MVA Final Project

Javier Ferrando Monsonis
Marcel Porta Valles
Mehmet Fatih ??agil

February 20, 2018

```
library(magrittr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(ggplot2)

library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(dplyr)
library(magrittr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggExtra)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(stringr)
library(FactoMineR)
```

```r
#library(kableExtra)
library(knitr)

setwd("/Users/JaviFerrando/Desktop/MLProject/")
dir <- '/Users/JaviFerrando/Desktop/MLProject/input/'

# Get data
dseeds_tournament <- fread(paste(dir,'NCAATourneySeeds.csv',sep=''))
dg_tournment <- fread(paste(dir,'NCAATourneyCompactResults.csv',sep=''))


# keep only season, daynum, win and loss team ids for the dg_tournament data
outcome_tournament <- dg_tournment %>% select(Season, DayNum, WTeamID, LTeamID)
names(outcome_tournament) <- tolower(names(outcome_tournament))

# randomize winning and losing team into team 1 and team 2 (necessary for probabilities later) and drop
outcome_tournament <- outcome_tournament %>%
  mutate(rand = runif(dim(outcome_tournament)[1]),
         team1id = ifelse(rand >= 0.5, wteamid, lteamid),
         team2id = ifelse(rand <0.5, wteamid, lteamid),
         team1win = ifelse(team1id == wteamid, 1, 0)) %>%
  select(-rand, -wteamid,-lteamid)

# Add seeding information to games:

# make seeds 1-16 without letters (except for certain seed)
dseeds_tournament <- dseeds_tournament %>%
  mutate(ranking = as.factor((str_replace(Seed, "[A-Z]",""))),
         rank_num = as.numeric(str_replace(ranking, ".[a-z]","")))
names(dseeds_tournament) <- tolower(names(dseeds_tournament))

# team 1
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(dseeds_tournament, t1_rank = ranking, t1_rank_n = rank_num, teamid, season),
    by = c("team1id"="teamid","season"="season"))

# team 2
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(dseeds_tournament, t2_rank = ranking, t2_rank_n = rank_num, teamid, season),
    by = c("team2id"="teamid","season"="season"))


# replace NA seeds
outcome_tournament <- outcome_tournament %>% mutate(t1_rank = ifelse(is.na(t1_rank), 8.5, t1_rank),
                                                    t2_rank = ifelse(is.na(t2_rank), 8.5, t2_rank),
                                                    t1_rank_n = ifelse(is.na(t1_rank_n), 8.5, t1_rank_n
                                                    t2_rank_n = ifelse(is.na(t2_rank_n), 8.5, t2_rank_n
                                                    diff_rank = t1_rank_n - t2_rank_n)
```

```
season_elos <- read.csv(paste(dir,'season_elos.csv',sep='')) %>% rename(teamid = team_id)


#Add season_elos (for t1 and t2) to outcome tournament
# Join team 1 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(season_elos,
           season,
           teamid,
           t1_season_elo = season_elo),
    by = c("team1id" = "teamid","season" = "season"))


# Join team 2 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(season_elos,
           season,
           teamid,
           t2_season_elo = season_elo),
    by = c("team2id" = "teamid","season" = "season"))


# Compute ELO probabilities for the game, and the difference in ELO scores

outcome_tournament <- outcome_tournament %>%
  mutate(elo_diff = t1_season_elo - t2_season_elo,
         elo_prob_1 = 1/(10^(-elo_diff/400)+1)
  )

################################
