

## Scenario

Congratulations! You were just hired at Nike in sports marketing specific to the National Basketball Association (NBA). Your superiors have always relied on qualitative focus groups to learn about NBA fan engagement. Since you are a freshly minted analytics professional you have been asked to apply new methods to explore a trove of fan tweets. Your bosses are interested in learning more about the commentary for popular teams, athletes, or the league as a whole. Additionally, examining the sentiment and polarity for your analysis will benefit the marketing team’s efforts helping them understand where to allocate spend. At the conclusion of your analysis you need to present your findings in a slide deck to the marketing team, keep in mind they are qualitative so your analysis should describe the data, processing and methods employed not just the results.

## Contextual Information:

*If you decide to examine at the athlete level a comparison may be of interest but you are free to do more research and/or use different athletes.*

* Top NIKE athletes: Lebron James, Giannis Antetokounmpo Kawhi Leonard
* Top Adidas athletes: Damian Lillard, James Harden, Steph Curry

*If you decide to examine the teams individually or compare a subsection (such as a conference, division or popular teams) the following links help provide context on teams.*

* <https://www.nba.com/teams>
* <https://www.hoopsrumors.com/2020/02/forbes-releases-2020-nba-franchise-valuations.html>

## Technical Considerations:

* Provide some insights such as frequent words, emerging topics, share of voice, sentiment/polarity by athlete, team, or across the NBA
* You may use a subset of the data provided based on your analysis and/or compute constraints. If so, you must show the manner the data was reduced in a script and describe why in your presentation.
  + You may look across the entire year, a single month, aggregate to quarters, subset in a manner you decide.
  + Reducing data and not exploring temporal aspects of the data may decrease the robustness of your analysis i.e. exploring month to month change is more robust than all data as a single corpus

## Non-Technical:

* Describe the preprocessing steps and why are they are applied to the documents
* Describe the various techniques used to create the visuals and findings in a powerpoint to a non-technical user

## Project Deliverables include

1. R scripts for data processing & exploration “lastName\_TM\_NBA\_case.R”
   1. Your script(s) must account for all aspects of the material in your presentation to ensure the presentation is data driven (no cheating with Excel or other tools!)
2. Powerpoint of any visualizations, findings and descriptions of non-technical results as if presented to the thinktank’s leader. “lastName\_TM\_NBA\_case\_.pptx”
   1. The PowerPoint must be accompanied by a voice over embedded in the file, or screenshare video uploaded

## Example Data

|  |  |  |  |
| --- | --- | --- | --- |
| doc\_id | Text | Created | team |
| 1179143769574297600 | "Ole Miss: 15 Braves: 0 Falcons: 0 Hawks: 0 I enjoy suffering with my Atlanta friends" | 2019-10-01 21:19:16 | Atlanta Hawks |

## Criteria for Success

The case material will be evaluated according to the following criteria. Each is worth 5pts for a total of 20pts.

## **Organization of content**– Logical ordering of ideas, modeling artifacts, applicable visualizations in slides

## **Organization of code**- R Code is well organized, concise, and free from error

## **Text mining process** – Recognize the type of data mining problem, adherence to established main data and text mining steps.

## **Completeness** – Understood impact, and mined the data for relevant insights/recommendations

A\_Oct2019.csv')

'B\_Nov2019.csv')

'C\_Dec2019.csv')

'D\_Jan2020.csv')

'E\_Feb2020.csv')

'F\_Mar2020.csv')

'G\_Apr2020.csv')

H\_May2020.csv')

'I\_June2020.csv')

'J\_July2020.csv')

'K\_Aug2020.csv')

'L\_Sep2020.csv')

'M\_Oct2020.csv')

# Title: Case1-NBA Fan Engagement

# NAME: Qi Yang

# Date: Jan 20 2021

# Options&Functions

options(stringsAsFactors = FALSE)

Sys.setlocale('LC\_ALL','C')

# set working directory

setwd("~/Downloads/R/hult\_NLP\_student/cases/NBA Fan Engagement/data")

# Load the following libraries ggplot2, ggthemes stringi, and tm

library(ggplot2)

library(ggthemes)

library(stringi)

library(tm)

