R Notebook

```
library(tidyverse)
```

```
## Warning: le package 'tidyverse' a été compilé avec la version R 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.5     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.4     v forcats 0.5.1
```

```
## Warning: le package 'tibble' a été compilé avec la version R 4.1.3
```

```
## Warning: le package 'tidyr' a été compilé avec la version R 4.1.2
```

```
## Warning: le package 'readr' a été compilé avec la version R 4.1.3
```

```
## Warning: le package 'purrr' a été compilé avec la version R 4.1.3
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.1.3
```

```
## Warning: le package 'forcats' a été compilé avec la version R 4.1.2
```

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

I - Avant la modélisation

1 - Importation des fichiers

```
#1-1 Directement avec les liens URL--------
# match_stats_2017 <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/match_stat
# match_stats_1991_2016 <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/match_
# match_scores_1991_2016 <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/matc
# match_scores_2017 <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/match_sco
# joueurs <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/player_overviews_un
# tournois <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/tournaments_1877-2
# classements_1973_2017 <- read.csv2("https://datahub.io/sports-data/atp-world-tour-tennis-data/r/ranki
# #le fichier classement fait près de 300 Mo
```

```r
#1-2 Si Les fichiers plats sont stockés sur le disque dur dans un dossier data
match_stats_2017 <- read.csv2("data/match_stats_2017_unindexed_csv.csv",sep = ",")
match_stats_1991_2016 <- read.csv2("data/match_stats_1991-2016_unindexed_csv.csv",sep = ",")
joueurs <- read.csv2("data/player_overviews_unindexed_csv.csv",sep = ",")
tournois <- read.csv2("data/tournaments_1877-2017_unindexed_csv.csv",sep = ",")
match_scores_1991_2016 <- read.csv2("data/match_scores_1991-2016_unindexed_csv.csv",sep = ",")
match_scores_2017 <- read.csv2("data/match_scores_2017_unindexed_csv.csv",sep = ",")
classements_1973_2017 <- read.csv2("data/rankings_1973-2017_csv.csv",sep = ",")
```

2 - Preprocessing

2-1 Nettoyage des fichiers match_stats et match_scores

```r
#2 - Nettoyage des fichiers match_stats et match_scores

#2-1 Match_stats

#Fusion des fichiers sur les matches de 1991 à 2017
match_stats <- match_stats_1991_2016 %>% bind_rows(match_stats_2017)
rm(match_stats_1991_2016,match_stats_2017)

#on enlève les doublons de matches en utilisant l'url comme identifiant
x <- unique(match_stats[duplicated(match_stats$match_stats_url_suffix),"match_stats_url_suffix"]) #list
match_stats <- match_stats %>% filter(!(match_stats_url_suffix %in% x))

#2-2 Match_scores

match_scores <- match_scores_1991_2016 %>% bind_rows(match_scores_2017)
rm(match_scores_1991_2016,match_scores_2017)
x <- unique(match_scores[duplicated(match_scores$match_stats_url_suffix),"match_stats_url_suffix"]) #li
match_scores <- match_scores %>% filter(!(match_stats_url_suffix %in% x))
#il manque les url des 127 matches de l'us open 2017 dans le fichier des scores...à compléter

#2- Ajout d'une variable d'ordre des matches--------
match_stats <- match_stats %>%
  mutate(annee = as.numeric(substr(match_id,1,4)),
         toto = substr(match_stats_url_suffix,21,26)) %>%
  arrange(annee,tourney_order,desc(toto)) %>%
  cbind(1:nrow(match_stats)) %>%
  rename(numero_ordre = "1:nrow(match_stats)")

#Remarque : Il n'y a pas toujours les données des matchs de qualification

# 3 Filtre des matches pour lesquels on ne retrouve pas les scores--------
match_stats <- match_stats %>% semi_join(match_scores,by="match_stats_url_suffix")
#sans l'us open, on est sur une base de 95 610 matchs

rm(x)
```

Ajout de variables dans la base match_stats :

- pourcentage de premier service
- pourcentage de points gagnés au premier service
- pourcentage de points gagnés au deuxième service
- pourcentage de doubles fautes (sur le nombre de services)
- poucentage d'aces
- pourcentage de balles de breaks sur les points retournés
- pourcentage de balles de breaks sur les points servis

```
#Ajout de stats :
match_stats <- match_stats %>%
  mutate(winner_pct_first_serves_in = winner_first_serves_in/winner_first_serves_total*100,
         winner_pct_first_serves_points_won = winner_first_serve_points_won/winner_first_serves_total*1
         winner_pct_second_serves_points_won = winner_second_serve_points_won/winner_second_serve_point
         winner_pct_aces = winner_aces/winner_service_points_total*100,
         winner_pct_double_faults = winner_double_faults/winner_service_points_total*100,
         winner_pct_break_points_serve_total = winner_break_points_serve_total/winner_service_points_to
         winner_pct_break_points_return_total = winner_break_points_return_total/winner_return_points_t
         loser_pct_first_serves_in = loser_first_serves_in/loser_first_serves_total*100,
         loser_pct_first_serves_points_won = loser_first_serve_points_won/loser_first_serves_total*100,
         loser_pct_second_serves_points_won = loser_second_serve_points_won/loser_second_serve_points_t
         loser_pct_aces = loser_aces/loser_service_points_total*100,
         loser_pct_double_faults = loser_double_faults/loser_service_points_total*100,
         loser_pct_break_points_serve_total = loser_break_points_serve_total/loser_service_points_total
         loser_pct_break_points_return_total = loser_break_points_return_total/loser_return_points_total
```

