SAMET TURGUT

35909310684

Veri Madenciliği Final Projesi

PROJENİN ÖZETİ

15 Aralık 2021 ve 17 Ocak 2022 tarihleri arasında Türkiye bölgesinde “Steam” hakkında atılan 7500 adet tweet toplanmış olup, bu tweet’ler veri temizleme aşamalarından geçirildikten sonra metin madenciliği ve duygu analizi işlemlerine tabi tutulmuştur. Metin madenciliği sayesinde belirtilen tarihler aralığında bulunan tweet’ler içerisinde “Steam” kelimesi ile birlikte en fazla kullanılan diğer kelimeler ortaya çıkmıştır. Duygu analizi aşamasında ise belirtilen bu tarihler aralığında atılan tweet’lerde bulunan pozitif ve negatif kelimeler tespit edilmiştir. Cümlelerde bulunan pozitif ve negatif kelimelerden hangisi fazlalıktaysa tweet’ler bu işleme göre pozitif veya negatif cümle olarak ayrılmıştır. Cümlelerde bulunan negatif ve pozitif kelimeler eşitse veya bunlardan her ikisi de cümlelerde bulunmuyorsa ise bu cümleler de nötr olarak gruplandırılmıştır. Bu sayede bu tarihler aralığında atılan tweet’lerden yola çıkarak 7500 adet tweet içerinden kaç tanesinin “Steam” hakkında pozitif, negatif veya nötr fikre sahip olduğuna ulaşılmıştır. Bu işlemlerin tamamı R programlama diliyle yapılmıştır.

Metin Madenciliği ve Duygu Analizi Aşamasında Kullanılan Paketler

Aşağıda belirtilen paketlerin tamamı bu projede aktif durumda değildir. Bazı paketler atıl durumda kalmıştır ancak diğer popüler opsiyonları göstermek amacıyla projeye eklenmiştir.

library(ROAuth): Twitter Developer hesabının yetkilerine erişebilmek için kullanılır.

library(twitteR): Twitter üzerinden veri çekmek için gerekli olan pakettir.

library(tm): Metin madenciliği sürecinde kullanılan pakettir.

library(plyr): Veri setlerini parçalayıp üzerinde fonksiyon kullanıp tekrar birleştirmeye yarayan pakettir.

library(dplyr): Veri manipülasyonu için gerekli olan pakettir.

library(purrr): R’ın fonksiyonel programlama alet takımını geliştiren pakettir.

library(RCurl): Fonksiyonlara HTTP istekleri sağlayan pakettir.

library(SnowballC): Porter’ın farklı dillerde kelime sıkma algoritmasını kullanan pakettir.

library(openssl): OpenSSL'e bağlanma paketidir.

library(wordcloud): Veri görselleştirme aşamasında kelime bulutu oluşturmak için kullanılan pakettir.

library(ggplot2): Veri görselleştirme aşamasında grafikler oluşturmak için kullanılan pakettir.

library(RColorBrewer): Veri görselleştirme aşamasında renkli görseller oluşturmak için kullanılan pakettir.

library(stringr): Hızlı ve doğru bir şekilde string uygulamaları ve string manipülasyonları sağlayan pakettir.

library(stringi): stringr'nin yaptığı işlemleri yapan, içerisinde bazı farklı fonksiyonlar bulunduran ve stringr'nin eski versiyonu olan bir pakettir.

library(tidytext): Metin madenciliği sürecinde verileri düzenlemeyi sağlayan pakettir.

library(readr): "csv", "tsv","fwf" uzantılarına sahip verileri okumayı sağlayan pakettir.

library(hms): Zaman değişkenleri barındıran verileri şekillendirmek için kullanılan pakettir.

library(lubridate): Saat dilimi, artık gün, yaz saati uygulamaları gibi zaman değişkenleri barındıran verileri şekillendirmek için kullanılan pakettir.

library(igraph): Grafikler yaratmak, şekillendirmek ve analiz etmek için kullanılan pakettir.

library(glue): Yorumlanmış string kalıpları için kullanılan pakettir.

library(networkD3): Ağ grafikleri oluşturmak için kullanılan HTML aygıtı paketidir.

library(rtweet): Twitter'ın REST ve yayın API'ına erişmek için kullanılan pakettir.