# load the data

text <- read.csv('A\_Oct2019.csv')

text <- read.csv('B\_Nov2019.csv')

text <- read.csv('C\_Dec2019.csv')

text <- read.csv('D\_Jan2020.csv')

text <- read.csv('E\_Feb2020.csv')

text <- read.csv('F\_Mar2020.csv')

text <- read.csv('G\_Apr2020.csv')

text <- read.csv('H\_May2020.csv')

text <- read.csv('I\_June2020.csv')

text <- read.csv('J\_July2020.csv')

text <- read.csv('K\_Aug2020.csv')

text <- read.csv('L\_Sep2020.csv')

text <- read.csv('M\_Oct2020.csv')

# Logical T/F vector that a string appears at least ONCE

team <- grepl("team", text$text, ignore.case=TRUE)

best <- grepl("best", text$text, ignore.case=TRUE)

Hawks <- grepl("Hawks", text$text, ignore.case=TRUE)

Celtics <- grepl("Celtics", text$text, ignore.case=TRUE)

Nets <- grepl("Nets", text$text, ignore.case=TRUE)

Knicks <- grepl("Knicks", text$text, ignore.case=TRUE)

Philadelphia <- grepl("Philadelphia", text$text, ignore.case=TRUE)

Raptors <- grepl("Raptors", text$text, ignore.case=TRUE)

Bulls <- grepl("Bulls", text$text, ignore.case=TRUE)

Cavaliers <- grepl("Cavaliers", text$text, ignore.case=TRUE)

Pistons <- grepl("Pistons", text$text, ignore.case=TRUE)

Pacers <- grepl("Pacers", text$text, ignore.case=TRUE)

Bucks <- grepl("Bucks", text$text, ignore.case=TRUE)

Hornets <- grepl("Hornets", text$text, ignore.case=TRUE)

Heat <- grepl("Heat", text$text, ignore.case=TRUE)

Magic <- grepl("Magic", text$text, ignore.case=TRUE)

Wizards <- grepl("Wizards", text$text, ignore.case=TRUE)

Nuggets <- grepl("Nuggets", text$text, ignore.case=TRUE)

Timberwolves <- grepl("Timberwolves", text$text, ignore.case=TRUE)

Thunder <- grepl("Thunder", text$text, ignore.case=TRUE)

Blazers <- grepl("Blazers", text$text, ignore.case=TRUE)

Jazz <- grepl("Jazz", text$text, ignore.case=TRUE)

Warriors <- grepl("Warriors", text$text, ignore.case=TRUE)

Clippers <- grepl("Clippers", text$text, ignore.case=TRUE)

Lakers <- grepl("Lakers", text$text, ignore.case=TRUE)

Suns <- grepl("Suns", text$text, ignore.case=TRUE)

Kings <- grepl("Kings", text$text, ignore.case=TRUE)

Mavericks <- grepl("Mavericks", text$text, ignore.case=TRUE)

Rockets <- grepl("Rockets", text$text, ignore.case=TRUE)

Grizzlies <- grepl("Grizzlies", text$text, ignore.case=TRUE)

Pelicans <- grepl("Pelicans", text$text, ignore.case=TRUE)

Spurs <- grepl("Spurs", text$text, ignore.case=TRUE)

# Review Logical Output

head(Hawks,10)

head(Celtics,10)

head(Nets,10)

head(Knicks,10)

head(Philadelphia,10)

head(Raptors,10)

head(Bulls,10)

head(Cavaliers,10)

head(Pistons,10)

head(Pacers,10)

head(Bucks,10)

head(Hornets,10)

head(Heat,10)

head(Magic,10)

head(Wizards,10)

head(Nuggets,10)

head(Timberwolves,10)

head(Thunder,10)

head(Blazers,10)

head(Jazz,10)

head(Warriors,10)

head(Clippers,10)

head(Lakers,10)

head(Suns,10)

head(Kings,10)

head(Mavericks,10)

head(Rockets,10)

head(Grizzlies,10)

head(Pelicans,10)

head(Spurs,10)

#Find the row positions of a specific word appearing at least ONCE

grep("the best show", text$text, ignore.case=TRUE)

grep("the best athlete", text$text, ignore.case=TRUE)

grep("the best team", text$text, ignore.case=TRUE)

# Grep for indexing

text[grep('the best show', text$text),2]

text[grep("the best athlete", text$text),2]

text[grep("the best team", text$text),2]

