C1X Data Challenge

Manikandan .R

November 23, 2016

## Data

500k User section collection- consists of ~20M web queries collected from ~650k users over three months.

setwd("C:\\intro\_to\_datascience\\competitions\\c1x")  
df<- read.csv("user-ct-test-collection-02.txt", header= TRUE, sep='\t')  
str(df)

## 'data.frame': 3537372 obs. of 5 variables:  
## $ AnonID : int 479 479 479 479 479 479 479 479 479 479 ...  
## $ Query : Factor w/ 1219923 levels "'","''","''''''''''",..: 319265 46899 319284 941157 941157 159737 159737 159737 160378 160379 ...  
## $ QueryTime: Factor w/ 2247762 levels "2006-03-01 00:01:16",..: 13668 40439 81876 82022 82022 82996 82996 82996 83100 83183 ...  
## $ ItemRank : int NA NA 1 1 2 4 1 5 9 3 ...  
## $ ClickURL : Factor w/ 384917 levels "","ftp://dftp.maritimemagazine.com/mn200502o2.pdf",..: 1 1 171859 46123 210428 150789 253249 367070 146623 331564 ...

## Data Pre processing

We split the data into 2 sets-

1. One set containing all queries that was NOT followed by the user clicking on a result item.\*\*df1\*\*  
 2. Other set containing click through on an item in the result list returned from a query.\*\*df2\*\*

df1<- subset(df,ItemRank!= 'NA')  
df[df=='NA']<- NA  
df2<- subset(df,is.na(df$ItemRank))  
str(df1)

## 'data.frame': 1902838 obs. of 5 variables:  
## $ AnonID : int 479 479 479 479 479 479 479 479 479 479 ...  
## $ Query : Factor w/ 1219923 levels "'","''","''''''''''",..: 319284 941157 941157 159737 159737 159737 160378 160379 129184 129192 ...  
## $ QueryTime: Factor w/ 2247762 levels "2006-03-01 00:01:16",..: 81876 82022 82022 82996 82996 82996 83100 83183 83261 83299 ...  
## $ ItemRank : int 1 1 2 4 1 5 9 3 1 1 ...  
## $ ClickURL : Factor w/ 384917 levels "","ftp://dftp.maritimemagazine.com/mn200502o2.pdf",..: 171859 46123 210428 150789 253249 367070 146623 331564 113929 63921 ...

str(df2)

## 'data.frame': 1634534 obs. of 5 variables:  
## $ AnonID : int 479 479 479 479 479 479 479 479 479 479 ...  
## $ Query : Factor w/ 1219923 levels "'","''","''''''''''",..: 319265 46899 160248 159610 132650 186080 186084 447309 695020 1034559 ...  
## $ QueryTime: Factor w/ 2247762 levels "2006-03-01 00:01:16",..: 13668 40439 83231 83245 83257 109827 109832 201284 234078 360616 ...  
## $ ItemRank : int NA NA NA NA NA NA NA NA NA NA ...  
## $ ClickURL : Factor w/ 384917 levels "","ftp://dftp.maritimemagazine.com/mn200502o2.pdf",..: 1 1 1 1 1 1 1 1 1 1 ...

Removing duplicate entries

df1<- unique(df1)  
df2<- unique(df2)  
nrow(df1)

## [1] 1754149

nrow(df2)

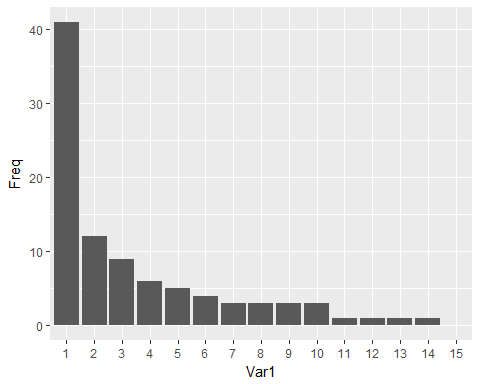
## [1] 1623197

## Exploring ItemRank variable

ranktab<-round(prop.table(table(df1$ItemRank))\*100)  
ranktabdf<- as.data.frame(ranktab)  
str(ranktabdf)