library(ggeasy): ggplot2 paketinde yapması zor olan bazı grafik işlemlerini gerçekleştirmek için kullanılan pakettir.

library(plotly): Javascript kullanarak interaktif web grafikleri oluşturmak için kullanılan pakettir.

library(magrittr): Bazı fonksiyonların daha rahat okunabilmesi için operatörler sağlayan bir pakettir.

library(tidyverse): Verileri düzenlemek için kullanılan paketleri kapsayan ana pakettir.

library(janeaustenr): Jane Austen'in yayımlanmış 6 kitabının tüm metinlerine UTF-8 formatında erişmek için kullanılan pakettir.

library(widyr): Verileri matrislere dönüştürüp tekrar düzenli forma getiren pakettir.

library(xlsx): Excel dosyalarını okumak için kullanılan pakettir.

library(readxl): Verileri Excel'den R'a aktarmak için kullanılan ve xlsx paketinden daha kolay çalışan bir pakettir.

METİN MADENCİLİĞİ

R kodlarının çalışması için öncelikle gerekli olan paketlerin bilgisayara yüklenmesi gerekiyor. Bunun için install.packages() komutunu kullanıyoruz.

```
install.packages("ROAuth")
install.packages("twitteR")
install.packages("tm")
install.packages("plyr")
install.packages("dplyr")
install.packages("purrr")
install.packages("RCurl")
install.packages("SnowballC")
install.packages("openssl")
install.packages("wordcloud")
install.packages("ggplot2")
install.packages("RColorBrewed")
install.packages("stringr")
install.packages("textclean")
install.packages("tidytext")
install.packages("readr")
install.packages("xlsx")

install.packages("hms")
install.packages("lubridate")
install.packages("igraph")
install.packages("glue")
install.packages("networkD3")
install.packages("rtweet")
install.packages("ggeasy")
install.packages("plotly")
install.packages("hms")
install.packages("lubridate")
install.packages("magrittr")
install.packages("tidyverse")
install.packages("janeaustenr")
install.packages("widyr")
```

Gerekli olan paketler bilgisayara yüklendikten sonra ise bu paketlerin kütüphaneden çağrılması gerekiyor. Bunun için ise library() komutunu kullanıyoruz.

```
library(ROAuth)
library(twitteR)
library(tm)
library(plyr)
library(dplyr)
library(purrr)
library(RCurl)
library(SnowballC)
library(openssl)
library(wordcloud)
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(tidytext)
library(readr)
library(stringi)

library(hms)
library(lubridate)
library(igraph)
library(glue)
library(networkD3)
library(rtweet)
library(ggeasy)
library(plotly)
library(hms)
library(lubridate)
library(magrittr)
library(tidyverse)
library(janeaustenr)
library(widyr)

library(xlsx)
library(readxl)
```

Paketleri kütüphaneden çağırdıktan sonra ise bu projenin çalışması için gerekli olan ve Twitter tarafından verilen Twitter Developer hesabına ait olan bilgileri çağırıyoruz.

```
options(httr_oauth_cache=T)

APIkey <- "[redacted]"
APIsecret <- "[redacted]"
access_token <- "[redacted]"
access_secret <- "[redacted]"

setup_twitter_oauth(APIkey, APIsecret, access_token, access_secret)
```

Tweet'leri Çekme

Bu işlemlerin ardından Twitter'dan tweet çekme işlemine başlayabiliriz. Türkçe dilinde içinde "steam" kelimesinin geçtiği 5000 adet tweet için arama yapıyoruz. Bu işlem sırasında otomatik olarak en fazla 1 hafta geriye gidecek şekilde veri çekiliyor. Anlık olarak çekilebilecek 5000 adet tweet bulunmuyorsa bulunan miktar kadar tweet gelecektir. Çekilen tweetleri data frame'e dönüştürüyoruz ve ardından bu data frame'i ise tweet'leri temizleme işleminde kullanacağımız yeni bir data frame'e dönüştürüyoruz. Gelen tweetleri ise UTF-8 formatına dönüştürüyoruz.

```
82
83  steam <- searchTwitteR("steam", n=5000,  lang = "tr")
84
85
86  steam.df <- twListToDF(steam)
87
88  length(steam)
89
90  tweet_clean <- steam.df
91
92  tweet_clean$text <- stri_enc_toutf8(tweet_clean$text)
```