2 -2 ajout d'information à partir de la base des joueurs

```
# 4-1 Ajout d'informations à partir de la base des joueurs (taille, poids, âge)-------

don <- match_stats %>%
  left_join(match_scores,by="match_stats_url_suffix") %>%
  select(match_stats_url_suffix,winner_player_id,loser_player_id,tourney_year_id,numero_ordre,match_sta
  rename(match_id = match_id.x) %>%
  mutate(year = as.integer(substr(match_id,1,4)))

#Age au moment du tournoi (en age révolu : En 2017, un joueur né en 1988 a 29 ans)
don <- don %>%
  left_join(joueurs,by = c("winner_player_id"="player_id")) %>%
  left_join(joueurs,by=c("loser_player_id" = "player_id")) %>%
  mutate(winner_age = year - birth_year.x, loser_age = year - birth_year.y) %>%
  rename(winner_weight = weight_kg.x,
         loser_weight = weight_kg.y,
         winner_height = height_cm.x,
         loser_height = height_cm.y) %>%
  select(match_id,numero_ordre,match_stats_url_suffix,winner_player_id,winner_weight,winner_height,winn
```

2 - 3 ajout d'information à partir de la base du classement

- ajout du classement du gagnant et du perdant au moment du match

A partir de la base des classements, on affecte un classement moyen à chaque joueur pour chaque mois de l'année.

```
#Calcul de la moyenne de classement par joueur, par mois et par année entre 1991 et 2017
toto <- classements_1973_2017 %>% filter(week_year > 1990) %>% group_by(player_id,week_year,week_month)
  summarise(classement = round(mean(rank_number))) %>% ungroup()
```

```
## `summarise()` has grouped output by 'player_id', 'week_year'. You can override
## using the `.groups` argument.
```

```
#ajout d'un identifiant dans la base du classement
toto <- toto %>%
  mutate(id_classement = paste(player_id,as.character(week_year),as.character(week_month),sep = "-")) %
  select(id_classement,classement)

#ajout d'un identifiant dans la base des match pour les winners et les losers
toto2 <- don %>%
  left_join(match_scores[,c("tourney_year_id","match_stats_url_suffix")],by="match_stats_url_suffix") %
  left_join(tournois[,c("tourney_year_id","tourney_year","tourney_month")],by="tourney_year_id") %>%
  mutate(winner_id_classement = paste(winner_player_id,as.character(tourney_year),as.character(tourney_
         loser_id_classement = paste(loser_player_id,as.character(tourney_year),as.character(tourney_mo


toto3 <- toto2 %>% left_join(toto,by=c("winner_id_classement" = "id_classement")) %>% rename(winner_cla
toto4 <- toto3 %>% left_join(toto,by=c("loser_id_classement" = "id_classement")) %>% rename(loser_classe

don <- toto4
rm(toto,toto2,toto3,toto4)
```