## 'data.frame': 497 obs. of 2 variables:  
## $ Var1: Factor w/ 497 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Freq: num 0 41 12 9 6 5 4 3 3 3 ...

ranktabdf1<- ranktabdf[2:16,]  
library(ggplot2)  
ggplot(ranktabdf1,aes(x= Var1,y= Freq))+geom\_bar(stat = "identity")



It is infered that 89% of clicked users were able to get their searches within 10 ranked items

## Exploring Queries searched by user

library(data.table)  
library(plyr)  
library(dplyr)

## -------------------------------------------------------------------------

## data.table + dplyr code now lives in dtplyr.  
## Please library(dtplyr)!

## -------------------------------------------------------------------------

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:data.table':  
##   
## between, last

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

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

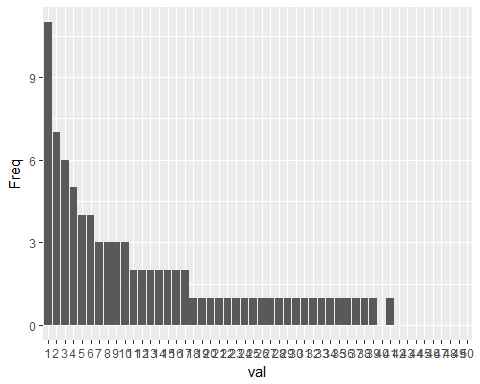
dtbl<- tbl\_df(df)  
dtbl3<- dtbl %>% group\_by(AnonID) %>% summarise(n=n\_distinct(Query))  
head(dtbl3,10)

## # A tibble: 10 × 2  
## AnonID n  
## <int> <int>  
## 1 479 77  
## 2 507 26  
## 3 946 2  
## 4 1020 66  
## 5 1021 51  
## 6 1521 20  
## 7 2015 11  
## 8 2708 305  
## 9 2729 57  
## 10 2914 12

val<- dtbl3[,2]  
querytab<- round(prop.table(table(val))\*100)  
querytab<- as.data.frame(querytab)  
querytab1<- querytab[1:50,]  
querytbl<- tbl\_df(querytab1)  
mutate(querytbl,cumulative=cumsum(querytbl$Freq))

## # A tibble: 50 × 3  
## val Freq cumulative  
## <fctr> <dbl> <dbl>  
## 1 1 11 11  
## 2 2 7 18  
## 3 3 6 24  
## 4 4 5 29  
## 5 5 4 33  
## 6 6 4 37  
## 7 7 3 40  
## 8 8 3 43  
## 9 9 3 46  
## 10 10 3 49  
## # ... with 40 more rows

ggplot(querytab1,aes(x=val,y=Freq))+geom\_bar(stat = "identity")



Nearly 50% of the users queried the search engine with 1-10 query topics

## Exploring at Overall search engine level

We separate the users into 2 segments-

users who got relevant search- those who had atleast clicked on one query search result. users who didn't get relevant search- those who hasn't clicked on any query search result.

Reshping the data to get click details at user level Deriving metrics at an overall search engine level- Overall Click Rate and Overall Non-Click Rate.

library("magrittr")  
dbtl4<- dtbl %>% select(AnonID,ItemRank)  
dbtl4$rank<-dbtl4$ItemRank  
dbtl4[dbtl4=='NA']<- NA  
  
dbtl4$rank<- apply(dbtl4[,c("ItemRank")],1, function(i) ifelse((is.na(i)),'NA','CL'))  
head(dbtl4)

