Process Play by Play Data from nflfastR

Joseph Director

3/2/2022

Introduction

The following R script is designed to load and clean play by play data for the 2019 season from the `nflfastR` package. This produces an exportable `.csv` file for the purpose of analyzing and fitting models.
Load Libraries

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(nflfastR)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

Load Data

```
options(nflreadr.verbose = FALSE)
pbp <- load_pbp(2019)
```

Check Structure

```
dim(pbp)
```

```
## [1] 48034   372
```

Currently, there are 48034 rows representing each play in the 2019 season and 372 columns representing various features related to a particular play. Get a preview of the data:

```
head(pbp)
```

```
## # A tibble: 6 x 372
##   play_id game_id old_game_id home_team away_team season_type  week
posteam
##     <dbl> <chr>   <chr>       <chr>     <chr>     <chr>       <int> <chr>
## 1       1 2019_0… 2019090804  MIN       ATL       REG             1 <NA>
## 2      36 2019_0… 2019090804  MIN       ATL       REG             1 ATL
## 3      51 2019_0… 2019090804  MIN       ATL       REG             1 ATL
## 4      79 2019_0… 2019090804  MIN       ATL       REG             1 ATL
## 5     100 2019_0… 2019090804  MIN       ATL       REG             1 ATL
## 6     121 2019_0… 2019090804  MIN       ATL       REG             1 ATL
## # … with 364 more variables: posteam_type <chr>, defteam <chr>,
## #   side_of_field <chr>, yardline_100 <dbl>, game_date <chr>,
## #   quarter_seconds_remaining <dbl>, half_seconds_remaining <dbl>,
## #   game_seconds_remaining <dbl>, game_half <chr>, quarter_end <dbl>,
## #   drive <dbl>, sp <dbl>, qtr <dbl>, down <dbl>, goal_to_go <dbl>, time
<chr>,
## #   yrdln <chr>, ydstogo <dbl>, ydsnet <dbl>, desc <chr>, play_type <chr>,
## #   yards_gained <dbl>, shotgun <dbl>, no_huddle <dbl>, qb_dropback <dbl>,
## #   qb_kneel <dbl>, qb_spike <dbl>, qb_scramble <dbl>, pass_length <chr>,
## #   pass_location <chr>, air_yards <dbl>, yards_after_catch <dbl>,
## #   run_location <chr>, run_gap <chr>, field_goal_result <chr>,
## #   kick_distance <dbl>, extra_point_result <chr>, two_point_conv_result
<chr>,
## #   home_timeouts_remaining <dbl>, away_timeouts_remaining <dbl>,
## #   timeout <dbl>, timeout_team <chr>, td_team <chr>, td_player_name
<chr>,
## #   td_player_id <chr>, posteam_timeouts_remaining <dbl>,
## #   defteam_timeouts_remaining <dbl>, total_home_score <dbl>,
## #   total_away_score <dbl>, posteam_score <dbl>, defteam_score <dbl>,
## #   score_differential <dbl>, posteam_score_post <dbl>,
## #   defteam_score_post <dbl>, score_differential_post <dbl>,
## #   no_score_prob <dbl>, opp_fg_prob <dbl>, opp_safety_prob <dbl>,
## #   opp_td_prob <dbl>, fg_prob <dbl>, safety_prob <dbl>, td_prob <dbl>,
## #   extra_point_prob <dbl>, two_point_conversion_prob <dbl>, ep <dbl>,
## #   epa <dbl>, total_home_epa <dbl>, total_away_epa <dbl>,
## #   total_home_rush_epa <dbl>, total_away_rush_epa <dbl>,
## #   total_home_pass_epa <dbl>, total_away_pass_epa <dbl>, air_epa <dbl>,
## #   yac_epa <dbl>, comp_air_epa <dbl>, comp_yac_epa <dbl>,
## #   total_home_comp_air_epa <dbl>, total_away_comp_air_epa <dbl>,
## #   total_home_comp_yac_epa <dbl>, total_away_comp_yac_epa <dbl>,
## #   total_home_raw_air_epa <dbl>, total_away_raw_air_epa <dbl>,
## #   total_home_raw_yac_epa <dbl>, total_away_raw_yac_epa <dbl>, wp <dbl>,
## #   def_wp <dbl>, home_wp <dbl>, away_wp <dbl>, wpa <dbl>, vegas_wpa
<dbl>,
```

```
## #   vegas_home_wpa <dbl>, home_wp_post <dbl>, away_wp_post <dbl>,
## #   vegas_wp <dbl>, vegas_home_wp <dbl>, total_home_rush_wpa <dbl>,
## #   total_away_rush_wpa <dbl>, total_home_pass_wpa <dbl>,
## #   total_away_pass_wpa <dbl>, air_wpa <dbl>, …
```