O anlığına çekilen tweet miktarı için;

```
> length(steam)
[1] 1248
```

Çekilen tweet'lerin içeriğini görmek için summary(steam.df) komutu ile inceliyoruz. 16 adet sütun ve 1248 adet satırdan oluştuğunu görebiliyoruz. Ayrıca her sütunun başlığı, veri tipleri, minimum, maksimum, ortalama, 1. Çeyrek, 3. Çeyrek, medyan gibi değerleri de görebiliyoruz.

```
> summary(steam.df)
     text           favorited       favoriteCount      replyToSN           created                       truncated
 Length:1248        Mode :logical   Min.    :   0.00   Length:1248        Min.   :2022-01-09 11:31:15   Mode :logical
 Class :character   FALSE:1248      1st Qu.:   0.00   Class :character   1st Qu.:2022-01-11 22:08:47   FALSE:943
 Mode  :character                   Median :   0.00   Mode  :character   Median :2022-01-13 18:23:01   TRUE :305
                                    Mean   :  10.85                      Mean   :2022-01-13 17:11:34
                                    3rd Qu.:   2.00                      3rd Qu.:2022-01-15 10:55:52
                                    Max.   :3874.00                      Max.   :2022-01-17 20:20:45
  replyToSID            id            replyToUID        statusSource        screenName         retweetCount
 Length:1248        Length:1248        Length:1248        Length:1248        Length:1248        Min.    :  0.00
 Class :character   Class :character   Class :character   Class :character   Class :character   1st Qu.:  0.00
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :  0.00
                                                                                               Mean   : 17.18
                                                                                               3rd Qu.:  4.00
                                                                                               Max.   :122.00
 isRetweet       retweeted        longitude          latitude
 Mode :logical   Mode :logical   Length:1248        Length:1248
 FALSE:832       FALSE:1248      Class :character   Class :character
 TRUE :416                       Mode  :character   Mode  :character
```

Bu adımın ardından veri temizleme adımına geçilebilir ancak 1 ay süren bir veri toplama işleminin ardından çekilen tüm tweet'lerin tek bir dosya haline getirilip birleştirilmesinden sonra göstermek tekrarı önlemek açısından daha mantıklı olacaktır. Şimdilik tweet'leri çeker çekmez kaydettiğimizi varsayalım. Tweet'leri nereye kaydetmek istiyorsak dosya konumunu yazdıktan sonra tweet'leri hangi tarihte çektiğimizi belirten bir isimle kaydetmek oldukça faydalı olacaktır. Burada "tweet_clean" yani temizleme işlemi yapılan tweet'leri kaydettim ancak bu adımı bir sonraki aşamaya bıraktığımız için steam data frame'ini kullanarak kaydetmek daha tutarlı olacaktır.

```
154
155 write.xlsx(tweet_clean, "C:/Users/samet/Desktop/twitter_steam_text_mining/17.01.2022tweets.xlsx")
156
```

Tweet'leri Birleştirme

Artık tweet'leri Twitter'dan çekmek yerine bilgisayarımızda depoladığımız, butun-tweetler adını verdiğimiz excel dosyasından açıyoruz. Twitter Developer hesabından yaptığımız bağlantıya da ihtiyaç duymuyoruz. Bu sebeple twListToDF komutu artık burada çalışmayacak. Onun yerine as.data.frame komutunu kullanarak tweet'lerimizi tekrar data frame'e çeviriyoruz. Tweet'ler temizlendikten sonra ayrı bir data frame olacağı için tweet_clean adında bir data frame oluşturuyoruz ve UTF-8 formatına dönüştürüyoruz.

```
78 steam <- read_excel("C:/Users/samet/Desktop/twitter_steam_text_mining/butun-tweetler.xlsx")
79 steam.df <- as.data.frame(steam)
80 tweet_clean <- steam.df
81
82
83
84 tweet_clean$text <- stri_enc_toutf8(tweet_clean$text)
```