outcome_tournament <- outcome_tournament[outcome_tournament$season>=2003,]


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

#Add advanced statistics

seas_enrich <- fread(paste(dir,'NCAASeasonDetailedResultsEnriched.csv',sep=''))

win_stats <- seas_enrich[, .(
  Season,
  TeamID = WTeamID,
  Result = rep('W', .N),
  FGM = WFGM,
  FGA = WFGA,
  FGP = WFGM / WFGA,
  FGP2 = (WFGM - WFGM3) / (WFGA - WFGA3),
  FGM3 = WFGM3,
  FGA3 = WFGA3,
  FGP3 = WFGM3 / WFGA3,
  FTM = WFTM,
```

```
  FTA = WFTA,
  FTP = WFTM / WFTA,
  OR = WOR,
  DR = WDR,
  AST = WAst,
  TO = WTO,
  STL = WStl,
  BLK = WBlk,
  PF = WPF,
  PIE = WPIE,
  ORP = WOR / (WOR + LDR),
  DRP = WDR / (WDR + LOR),
  eFG = WeFGP,
  NetRTG = WNetRtg,
  POS = 0.96 * (WFGA + WTO + 0.44 * WFTA - WOR)
)]

los_stats <- seas_enrich[, .(
  Season,
  TeamID = LTeamID,
  Result = rep('L', .N),
  FGM = LFGM,
  FGA = LFGA,
  FGP = LFGM / LFGA,
  FGP2 = (LFGM - LFGM3) / (LFGA - LFGA3),
  FGM3 = LFGM3,
  FGA3 = LFGA3,
  FGP3 = LFGM3 / LFGA3,
  FTM = LFTM,
  FTA = LFTA,
  FTP = LFTM / LFTA,
  OR = LOR,
  DR = LDR,
  AST = LAst,
  TO = LTO,
  STL = LStl,
  BLK = LBlk,
  PF = LPF,
  PIE = LPIE,
  ORP = (LOR / (LOR + WDR)),
  DRP = LDR / (LDR + WOR),
  eFG = LeFGP,
  NetRTG = LNetRtg,
  POS = 0.96 * (LFGA + LTO + 0.44 * LFTA - LOR)
)]

stats_all <- rbindlist(list(win_stats, los_stats))


stats_season <- stats_all[, .(
  FGP = sum(FGM) / sum(FGA),
  FGP3 = sum(FGM3) / sum(FGA3),
  FTP = sum(FTM) / sum(FTA),
```

```
  ORPG = mean(OR),
  DRPG = mean(DR),
  ASPG = mean(AST),
  TOPG = mean(TO),
  STPG = mean(STL),
  #BLPG = mean(BLK),
  #PFPG = mean(PF),
  MeFG = mean(eFG),
  MNetRTG = mean(NetRTG),
  #MORP = mean(ORP),
  MPIE = mean(PIE),
  MPOS = mean(POS),
  EFG = (mean(FGM)+0.5*mean(FGM3))/mean(FGA))

  , by = c('TeamID', 'Season')]

#########################################################################################################################
#MPIE feature

# Join team 1 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(stats_season,
           Season,
           TeamID,
           t1_mpie = MPIE),
    by = c("team1id" = "TeamID","season" = "Season"))

# Join team 2 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(stats_season,
           Season,
           TeamID,
           t2_mpie = MPIE),
    by = c("team2id" = "TeamID","season" = "Season"))

################################
#Netrtg feature

# Join team 1 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(stats_season,
           Season,
           TeamID,
           t1_netrtg = MNetRTG),
    by = c("team1id" = "TeamID","season" = "Season"))

# Join team 2 data
outcome_tournament <- outcome_tournament %>%
  left_join(
    select(stats_season,
```

```
            Season,
            TeamID,
            t2_netrtg = MNetRTG),
    by = c("team2id" = "TeamID","season" = "Season"))



### Load data

#sample_submission <- read.csv(paste(dir,'SampleSubmissionStage2.csv',sep=''))#2019 every possible matc
sample_submission <- read.csv(paste(dir,'SampleSubmissionStage1.csv',sep=''))#2014-2018 every possible




##############################################################################################################
#d_ss -> same as outcome_tournament but with sample_submission format (every possible matchup)

### Join team data and ranking data

d_ss <- sample_submission


# Add season, team1id and team2id columns from sample submission ID
d_ss <- d_ss %>% mutate(season = as.numeric(gsub("(.*)_(.*)_(.*)",ID, replacement = "\\1")),
