tejseth / nfl-r-tutorials Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master

nfl-r-tutorials / mfans-rf-xgboost.R

tejseth random forest and xgboost tutorial done History

1 contributor

203 lines (165 sloc) | 6.39 KB

```r
# Tutorial 1: Introduction to nflfastR: https://www.youtube.com/watch?v=uw6PH1YiCtI&t=155s
# Tutorial 2: Data Viz With nflfastr: https://www.youtube.com/watch?v=tGaXtAAt6-4&t=255s
# Tutorial 3: Modeling I: https://www.youtube.com/watch?v=J4p8ZfYW5Oo&t=200s
# Tutorial 4: Modeling II (Today): https://github.com/tejseth/nfl-r-tutorials/blob/master/mfans-rf-xgboost.R

# Load packages
library(tidyverse)
library(nflfastR)
library(ggthemes)
library(ranger)
library(vip)
library(caret)
library(xgboost)
library(ggimage)
options(scipen = 9999)

# Make our custom package
```

```r
theme_reach <- function() {
  theme_fivethirtyeight() +
    theme(
      legend.position = "none",
      plot.title = element_text(size = 22, hjust = 0.5, face = "bold"),
      plot.subtitle = element_text(size = 18, hjust = 0.5),
      plot.caption = element_text(size = 16),
      axis.title.x = element_text(size=18),
      axis.title.y = element_text(size=18),
      axis.text = element_text(size = 14),
      strip.text = element_text(size = 16, face = "bold"),
      legend.text = element_text(size = 14)
    )
}

# Load in play-by-play data
pbp <- load_pbp(2014:2021)

# Check what type of plays happen on 4th down
pbp %>%
  filter(down == 4) %>%
  group_by(play_type) %>%
  tally(sort = T)

# Get 4th downs
fourth_downs <- pbp %>%
  filter(down == 4, !play_type %in% c("no_play", "qb_kneel", NA)) %>%
  mutate(went_for_it = ifelse(play_type %in% c("pass", "run"), 1, 0)) %>%
  select(posteam, defteam, home_team, season, week, game_id, play_id, desc,
         play_type, down, yardline_100, ydstogo, half_seconds_remaining, wp,
         wpa, score_differential, ep, epa, temp, wind, went_for_it) %>%
  filter(!is.na(epa))

# Check for NA's
colSums(is.na(fourth_downs))

fourth_downs <- fourth_downs %>%
  mutate(temp = ifelse(is.na(temp), 70, temp),
         wind = ifelse(is.na(wind), 0, wind))

# Select the data we want for the model
model_data <- fourth_downs %>%
  select(went_for_it, yardline_100, ydstogo, half_seconds_remaining, wp,
         score_differential, ep, temp, wind, season)

# Build our random forest
rf_4th <- ranger(went_for_it ~ ., data = model_data,
                 num.trees = 100, importance = "impurity")

# Check variable importance of random forest
vip(rf_4th) + theme_reach()

# Make a grid for tuning
dim(model_data)
rf_grid <- expand.grid(mtry = seq(2, 8, by = 1),
              splitrule = "variance") # For classification

set.seed(2014) # go lions

# Use the tuning grid
rf_4th_tune <-
  train(went_for_it ~ ., data = model_data,
        method = "ranger", num.trees = 100,
        trControl = trainControl(method = "cv", number = 5),
        tuneGrid = rf_grid)

# Get the results from the best tune
rf_4th_tune$bestTune

# Remake random forest with tuning parameters
rf_4th_best <- ranger(went_for_it ~ ., data = model_data,
                      num.trees = 100, importance = "impurity",
                      mtry = 5)