# Logical T/F for one word OR another appears at least ONCE

keywordsOR <-"show|team|play"

showTeamPlay <- grepl(keywordsOR, text$text,ignore.case=TRUE)

head(text$text[showTeamPlay])

# Logical Search AND operator, regular expression

keywordsAND <- "(?=.\*team)(?=.\*great)"

greatTeam <- grepl(keywordsAND, text$text,perl=TRUE)

head(text$text[greatTeam])

# Calculate the % of times among all tweets

sum(team) / nrow(text)

sum(best) / nrow(text)

sum(showTeamPlay) / nrow(text)

# Functions

tryTolower <- function(x){

y = NA

try\_error = tryCatch(tolower(x), error = function(e) e)

if (!inherits(try\_error, 'error'))

y = tolower(x)

return(y)

}

cleanCorpus<-function(corpus, customStopwords){

corpus <- tm\_map(corpus, content\_transformer(qdapRegex::rm\_url))

corpus <- tm\_map(corpus, content\_transformer(replace\_contraction))

corpus <- tm\_map(corpus, removeNumbers)

corpus <- tm\_map(corpus, removePunctuation)

corpus <- tm\_map(corpus, stripWhitespace)

corpus <- tm\_map(corpus, content\_transformer(tryTolower))

corpus <- tm\_map(corpus, removeWords, customStopwords)

return(corpus)

}

# Create custom stop words

stops <- c(stopwords('SMART'), 'team','best', 'great', 'play',

'show', 'like')

# Read in multiple files as individuals

txtFiles <- list.files(pattern = 'A\_Oct2019|B\_Nov2019|C\_Dec2019|D\_Jan2020|E\_Feb2020|F\_Mar2020|

G\_Apr2020|H\_May2020|I\_June2020|J\_July2020|K\_Aug2020|L\_Sep2020|M\_Oct2020')

for (i in 1:length(txtFiles)){

assign(txtFiles[i], read.csv(txtFiles[i]))

cat(paste('read completed:',txtFiles[i],'\n'))

}

# Vector Corpus

A\_Oct2019 <- VCorpus(VectorSource(A\_Oct2019.csv$text))

B\_Nov2019 <- VCorpus(VectorSource(B\_Nov2019.csv$text))

C\_Dec2019 <- VCorpus(VectorSource(C\_Dec2019.csv$text))

D\_Jan2020 <- VCorpus(VectorSource(D\_Jan2020.csv$text))

E\_Feb2020 <- VCorpus(VectorSource(E\_Feb2020.csv$text))

F\_Mar2020 <- VCorpus(VectorSource(F\_Mar2020.csv$text))

G\_Apr2020 <- VCorpus(VectorSource(G\_Apr2020.csv$text))

H\_May2020 <- VCorpus(VectorSource(H\_May2020.csv$text))

I\_June2020 <- VCorpus(VectorSource(I\_June2020.csv$text))

J\_July2020 <- VCorpus(VectorSource(J\_July2020.csv$text))

K\_Aug2020 <- VCorpus(VectorSource(K\_Aug2020.csv$text))

L\_Sep2020 <- VCorpus(VectorSource(L\_Sep2020.csv$text))

M\_Oct2020 <- VCorpus(VectorSource(M\_Oct2020.csv$text))

# Clean up the data

A\_Oct2019 <- cleanCorpus(VectorSource(A\_Oct2019,stops)

B\_Nov2019 <- cleanCorpus(VectorSource(B\_Nov2019, stops)

C\_Dec2019 <- cleanCorpus(VectorSource(C\_Dec2019, stops)

D\_Jan2020 <- cleanCorpus(VectorSource(D\_Jan2020, stops)

E\_Feb2020 <- cleanCorpus(VectorSource(E\_Feb2020, stops)

F\_Mar2020 <- cleanCorpus(VectorSource(F\_Mar2020, stops)

G\_Apr2020 <- cleanCorpus(VectorSource(G\_Apr2020, stops)

H\_May2020 <- cleanCorpus(VectorSource(H\_May2020, stops)

I\_June2020 <-cleanCorpus(VectorSource(I\_June2020, stops)

J\_July2020 <- cleanCorpus(VectorSource(J\_July2020, stops)