## # A tibble: 6 × 3  
## AnonID ItemRank rank  
## <int> <int> <chr>  
## 1 479 NA NA  
## 2 479 NA NA  
## 3 479 1 CL  
## 4 479 1 CL  
## 5 479 2 CL  
## 6 479 4 CL

dbtl6<- dbtl4 %>% group\_by(AnonID,rank) %>% summarise(n=n())  
head(dbtl6)

## Source: local data frame [6 x 3]  
## Groups: AnonID [4]  
##   
## AnonID rank n  
## <int> <chr> <int>  
## 1 479 CL 88  
## 2 479 NA 35  
## 3 507 CL 63  
## 4 507 NA 52  
## 5 946 NA 2  
## 6 1020 CL 64

library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':  
##   
## extract

dbtl7<- spread(dbtl6,rank,n)  
head(dbtl7)

## Source: local data frame [6 x 3]  
## Groups: AnonID [6]  
##   
## AnonID CL `NA`  
## <int> <int> <int>  
## 1 479 88 35  
## 2 507 63 52  
## 3 946 NA 2  
## 4 1020 64 44  
## 5 1021 35 48  
## 6 1521 NA 27

nrow(dbtl7)

## [1] 61592

dbtl9<- dbtl7[which(!is.na(dbtl7$CL)),]  
nrow(dbtl9)

## [1] 49294

dbtl8<- dbtl7[which(is.na(dbtl7$CL)),]  
nrow(dbtl8)

## [1] 12298

OverallCR<- nrow(dbtl9)/nrow(dbtl7)\*100  
OverallNCR<- nrow(dbtl8)/nrow(dbtl7)\*100  
OverallCR

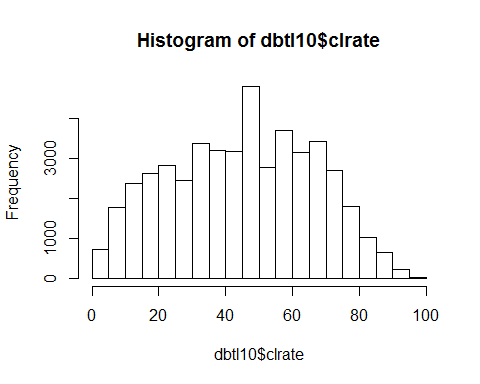
## [1] 80.03312

OverallNCR

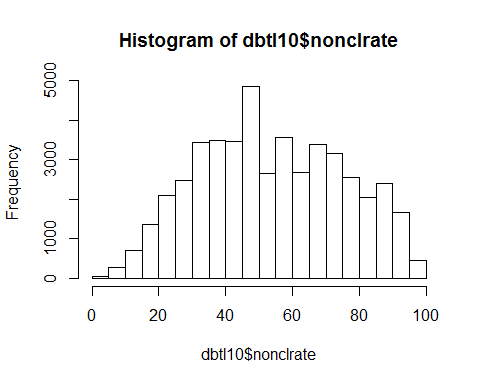
## [1] 19.96688

## Exploring the users of relevant search

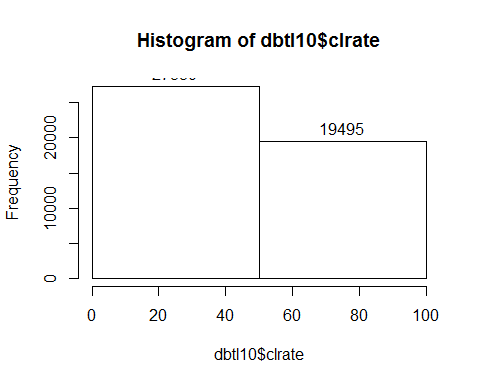
a<- data.frame(name=c("ID","Click","NoClick"))  
colnames(dbtl9)<- a$name  
dbtl10<- dbtl9 %>% mutate(Total = Click+NoClick) %>% mutate(clrate= round((Click/Total)\*100)) %>% mutate(nonclrate= round((NoClick/Total)\*100))  
  
hist(dbtl10$clrate)



hist(dbtl10$nonclrate)



hist(dbtl10$clrate,breaks = 2,labels = TRUE)



mean(dbtl10$clrate,na.rm = TRUE)

## [1] 45.5755

mean(dbtl10$nonclrate,na.rm = TRUE)

## [1] 54.42381

58% of the users have their click rate <= 50.