```

```r
# Get predictions
rf_preds <- data.frame(predict(rf_4th_best, data.frame(model_data))$predictions)

names(rf_preds)

rf_preds <- rf_preds %>%
  rename(exp_go = predict.rf_4th_best..data.frame.model_data...predictions)

# Bind the original dataset and predictions together
fourth_downs_rf_projs <- cbind(fourth_downs, rf_preds)

fourth_downs_rf_projs <- fourth_downs_rf_projs %>%
  mutate(go_over_expected = went_for_it - exp_go)

# Check 2021 stats
rf_team_stats <- fourth_downs_rf_projs %>%
  filter(season == 2021) %>%
  group_by(posteam) %>%
  summarize(avg_gooe = 100*mean(go_over_expected),
            wpa = 100*mean(wpa)) %>%
  left_join(teams_colors_logos, by = c("posteam" = "team_abbr"))

# Make graph
rf_team_stats %>%
  ggplot(aes(x = avg_gooe, y = wpa)) +
  geom_image(aes(image = team_logo_espn), asp = 16/9, size = 0.05) +
  theme_reach() +
  labs(x = "Go For It Rate Over Expected",
       y = "WPA Added on 4th Downs",
       title = "Go For It Rate Over Expected and WPA on 4th Downs in 2021",
       subtitle = "WPA = Win Probability Added",
       caption = "By Tej Seth | @tejfbanalytics | M-FANS")
ggsave('go-oe.png', width = 15, height = 10, dpi = "retina")

############################################################################

# Make xgboost grid
```

```r
xgboost_tune_grid <- expand.grid(nrounds = seq(from = 20, to = 200, by = 20),
                                 eta = c(0.025, 0.05, 0.1, 0.3), gamma = 0,
                                 max_depth = c(1, 2, 3, 4), colsample_bytree = 1,
                                 min_child_weight = 1, subsample = 1)
xgboost_tune_control <- trainControl(method = "cv", number = 5, verboseIter = FALSE)
set.seed(2011) # go lions

# Tune xgboost grid
xgb_tune <- train(x = as.matrix(dplyr::select(model_data, -went_for_it)),
                  y = model_data$went_for_it, trControl = xgboost_tune_control,
                  tuneGrid = xgboost_tune_grid,
                  objective = "reg:squarederror", method = "xgbTree",
                  verbose = TRUE)

# Get the best tune
xgb_tune$bestTune

# Set xgboost parameters
nrounds <- 100
params <-
  list(
    booster = "gbtree",
    objective = "binary:logistic",
    eval_metric = c("logloss"),
    eta = 0.025,
    gamma = 5,
    subsample = 0.8,
    colsample_bytree = 0.8,
    max_depth = 4,
    min_child_weight = 6,
    base_score = mean(model_data$went_for_it)
  )

seasons <- seq(2014, 2021, 1)

# Make the holdout model
cv_results <- map_dfr(seasons, function(x) {
  test_data <- model_data %>%
    filter(season == x) %>%
    select(-season)
  train_data <- model_data %>%
    filter(season != x) %>%
    select(-season)

  full_train <- xgboost::xgb.DMatrix(model.matrix(~ . + 0, data = train_data %>% select(-went_for_it)),
                                     label = train_data$went_for_it
  )
  xg_4th <- xgboost::xgboost(params = params, data = full_train, nrounds = nrounds, verbose = 2)

  preds <- as.data.frame(
    matrix(predict(xg_4th, as.matrix(test_data %>% select(-went_for_it))))
  ) %>%
    dplyr::rename(exp_go = V1)

  cv_data <- bind_cols(test_data, preds) %>% mutate(season = x)
  return(cv_data)
})

# Get the predictions
xg_preds <- cv_results %>% select(exp_go)

# Put it all together
fourth_downs_xg_projs <- cbind(fourth_downs, xg_preds)

fourth_downs_xg_projs <- fourth_downs_xg_projs %>%
  mutate(go_oe = went_for_it - exp_go)

# Check the team stats from xgboost
xg_team_stats <- fourth_downs_xg_projs %>%
  filter(season == 2021) %>%
  group_by(posteam) %>%
  summarize(avg_gooe = 100*mean(go_oe),
            wpa = 100*mean(wpa))

```