Filter Rows

First, filter rows to reflect the goals of the study. Only regular season games are being considered as well as only plays that are either a run or a pass (this excludes special teams plays, pre-snap penalties, pre-snap timeouts, spikes, kneels, and two point conversions/extra points).

```
pbp  <- pbp %>%
  filter(play_type == 'run' | play_type == 'pass') %>%
  filter(season_type ==  'REG') %>%
  filter(is.na(two_point_conv_result))

# check dimension after filtering
dim(pbp)
```

```
## [1] 32047   372
```

Filter Columns

Now, filter raw columns to be included in the analysis or for further processing. Factors included at this stage consist of basic pre-snap in-game information (i.e. down, yards to gain, time remaining, etc.) that would be immediately available to the coaching staff of a given team. Other features like weather, time of year, and whether the game was played indoors or out are also included. The yards gained on the play are included so cumulative totals and tendencies can be added for each team.

```
pbp <- pbp %>%
  select(play_id, game_id, posteam_type, posteam, defteam, yardline_100,
half_seconds_remaining,
         game_half, down, ydstogo, play_type, shotgun, no_huddle,
posteam_timeouts_remaining,
         defteam_timeouts_remaining, score_differential, roof, temp, wind,
yards_gained)
```

Handle NAs

The temperature and wind are NA if the game was played inside a dome. Set the wind to 0 in these cases and the temperature to 72 degrees Fahrenheit (the usual temperature that is set for indoor games).

```
pbp <- pbp %>%
  mutate_at(vars(wind), ~replace_na(.,0)) %>%
  mutate_at(vars(temp), ~replace_na(.,72))
```

One-Hot Encode Categorical Features

Categorical feature variables must be presented as one-hot encoded columns in order to be correctly interpreted by models late in the analysis. The target variable (whether the play is a run or a pass) is also encoded where a pass is a "positive case".

```
pbp <- pbp %>%
  mutate(posteam_home = ifelse(posteam_type == "home", 1, 0),
         frst_d = ifelse(down == 1, 1, 0),
         scnd_d = ifelse(down == 2, 1, 0),
         thrd_d = ifelse(down == 3, 1, 0),
         frth_d = ifelse(down == 4, 1, 0),
         half1 = ifelse(game_half == "Half1", 1, 0),
         dome = ifelse(roof == "dome", 1, 0),
         outdoors = ifelse(roof == "outdoors", 1, 0),
         closed = ifelse(roof == "closed", 1, 0),
         open = ifelse(roof == "open", 1, 0),
         pass = ifelse(play_type == "pass", 1, 0),
         run = ifelse(play_type == "run", 1, 0)) %>%
  select(-posteam_type, -game_half, -roof, -play_type, -down)
```

The offensive and defensive team will also be considered as a categorical feature. These columns will need to be one hot encoded as well.

```
# use the dummyVars function from caret since the team columns have many
possible values
posteam_dummys <- data.frame(predict(dummyVars("~ posteam", data = pbp,
fullRank = T), newdata = pbp))
defteam_dummys <- data.frame(predict(dummyVars("~ defteam", data = pbp,
fullRank = T), newdata = pbp))

pbp <- cbind(pbp, posteam_dummys, defteam_dummys)
```

Feature Engineering

Cumulative Offensive Run/Pass Tendencies

The defensive team's coaching staff will have a general sense of the opposition's play call tendency as the year progresses. This aspect can be built in to the feature space in two ways; through including the offensive teams overall effectiveness at running/passing (total yards gained per play) and their overall pass to run ratio. Note that both these values are lagged as to not include information that would be gained at the end of a given play.

```
## Add columns for cumulative run or pass yards gained
pbp <- pbp %>%
  # categorize yards gained for pass or run plays
  mutate(pass_yds = ifelse(pass == 1, yards_gained, 0),
         run_yds = ifelse(pass == 0, yards_gained, 0)) %>%

  # group by each team and accumulate total pass or run yards gained
```

```
  group_by(posteam) %>%
  mutate(cum_pass_yds = lag(cummean(pass_yds), order_by = posteam),
         cum_run_yds = lag(cummean(run_yds), order_by = posteam)) %>%

  # remove NAs at zero lag
  mutate_at(vars(cum_pass_yds), ~replace_na(.,0)) %>%
  mutate_at(vars(cum_run_yds), ~replace_na(.,0)) %>%

  # remove unwanted columns
  select(-pass_yds, -run_yds)

## Add columns for cumulative pass/run ratio
pbp <- pbp %>%
  # group by team and accumulate pass and run plays
  group_by(posteam) %>%
  mutate(cum_passes = lag(cumsum(pass), order_by = posteam),
         cum_runs = lag(cumsum(run), order_by = posteam)) %>%

  # calculate ratios
  mutate(ptr_ratio = cum_passes / (cum_runs + cum_passes)) %>%

  # remove NAs at zero lag
  mutate_at(vars(ptr_ratio), ~replace_na(.,0)) %>%

  # remove unwanted columns
  select(-cum_passes, -cum_runs)
```