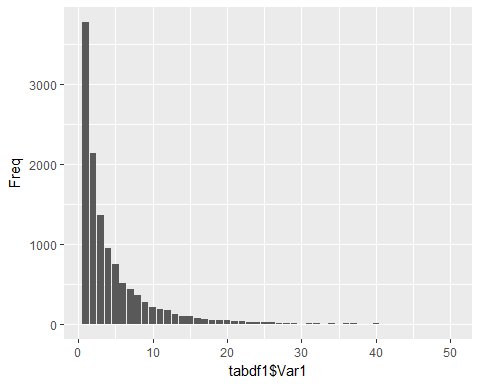
## Exploring the users who had not got relevant search

Here, we consider all those users who haven't clicked on any search results.

dbtl8<- as.data.frame(dbtl8)  
a<- data.frame(name=c("ID","Click","NoClick"))  
colnames(dbtl8)<- a$name   
head(dbtl8)

## ID Click NoClick  
## 1 946 NA 2  
## 2 1521 NA 27  
## 3 4349 NA 10  
## 4 6860 NA 6  
## 5 16312 NA 2  
## 6 27095 NA 36

tabdf<- table(dbtl8$NoClick)  
tabdf<- as.data.frame(tabdf)  
tabdf$Var1<- as.numeric(tabdf$Var1)  
tabdf1<- tabdf[which(tabdf$Var1<=50),]  
ggplot(tabdf1,aes(x=tabdf1$Var1,y=Freq))+geom\_bar(stat = "identity")



30% of users stop their search with one query(less business action could be taken on this segment). 57% of users do not click any search results for their query topics ranging from 2 to 10.(this segment calls for business attention)

## top urls showing up in the results

dtb1<- tbl\_df(df1)  
dtb2<- count(dtb1,ClickURL)  
head(dtb2)

## # A tibble: 6 × 2  
## ClickURL n  
## <fctr> <int>  
## 1 ftp://dftp.maritimemagazine.com/mn200502o2.pdf 1  
## 2 http:\017\017www.hotmail.com\017 1  
## 3 http:// 2  
## 4 http://-vraiment.blogspot.com 1  
## 5 http:// adulted.about.com 1  
## 6 http:// crime.about.com 1

dtb2<- arrange(dtb2,desc(n))  
head(dtb2,10)

## # A tibble: 10 × 2  
## ClickURL n  
## <fctr> <int>  
## 1 http://www.google.com 25794  
## 2 http://www.myspace.com 13304  
## 3 http://www.yahoo.com 12118  
## 4 http://en.wikipedia.org 11522  
## 5 http://www.amazon.com 10345  
## 6 http://www.imdb.com 9227  
## 7 http://www.mapquest.com 8414  
## 8 http://www.ebay.com 6035  
## 9 http://mail.yahoo.com 4002  
## 10 http://www.bankofamerica.com 3832

## Time spent by users in search engine.

The objective is to capture the time spent per day calculated to be one session. if the user had spent only on one query, time spent would be zero. if a user had spend 5 query topics in a day, then the time spent is calculated between the time of first query and last query of the day.

df4<- df  
dbtt3<- tbl\_df(df4)  
str(dbtt3)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 3537372 obs. of 5 variables:  
## $ AnonID : int 479 479 479 479 479 479 479 479 479 479 ...  
## $ Query : Factor w/ 1219923 levels "'","''","''''''''''",..: 319265 46899 319284 941157 941157 159737 159737 159737 160378 160379 ...  
## $ QueryTime: Factor w/ 2247762 levels "2006-03-01 00:01:16",..: 13668 40439 81876 82022 82022 82996 82996 82996 83100 83183 ...  
## $ ItemRank : int NA NA 1 1 2 4 1 5 9 3 ...  
## $ ClickURL : Factor w/ 384917 levels "","ftp://dftp.maritimemagazine.com/mn200502o2.pdf",..: 1 1 171859 46123 210428 150789 253249 367070 146623 331564 ...