Verileri Temizleme

Tweet'leri temizleme işlemine retweet eden kullanıcıların başında bulunan "RT" ifadelerini temizlemekle başlıyoruz. Ardından "http" ile başlayan url linklerini, "#" ve "@" ifadelerini temizliyoruz. Daha sonra ise noktalama işaretlerini ve rakamları da tweetlerden arındırıyoruz ve tüm harfleri küçük harfe çeviriyoruz. Devamında ise ASCII formatına uygun olmayan ve alfabetik olmayan karakterleri siliyoruz.

```
86  tweet_clean$text <- ifelse(str_sub(tweet_clean$text,1,2) == "RT",
87                             substring(tweet_clean$text,3),
88                             tweet_clean$text)
89
90
91  #URL linklerinin temizlenmesi
92  tweet_clean$text <- str_replace_all(tweet_clean$text, "http[^[:space:]]*", "")
93
94
95  #Hashtag "#" ve "@" iÃ·Â·aretlerinin kaldÃ·Â±rÃ·Â±lmasÃ·Â±
96  tweet_clean$text <- str_replace_all(tweet_clean$text, "#\\S+", "")
97  tweet_clean$text <- str_replace_all(tweet_clean$text, "@\\S+", "")
98
99  #Noktalama iÃ·Â·aretlerinin temizlenmesi
100 tweet_clean$text <- str_replace_all(tweet_clean$text, "[[:punct:][:blank:]]+", " ")
101
102 #TÃ·Â¼m harflerin kÃ·Â¼Ã·Â§Ã·Â¼k harfe dÃ·Â¶nÃ·Â¼Ã·Â·tÃ·Â¼rÃ·Â¼lmesi
103 tweet_clean$text  <- str_to_lower(tweet_clean$text, "tr")
104
```

```
105 #RakamlarÃ·Â±n temizlenmesi
106 tweet_clean$text <- removeNumbers(tweet_clean$text)
107
108 #ASCII formatÃ·Â±na uymayan karakterlerin temizlenmesi
109 tweet_clean$text <- str_replace_all(tweet_clean$text, "[<].*[>]", "")
110 tweet_clean$text <- gsub("\uFFFD", "", tweet_clean$text, fixed =  TRUE)
111 tweet_clean$text <- gsub("\n", "", tweet_clean$text, fixed =  TRUE)
112
113 #Alfabetik olmayan karakterlerin temizlenmesi
114 tweet_clean$text <- str_replace_all(tweet_clean$text, "[^[:alnum:]]"," " )
```

Daha sonra bir corpus oluşturarak Turkish Stopwords ve Durak Kelimeler adındaki dosyalarımızı da projeye ekleyerek veri temizleme işleminin son adımını gerçekleştiriyoruz.

```
114 steamCorpus <- Corpus(VectorSource(tweet_clean$text))
115 inspect(steamCorpus[1:10])
```

```
turkish_stopwords <- read_excel("C:/Users/samet/Desktop/twitter_steam_text_mining/Turkish-Stopwords.xlsx")
turkish_stopwords
steamCorpus <- tm_map(steamCorpus, removeWords, turkish_stopwords)

durakkelimeler <- read_excel("C:/Users/samet/Desktop/twitter_steam_text_mining/durakkelimeler.xlsx")
durakkelimeler
steamCorpus <- tm_map(steamCorpus, removeWords, durakkelimeler)
```

```
> turkish_stopwords
# A tibble: 10,349 x 1
   STOPWORD
   <chr>
 1 bir
 2 ve
 3 bu
 4 da
 5 de
 6 için
 7 daha
 8 çok
 9 gibi
10 o
# ... with 10,339 more rows
```

```
> durakkelimeler
# A tibble: 620 x 1
   a
   <chr>
 1 aa
 2 aaa
 3 acaba
 4 ad
 5 ah
 6 aha
 7 ahada
 8 aldi
 9 alem
10 âlem
# ... with 610 more rows
```

Bu temizleme işlemlerinin ardından temiz verileri artık butun-tweetler-temiz olarak ayrı bir excel dosyasında depolayabiliriz.

```
write.xlsx(tweet_clean, "C:/Users/samet/Desktop/twitter_steam_text_mining/butun-tweetler-temiz.xlsx")
```