                        team1id =  as.numeric(gsub("(.*)_(.*)_(.*)",ID, replacement = "\\2")),
                        team2id =  as.numeric(gsub("(.*)_(.*)_(.*)",ID, replacement = "\\3")))

# Add rank data

# team 1
d_ss <- d_ss %>%
  left_join(
    dplyr::select(dseeds_tournament, t1_rank = ranking, t1_rank_n = rank_num, teamid, season),
    by = c("team1id"="teamid","season"="season"))

# team 2
d_ss <- d_ss %>%
  left_join(
    dplyr::select(dseeds_tournament, t2_rank = ranking, t2_rank_n = rank_num, teamid, season),
    by = c("team2id"="teamid","season"="season"))



### Join ELO rating data
#season_elos <- read.csv("../input/fivethirtyeight-elo-ratings/season_elos.csv") %>% rename(teamid = te
season_elos <- read.csv(paste(dir,'season_elos.csv',sep='')) %>% rename(teamid = team_id)

# Join team 1 data
d_ss <- d_ss %>%
  left_join(
    select(season_elos,
           season,
```

```
            teamid,
            t1_season_elo = season_elo),
    by = c("team1id" = "teamid","season" = "season"))


# Join team 2 data
d_ss <- d_ss %>%
  left_join(
    select(season_elos,
            season,
            teamid,
            t2_season_elo = season_elo),
    by = c("team2id" = "teamid","season" = "season"))


# Key differences between winner and loser
#Add elop probability

d_ss <- d_ss %>%
  mutate(elo_diff = t1_season_elo - t2_season_elo,
         elo_prob_1 = 1/(10^(-elo_diff/400)+1),
         diff_rank = t1_rank_n - t2_rank_n
  )

#PIE feature
d_ss <- d_ss[d_ss$season>=2003,]
# Join team 1 data
d_ss <- d_ss %>%
  left_join(
    select(stats_season,
            Season,
            TeamID,
            t1_mpie = MPIE),
    by = c("team1id" = "TeamID","season" = "Season"))

# Join team 2 data
d_ss <- d_ss %>%
  left_join(
    select(stats_season,
            Season,
            TeamID,
            t2_mpie = MPIE),
    by = c("team2id" = "TeamID","season" = "Season"))

################################
#Netrtg

# Join team 1 data
d_ss <- d_ss %>%
  left_join(
    select(stats_season,
            Season,
            TeamID,
```

```
            t1_netrtg = MNetRTG),
    by = c("team1id" = "TeamID","season" = "Season"))

# Join team 2 data
d_ss <- d_ss %>%
  left_join(
    select(stats_season,
           Season,
           TeamID,
           t2_netrtg = MNetRTG),
    by = c("team2id" = "TeamID","season" = "Season"))



### Make predictions based on model
train <- outcome_tournament %>% filter(season <= 2013) #Takes occurred tournament games results (team1w



test_outcome_tournament <- outcome_tournament %>% filter(season > 2013) #Test sample, target team1win
```

```
train$daynum <- NULL
train$t1_rank <- NULL
train$t2_rank <- NULL
kable(train[sample(nrow(train), 6), ][,1:10])
```

	season	team1id	team2id	team1win	t1_rank_n	t2_rank_n	diff_rank	t1_season_elo	t2_season_elo
264	2007	1266	1277	0	8	9	-1	1827.897	1852.932
518	2011	1140	1459	1	3	14	-11	1948.359	1639.349
377	2008	1390	1400	0	3	2	1	1856.137	1968.462
257	2007	1197	1310	0	1	1	0	1329.205	1533.327
88	2004	1272	1376	1	7	10	-3	1849.112	1788.532
18	2003	1386	1120	0	7	10	-3	1840.296	1722.967

```
#Train model with train data
#Add predictions to dss
#Merge d_ss with test_outcome_tournament (games that occurred) -> validation
#validation has target and Pred for every game that occurered 2014-2018
#Apply LogLoss to validation$Pred and validation$team1win
```