Cumulative Defensive Yards Allowed

The overall effectiveness of how the defensive team handles either the run or the pass will be an important factor for the offensive teams play call decision. This can be added to the feature space in a similar way to the cumulative offensive totals; yards allowed on runs or passes per run or pass.

```
## Add columns for cumulative run or pass yards gained
pbp <- pbp %>%
  # categorize yards gained for pass or run plays
  mutate(pass_yds_all = ifelse(pass == 1, yards_gained, 0),
         run_yds_all = ifelse(pass == 0, yards_gained, 0)) %>%

  # group by each team and accumulate total pass or run yards gained
  group_by(defteam) %>%
  mutate(cum_pass_yds_all = lag(cummean(pass_yds_all), order_by = defteam),
         cum_run_yds_all = lag(cummean(run_yds_all), order_by = defteam)) %>%

  # remove NAs at zero lag
  mutate_at(vars(cum_pass_yds_all), ~replace_na(.,0)) %>%
  mutate_at(vars(cum_run_yds_all), ~replace_na(.,0)) %>%
```

```
  # remove unwanted columns
  select(-pass_yds_all, -run_yds_all, -run, -yards_gained)
```

Preview the Final Data Frame

```
head(pbp)
```

```
## # A tibble: 6 x 92
## # Groups:   defteam [2]
##   play_id game_id posteam defteam yardline_100 half_seconds_re… ydstogo
shotgun
##     <dbl> <chr>   <chr>   <chr>          <dbl>            <dbl>   <dbl>
<dbl>
## 1      51 2019_0… ATL     MIN               75             1800      10
0
## 2      79 2019_0… ATL     MIN               83             1760      18
0
## 3     100 2019_0… ATL     MIN               79             1721      14
1
## 4     185 2019_0… MIN     ATL               31             1652      20
0
## 5     214 2019_0… MIN     ATL               23             1617      12
0
## 6     277 2019_0… ATL     MIN               84             1604      10
0
## # … with 84 more variables: no_huddle <dbl>, posteam_timeouts_remaining
<dbl>,
## #   defteam_timeouts_remaining <dbl>, score_differential <dbl>, temp
<dbl>,
## #   wind <dbl>, posteam_home <dbl>, frst_d <dbl>, scnd_d <dbl>, thrd_d
<dbl>,
## #   frth_d <dbl>, half1 <dbl>, dome <dbl>, outdoors <dbl>, closed <dbl>,
## #   open <dbl>, pass <dbl>, posteamATL <dbl>, posteamBAL <dbl>,
## #   posteamBUF <dbl>, posteamCAR <dbl>, posteamCHI <dbl>, posteamCIN
<dbl>,
## #   posteamCLE <dbl>, posteamDAL <dbl>, posteamDEN <dbl>, posteamDET
<dbl>,
## #   posteamGB <dbl>, posteamHOU <dbl>, posteamIND <dbl>, posteamJAX <dbl>,
## #   posteamKC <dbl>, posteamLA <dbl>, posteamLAC <dbl>, posteamLV <dbl>,
## #   posteamMIA <dbl>, posteamMIN <dbl>, posteamNE <dbl>, posteamNO <dbl>,
## #   posteamNYG <dbl>, posteamNYJ <dbl>, posteamPHI <dbl>, posteamPIT
<dbl>,
## #   posteamSEA <dbl>, posteamSF <dbl>, posteamTB <dbl>, posteamTEN <dbl>,
## #   posteamWAS <dbl>, defteamATL <dbl>, defteamBAL <dbl>, defteamBUF
<dbl>,
## #   defteamCAR <dbl>, defteamCHI <dbl>, defteamCIN <dbl>, defteamCLE
<dbl>,
## #   defteamDAL <dbl>, defteamDEN <dbl>, defteamDET <dbl>, defteamGB <dbl>,
## #   defteamHOU <dbl>, defteamIND <dbl>, defteamJAX <dbl>, defteamKC <dbl>,
## #   defteamLA <dbl>, defteamLAC <dbl>, defteamLV <dbl>, defteamMIA <dbl>,
## #   defteamMIN <dbl>, defteamNE <dbl>, defteamNO <dbl>, defteamNYG <dbl>,
```

```
## #   defteamNYJ <dbl>, defteamPHI <dbl>, defteamPIT <dbl>, defteamSEA
<dbl>,
## #   defteamSF <dbl>, defteamTB <dbl>, defteamTEN <dbl>, defteamWAS <dbl>,
## #   cum_pass_yds <dbl>, cum_run_yds <dbl>, ptr_ratio <dbl>,
## #   cum_pass_yds_all <dbl>, cum_run_yds_all <dbl>
```

Export Data Frame to CSV

```
setwd("~/Documents/Masters_Project/NFL-Play-Call-Prediction-with-LSTM-Neural-
Networks/data")
write.csv(pbp, "processed_pbp.csv", row.names = F)
```