dbtt3$Date<- as.Date(dbtt3$QueryTime)  
dbtt3$Time<- format(as.POSIXct(dbtt3$QueryTime),format = "%H:%M:%S")  
  
dbtt3 %>% group\_by(AnonID,Date) %>% summarise(min(Time),max(Time))->dbtt4  
head(dbtt4)

## Source: local data frame [6 x 4]  
## Groups: AnonID [1]  
##   
## AnonID Date `min(Time)` `max(Time)`  
## <int> <date> <chr> <chr>  
## 1 479 2006-03-01 16:01:20 16:01:20  
## 2 479 2006-03-02 14:48:55 14:48:55  
## 3 479 2006-03-03 22:37:46 23:31:48  
## 4 479 2006-03-04 22:11:31 23:05:53  
## 5 479 2006-03-06 21:57:14 22:57:22  
## 6 479 2006-03-08 00:14:55 17:23:07

str(dbtt4)

## Classes 'grouped\_df', 'tbl\_df', 'tbl' and 'data.frame': 719461 obs. of 4 variables:  
## $ AnonID : int 479 479 479 479 479 479 479 479 479 479 ...  
## $ Date : Date, format: "2006-03-01" "2006-03-02" ...  
## $ min(Time): chr "16:01:20" "14:48:55" "22:37:46" "22:11:31" ...  
## $ max(Time): chr "16:01:20" "14:48:55" "23:31:48" "23:05:53" ...  
## - attr(\*, "vars")=List of 1  
## ..$ : symbol AnonID  
## - attr(\*, "drop")= logi TRUE

library("chron")  
  
dbtt4$MINTime<- chron(times=dbtt4$`min(Time)`)  
dbtt4$MAXTime<- chron(times=dbtt4$`max(Time)`)   
dbtt4$timespent<- dbtt4$MAXTime-dbtt4$MINTime  
head(dbtt4)

## Source: local data frame [6 x 7]  
## Groups: AnonID [1]  
##   
## AnonID Date `min(Time)` `max(Time)` MINTime MAXTime  
## <int> <date> <chr> <chr> <S3: times> <S3: times>  
## 1 479 2006-03-01 16:01:20 16:01:20 16:01:20 16:01:20  
## 2 479 2006-03-02 14:48:55 14:48:55 14:48:55 14:48:55  
## 3 479 2006-03-03 22:37:46 23:31:48 22:37:46 23:31:48  
## 4 479 2006-03-04 22:11:31 23:05:53 22:11:31 23:05:53  
## 5 479 2006-03-06 21:57:14 22:57:22 21:57:14 22:57:22  
## 6 479 2006-03-08 00:14:55 17:23:07 00:14:55 17:23:07  
## # ... with 1 more variables: timespent <S3: times>

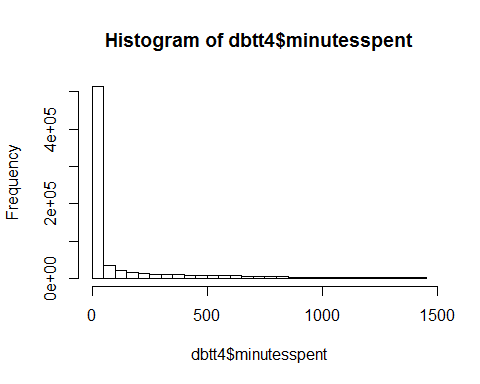
dim(dbtt4)

## [1] 719461 7

dbtt4$minutesspent<- 60\*24\* as.numeric(times(dbtt4$timespent))  
head(dbtt4)