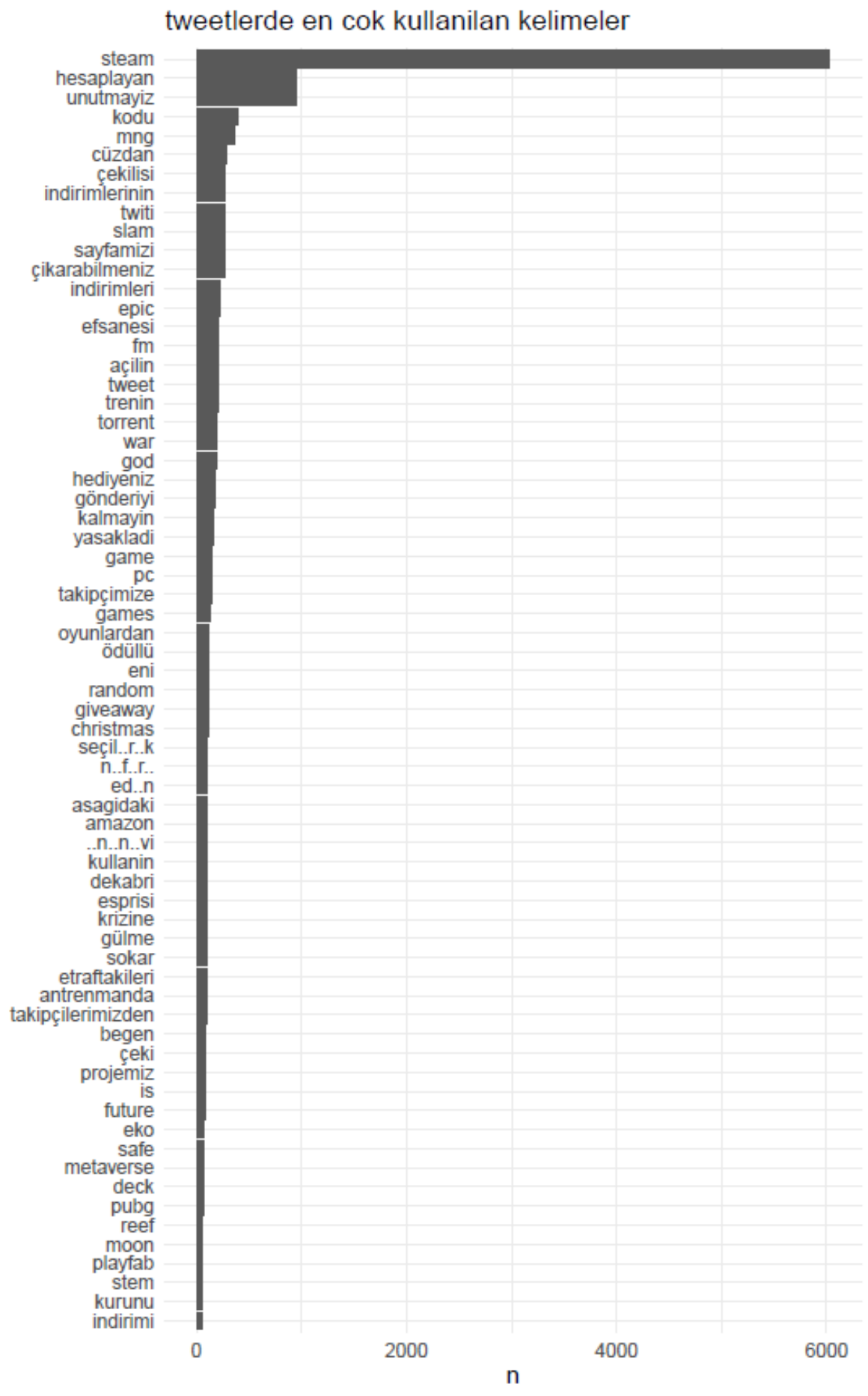
Veri Görselleştirme

Öncelikle görselleştirilecek olan verinin nereden çekileceği işlemini yapıyoruz. Temizlenmiş olan tweet_clean'dan sütun başlığı text olan, yani tweetlerin olduğu sütunu Turkish Stopwords'e uygun bir şekilde oluşturuyoruz.