2- 4 ajout d'informations à partir des stats matches

- 1 - pourcentage de matches gagnés sur les x matches précédents
- 2 - pourcentage de fautes directes/ensemble des points perdus sur les x matches précédents
- 3 - pourcentage de coups gagnants/pointsgagnés sur les x matches précédents
- 4 - pourcentage de points gagnés au premier service
- 5 - pourcentage de points gagnés au deuxième service

```
don_winner <- don %>%
  select(contains(match=c("match_stats_url_suffix", "numero_ordre", "winner"))) %>%
  rename_with(function(x){sub('winner_', '', x)}) %>%
  mutate(winner_or_loser  = "winner")

don_loser <- don %>%
  select(contains(match=c("match_stats_url_suffix", "numero_ordre", "loser"))) %>%
  rename_with(function(x){sub('loser_', '', x)}) %>%
  mutate(winner_or_loser  = "loser")

don_pivotted <- bind_rows(don_winner, don_loser)
```

```r
# ....

winner_stats <- match_stats %>%
  select(contains(match=c("match_stats_url_suffix", "winner"))) %>%
  rename_with(function(x){sub('winner_', '', x)}) %>%
  mutate(winner_or_loser  = "winner")

loser_stats <- match_stats %>%
  select(contains(match=c("match_stats_url_suffix", "loser"))) %>%
  rename_with(function(x){sub('loser_', '', x)}) %>%
  mutate(winner_or_loser  = "loser")

match_player_stats <- bind_rows(winner_stats, loser_stats)

don_with_stats <- don_pivotted %>%
  left_join(match_player_stats, by=c('match_stats_url_suffix', 'winner_or_loser'))

#fonction qui calcule une moyenne sur une variable X à partir de N matches précédents :
# X c'est la variable à tester,
# N c'est le nombre de matches précédents sur laquelle on calcule la stat

#Calcul de la moyenne de x sur les N derniers matches
moy_stat_last_matchs <- function(x, N){
  if(length(x)==1){
    return(NA)
  }
  rowMeans(sapply(1:N, function(i) dplyr::lag(x, n=i)))
}

#calcul du nombre de matches gagnés sur les N derniers matches
nb_last_matchs <- function(x, N){
  a <- if_else(x == "winner",1,0)
  if(length(a)==1){
    return(NA)
  }
  rowSums(sapply(1:N, function(i) dplyr::lag(a, n=i)))
}

# TODO add stats

don_with_stats2 <- don_with_stats %>%
  arrange(player_id, numero_ordre) %>%
  group_by(player_id) %>%
  # mutate(nb_match_won = dplyr::lag(cumsum(winner_or_loser=="winner"))) %>%
  mutate(nb_match_won_last_5 = nb_last_matchs(winner_or_loser,5)) %>%
  # mutate(avg_nb_ace = dplyr::lag(cummean(aces))) %>%
  mutate(avg_pct_first_serves_in_5 = moy_stat_last_matchs(x=pct_first_serves_in, N=5)) %>%
  mutate(avg_pct_first_serves_points_won_5 = moy_stat_last_matchs(x=pct_first_serves_points_won, N=5)) %
  # mutate(avg_pct_second_serves_in_5 = moy_stat_last_matchs(x=pct_second_serves_in, N=5)) %>%
  mutate(avg_pct_second_serves_points_won_5 = moy_stat_last_matchs(x=pct_second_serves_points_won, N=5))
  mutate(avg_pct_ace_last_5 = moy_stat_last_matchs(x=pct_aces, N=5)) %>%
  mutate(avg_pct_double_faults_last_5 = moy_stat_last_matchs(x=pct_double_faults, N=5)) %>%
  # mutate(avg_nb_ace_last_3 = (dplyr::lag(aces,n=1)+dplyr::lag(aces,n=2)+dplyr::lag(aces,n=3))/3) %>%
```

```
  ungroup()


don <- don_with_stats2 %>%
  pivot_wider(id_cols = c("match_stats_url_suffix", "numero_ordre"), names_from = "winner_or_loser", va
  select(contains(c("match_stats_url_suffix","numero_ordre","weight","height","age","classement","nb_ma

rm(don_loser,don_pivotted,don_winner,don_with_stats,don_with_stats2,loser_stats,match_player_stats,winn
```

2-5 Récupération d'informations sur le tournoi

- Informations sur la surface du terrain (4 possibilités : herbe, terre battue, dur et moquette)

```
tmp <- don %>%
  left_join(match_scores[,c("match_stats_url_suffix","tourney_url_suffix")],by="match_stats_url_suffix"
  left_join(tournois[,c("tourney_url_suffix","tourney_surface","tourney_conditions")],by="tourney_url_su
  select(-contains("tourney_url"))

don <- tmp
```

2-6 Mise en forme finale de la base

- 1 - Il faut doubler la base
- 2 - Rajouter une colonne target dans chacune des base (1 et 0)
- 3 - Concaténer les bases

```
don_winner <- don %>% mutate(target = "1") %>%
  rename_with(function(x){sub('_winner', '_j1', x)}) %>%
  rename_with(function(x){sub('_loser', '_j2', x)})

don_loser_part1 <-  don %>%
  select(contains("_winner")) %>%
  rename_with(function(x){sub('_winner', '_loser', x)})

don_loser_part2 <- don %>%
  select(contains("_loser")) %>%
  rename_with(function(x){sub('_loser', '_winner', x)})

don_loser <- don_loser_part1 %>% bind_cols(don_loser_part2) %>% bind_cols(don[,c("numero_ordre","tourney
  mutate(target = "0") %>%
  rename_with(function(x){sub('_winner', '_j1', x)}) %>%
  rename_with(function(x){sub('_loser', '_j2', x)})

don <- don_winner %>% bind_rows(don_loser)

# rename_with(function(x){sub('_winner', 'j1', x)}) %>%
```

1. On ne garde que les variables qui vont servir dans la modélisation
2. On enlève toutes les valeurs manquantes (pas d'imputation pour l'instant)

3. On transforme la variable target en factor

```
#1
don <- don %>% select(target,contains(c("j1","j2")),tourney_surface,tourney_conditions)

#2
don <- na.omit(don) # on enlève toutes les valeurs manquantes

#3
don <- don %>% mutate(target = as.factor(target))

saveRDS(don,"don.rds")
```

II - Comparaison des classifications