## Source: local data frame [6 x 8]  
## Groups: AnonID [1]  
##   
## AnonID Date `min(Time)` `max(Time)` MINTime MAXTime  
## <int> <date> <chr> <chr> <S3: times> <S3: times>  
## 1 479 2006-03-01 16:01:20 16:01:20 16:01:20 16:01:20  
## 2 479 2006-03-02 14:48:55 14:48:55 14:48:55 14:48:55  
## 3 479 2006-03-03 22:37:46 23:31:48 22:37:46 23:31:48  
## 4 479 2006-03-04 22:11:31 23:05:53 22:11:31 23:05:53  
## 5 479 2006-03-06 21:57:14 22:57:22 21:57:14 22:57:22  
## 6 479 2006-03-08 00:14:55 17:23:07 00:14:55 17:23:07  
## # ... with 2 more variables: timespent <S3: times>, minutesspent <dbl>

dbtt4$minutesspent<- round(dbtt4$minutesspent,0)  
hist(dbtt4$minutesspent)

 The distribution appears to be right skewed.

## Insights to the Product manager of Search

*The Search display page should show 14 results per page. This covers 93% click probability of relevant users and hence, higher click rates* *Eventhough overall click rate figures are better, click rate per user needs attention*

## query words that not leads to a click

## Code not executed due to memory limit exceeded error(Error: could not allocate memory (0 Mb) in C function 'R\_AllocStringBuffer')

rec<-df2$Query head(rec)

library(tm) library("wordcloud") library("qdap") rec\_source<- VectorSource(rec) rec\_corpus<- VCorpus(rec\_source) rec\_corpus[[1]][1]

clean\_corpus<- function(corpus){ corpus<- tm\_map(corpus, content\_transformer(replace\_abbreviation)) corpus<- tm\_map(corpus, removePunctuation) corpus<- tm\_map(corpus, removeNumbers) corpus<- tm\_map(corpus, removeWords, c(stopwords("en"),"hpe")) corpus<- tm\_map(corpus, content\_transformer(tolower)) return(corpus) }

clean\_rec<- clean\_corpus(rec\_corpus) clean\_rec[[1]][1]

rec\_tdm<- TermDocumentMatrix(clean\_rec) rec\_tdm

rec\_m<- as.matrix(rec\_tdm) term\_freqs<- rowSums(rec\_m) term\_freqs

term\_freqs<- sort(term\_freqs,decreasing = TRUE)

wordfreq<- data.frame(term= names(term\_freqs), num= term\_freqs) wordcloud(wordfreqnum, max.words = 100)

## query words that leads to a click

## Code not executed due to memory limit exceeded error(Error: could not allocate memory (0 Mb) in C function 'R\_AllocStringBuffer')

rec1<-df1$query head(rec1)

library(tm) library("wordcloud") library("qdap") rec\_source1<- VectorSource(rec1) rec\_corpus1<- VCorpus(rec\_source1) rec\_corpus1[[1]][1]

clean\_corpus1<- function(corpus){ corpus<- tm\_map(corpus, content\_transformer(replace\_abbreviation)) corpus<- tm\_map(corpus, removePunctuation) corpus<- tm\_map(corpus, removeNumbers) corpus<- tm\_map(corpus, removeWords, c(stopwords("en"),"hpe")) corpus<- tm\_map(corpus, content\_transformer(tolower)) return(corpus) }

clean\_rec1<- clean\_corpus1(rec\_corpus1) clean\_rec1[[1]][1]

rec\_tdm1<- TermDocumentMatrix(clean\_rec1) rec\_tdm1

rec\_m1<- as.matrix(rec\_tdm1) term\_freqs1<- rowSums(rec\_m1) term\_freqs1

term\_freqs1<- sort(term\_freqs1,decreasing = TRUE)

wordfreq1<- data.frame(term= names(term\_freqs1), num= term\_freqs1) wordcloud(wordfreq1num, max.words = 100) ```