Data Querying & Formatting

Pranav Natarajan

15/04/2022

This Notebook contains the code to obtain the raw data required for the supervised learning problem of predicting (normalised) NBA yearly salaries.

```
# loading required packages onto workbook instance
require(nbastatR) # to query for NBA data
require(tidyr) # for data manipulation
require(dplyr) # for data manipulation
```

Getting the dictionary of the players in BREF

```
dictionary_bref_players<- dictionary_bref_players()
```

Creating the column of first season

```
dictionary_bref_players<- dictionary_bref_players %>%
  tidyr::separate(col=slugSeasonRookie,
                  into = c('yearDraft', NA),
                  remove=F)
```

Getting draft data from 1985-2021, since 1985 is the year when the 3 pt. line and modern basketball came into being.

```
draft_data<- drafts(draft_years = 1985:2021,
                    nest_data = F) #draft from 1985-86 to 2021-22 szns
```

Renaming `namePlayer` to `namePlayerBREF` in `draft_data` for ease in merging

```
draft_data<- draft_data %>% dplyr::rename(namePlayerBREF = namePlayer)
```

performing the join of `draft_data` and `dictionary_bref_players['namePlayerBREF', 'yearDraft']` into the initial feature matrix `X`

```
X<- merge.data.frame(x = draft_data,
                     y= dictionary_bref_players[, c("namePlayerBREF",
                                                    "yearDraft",
                                                    "slugPlayerBREF")],
                     by=c("namePlayerBREF", "yearDraft"), all=F)
```

relocating the `slugPlayerBREF` column to be the first from the left for easier readability, and dropping `Player_Profile_Flag`

```
X<- X %>%
  relocate(slugPlayerBREF, .before = namePlayerBREF) %>%
  select(!(PLAYER_PROFILE_FLAG))
```

Querying the BasketBallReference player bios (salaries by season)

```
bref_bios(player_ids = X$slugPlayerBREF)
```

Getting player stats of the players in X

```
player_stats<- bref_players_stats(seasons = 1985:2021,
                                  tables = c("per_game",
                                             "advanced",
                                             "per_minute"),
                                  return_message = T,
                                  join_data = T,
                                  nest_data = F)
```

Removing the columns which begin with `url` (not required at this point in the process)

```
player_stats<- player_stats %>%
  select(!starts_with("url"))# 80 variables currently
```

Sanity Check:- checking number of distinct values in the slugplayerseason column, as that is our primary key

```
nums_pl<- player_stats %>%
  count(slugPlayerSeason) %>%
  arrange(desc(n)) %>%
  head()
nums_pl # these players have faulty NBA ID's and are therefore fail integrity check
```

Noting that there are cases where there are more rows than needed, we remove those duplicate rows from the dataset.

```
# code to remove duplicates from the comprehensive_stats dataframe
player_stats<- player_stats %>%
  distinct(slugPlayerSeason,
           .keep_all = T)# nrow = 16150
```

Creating a slugPlayerSeason column for the Salaries data, and adding salary information for a player for a given season.

```
# first, creating a separate column for the yearSeason
# then, using that and the slugPlayerBREF Column to create slugPlayerSeason
```

```r
dataBREFPlayersSalaries<- dataBREFPlayersSalaries %>%
  separate(col=slugSeason, into=c("yearSeason", NA),
           sep='-',
           remove = F)

# add a 1 to yearseason column for uniformity and help in merging
dataBREFPlayersSalaries$yearSeason<- as.numeric(dataBREFPlayersSalaries$yearSeason) + 1


dataBREFPlayersSalaries<- dataBREFPlayersSalaries %>%
  unite(slugPlayerSeason, c("slugPlayerBREF",
                            "yearSeason"),
        remove = F)
```

Dropping the yearSeason column

```r
dataBREFPlayersSalaries<- dataBREFPlayersSalaries %>%
  select(!c("yearSeason"))
```

Computing the salaries of players by season, storing values as a vector

```r
# grouping salary data by slugPlayerSeason
dataBREFPlayersSalaries_grp<- dataBREFPlayersSalaries %>%
  select(c("slugPlayerSeason", "amountSalary")) %>%
  group_by(slugPlayerSeason) %>%
  summarise(totSalary = sum(amountSalary)) # this works!
```

Joining Salary data vector to comprehensive_stats

```r
player_stats<- merge(x=player_stats,
                     y=dataBREFPlayersSalaries_grp,
                     by="slugPlayerSeason",
                     all=F) # Success! 9684
```

Dropping all instances of countTeamsPlayerSeasonTotals

```r
player_stats<- player_stats %>%
  select(!tidyselect::starts_with("countTeamsPlayerSeason"))# 3 columns dropped
```

Dropping extra info columns- Pt1

```r
player_stats<- player_stats %>%
  select(!c("minutesPerGame", "minutesPerMinute","isSeasonCurrent")) # 75 variables currently
```

getting the counts of NAs in each column

```r
count_NAs<- apply(X=player_stats,
                  MARGIN=2,
                  FUN= function(x) {return(sum(is.na(x)))})
count_NAs # 56 NAs in salary column, remove those rows
```

drop 56 columns with NA salary values in them

```
player_stats<- player_stats %>%
  drop_na(totSalary) # 9628 tuples left
```

Counting the number of duplicate primary keys (if any) in comprehensive_stats

```
nums_pl<- player_stats %>%
  count(slugPlayerSeason) %>%
  arrange(desc(n))
View(nums_pl) # no duplicates!
```

importing the csv of salary cap data

Note that this csv was obtained directly from the BasketballReference Website.

```
salary_cap_data<- read.csv(file="sportsref_download.csv",
                           header = T)
salary_cap_data<- salary_cap_data %>%
  dplyr::select(-salary_cap_in2021valuation)
```

formatting the dataframe to remove the $ sign and commas from columns, and change datatype to numeric

```
salary_cap_data$salary_cap<- stringr::str_replace_all(pattern="[$,]",
                                 replacement="",
                                 string =salary_cap_data$salary_cap)
```

adding 1 to salary_cap_data's `yearSeason` for easy merging

```
salary_cap_data$yearSeason<- as.numeric(salary_cap_data$yearSeason) + 1
```

Formatting the `yearSeason` column to ensure easy merging

```
salary_cap_data<- salary_cap_data %>% tidyr::separate(col=yearSeason,
                                                      into=c("yearSeason",
                                                             NA),
                                                      sep='-',
                                                      remove=F)
```

merging `salary_cap_data` to comprehensive_stats

```
player_stats<- merge(x=player_stats,
                     y=salary_cap_data,
                     by="yearSeason")
```

mutating a new column as the ratio between salary and `salary_cap`, called `normalised_salary`

```r
player_stats<- player_stats %>%
  dplyr::mutate(normalised_salary = totSalary / as.numeric(salary_cap))
# WORKS. do not change anything else.
```

saving comprehensive stats in an RData file

```r
# saving as RDS
saveRDS(object=player_stats, file="player_stats.rds")
```