names(global.matrix$byClass),
      `Celkový model` = global.matrix$byClass
    )
  ) |>
  filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
  mutate(statistika = c("Přesnost", "Senzitivita", "Specificita"))

# Vytvoření statistik z matice záměn pro tým Astralis
m2 <-
  bind_rows(
    tibble(
      statistika = names(astralis.matrix$overall),
      `Tým Astralis` = astralis.matrix$overall
    ),
    tibble(
      statistika = names(astralis.matrix$byClass),
      `Tým Astralis` = astralis.matrix$byClass
    )
  ) |>
  filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
  mutate(statistika = c("Přesnost", "Senzitivita", "Specificita"))

# Vytvoření statistik z matice záměn pro tým Sprout
m3 <-
  bind_rows(
    tibble(
      statistika = names(sprout.matrix$overall),
      `Tým Sprout` = sprout.matrix$overall
    ),
    tibble(
      statistika = names(sprout.matrix$byClass),
      `Tým Sprout` = sprout.matrix$byClass
    )
  ) |>
  filter(statistika %in% c("Accuracy", "Sensitivity", "Specificity")) |>
  mutate(statistika = c("Přesnost", "Senzitivita", "Specificita"))

m123 <-
  cbind(
    m1,
    m2 |> select(-statistika),
    m3 |> select(-statistika)
  )

save_bc_table(

```

```
.tibble = m123,  
.file = "Tymy_porovnani.text",  
.cap = "Porovnání statistik pro týmové modely",  
.lab = NA  
)
```