K\_Aug2020 <- cleanCorpus(VectorSource(K\_Aug2020, stops)

L\_Sep2020 <-cleanCorpus(VectorSource(L\_Sep2020, stops)

M\_Oct2020 <- cleanCorpus(VectorSource(M\_Oct2020, stops)

#Cleaned text

A\_Oct2019 <- unlist(VectorSource(A\_Oct2019,content))

B\_Nov2019 <- unlist(VectorSource(B\_Nov2019, content))

C\_Dec2019 <- unlist(VectorSource(C\_Dec2019, content))

D\_Jan2020 <- unlist(VectorSource(D\_Jan2020, content))

E\_Feb2020 <- unlist(VectorSource(E\_Feb2020, content))

F\_Mar2020 <- unlist(VectorSource(F\_Mar2020, content))

G\_Apr2020 <- unlist(VectorSource(G\_Apr2020, content))

H\_May2020 <- unlist(VectorSource(H\_May2020, content))

I\_June2020 <-unlist(VectorSource(I\_June2020, content))

J\_July2020 <- unlist(VectorSource(J\_July2020, content))

K\_Aug2020 <- unlist(VectorSource(K\_Aug2020, content))

M\_Oct2020 <- unlist(VectorSource(M\_Oct2020, content))

# FYI

length(A\_Oct2019)

length(B\_Nov2019)

length(C\_Dec2019)

length(D\_Jan2020)

length(E\_Feb2020)

length(F\_Mar2020)

length(G\_Apr2020)

length(H\_May2020)

length(I\_June2020)

length(J\_July2020)

length(K\_Aug2020)

length(L\_Sep2020)

length(M\_Oct2020)

#collapse each into a single "subject"

A\_Oct2019 <- paste(A\_Oct2019, collapse = ' ')

B\_Nov2019 <- paste(B\_Nov2019, collapse = ' ')

C\_Dec2019 <- paste(C\_Dec2019, collapse = ' ')

D\_Jan2020 <- paste(D\_Jan2020, collapse = ' ')

E\_Feb2020 <- paste(E\_Feb2020, collapse = ' ')

F\_Mar2020 <- paste(F\_Mar2020, collapse = ' ')

G\_Apr2020 <- paste(G\_Apr2020, collapse = ' ')

H\_May2020 <- paste(H\_May2020, collapse = ' ')

I\_June2020 <- paste(I\_June2020, collapse = ' ')

J\_July2020 <- paste(J\_July2020, collapse = ' ')

K\_Aug2020 <- paste(K\_Aug2020, collapse = ' ')

L\_Sep2020 <- paste(L\_Sep2020, collapse = ' ')

M\_Oct2020 <- paste(M\_Oct2020, collapse = ' ')

# Combine the subject documents into a corpus of \*2\* documents

allTeams <- c(A\_Oct2019,B\_Nov2019,C\_Dec2019,D\_Jan2020,E\_Feb2020,F\_Mar2020,

G\_Apr2020,H\_May2020,I\_June2020,J\_July2020,K\_Aug2020,L\_Sep2020,M\_Oct2020)

allTeams <- VCorpus((VectorSource(allTeams)))

# Make TDM with a different control parameter

ctrl <- list(weighting = weightTfIdf)

teamTDM <- TermDocumentMatrix(allTeams, control = ctrl)

teamTDMm <- as.matrix(teamTDM)

colnames(teamTDMm) <- c('A\_Oct2019','B\_Nov2019','C\_Dec2019','D\_Jan2020','E\_Feb2020','F\_Mar2020','G\_Apr2020','H\_May2020','I\_June2020','J\_July2020','K\_Aug2020','L\_Sep2020','M\_Oct2020')

# Examine

head(teamTDMm)

# Make comparison cloud

comparison.cloud(teamTDMm,

max.words=75,

random.order=FALSE,

title.size=0.5,

colors=brewer.pal(ncol(teamTDMm),"Dark2"),

scale=c(3,0.1))

text$text <- factor(text$text,

levels=unique(as.character(text$text)))

ggplot(team, aes(x=word, y=frequency)) +

geom\_bar(stat="identity", fill='darkred') +

coord\_flip()+ theme\_gdocs() +

geom\_text(aes(label=frequency), colour="white",hjust=1.25, size=3.0)