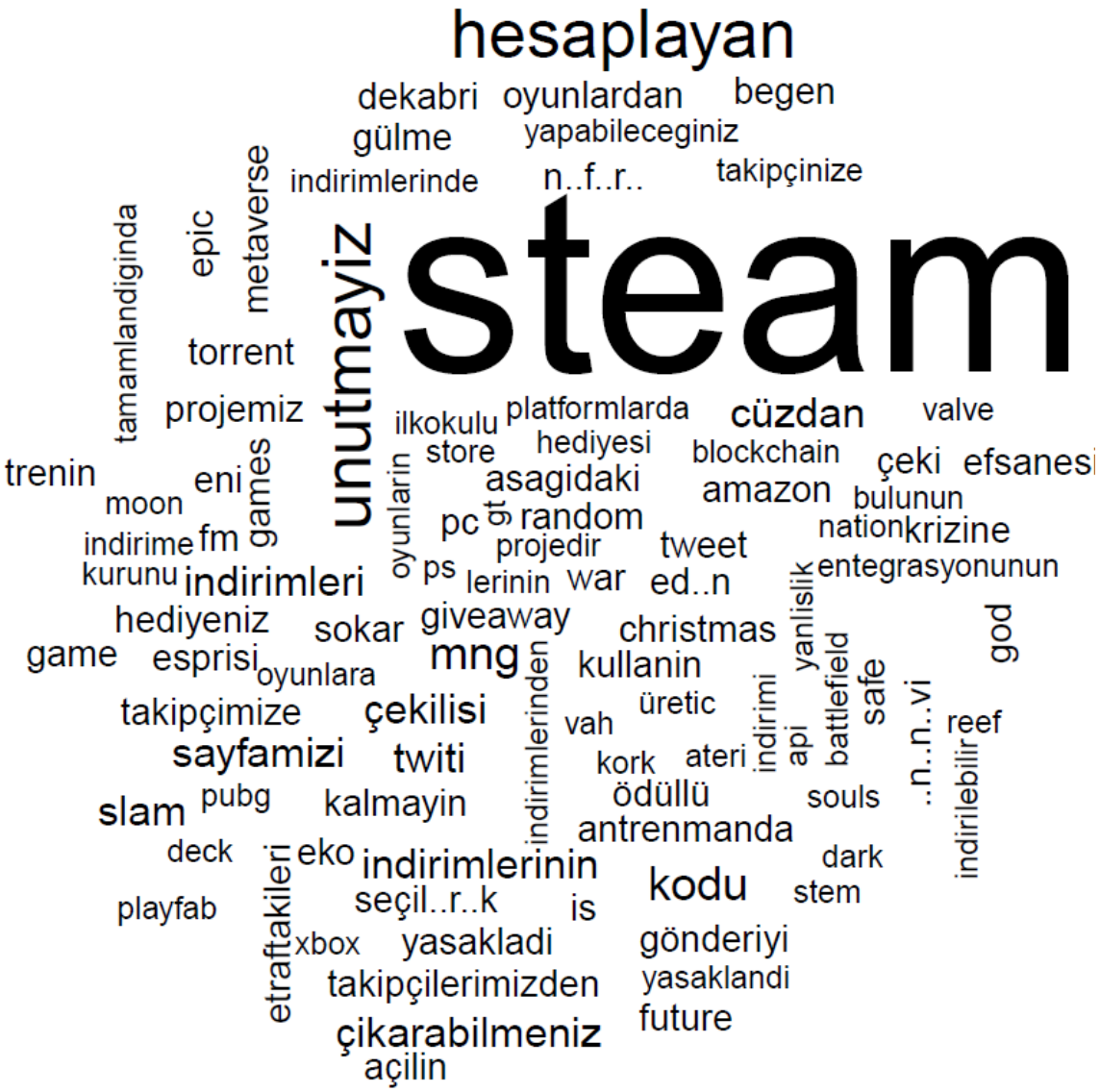
```
147
148 tidy_tweets <- tweet_clean %>% select(text) %>%
149   mutate(linenumber = row_number()) %>% unnest_tokens(word, text)
150 tidy_tweets <- tidy_tweets %>% anti_join(turkish_stopwords, by=c("word"="STOPWORD"))
151
```

Hemen ardından görselleştirme işlemine başlayabiliriz. Tweet'lerde 50 defadan fazla tekrar eden kelimeleri sıralı bir şekilde bar grafiğinde gösteriyoruz.

```
153 tidy_tweets %>%
154   count(word, sort = TRUE) %>%
155   filter(n > 50) %>%
156   mutate(word = reorder(word, n)) %>%
157   ggplot(aes(word, n)) +
158   geom_col() +
159   xlab(NULL) +
160   coord_flip() + theme_minimal() +
161   ggtitle("tweetlerde en cok kullanilan kelimeler")
```


tweetlerde en cok kullanilan kelimeler
steam
hesaplayan
unutmayiz
kodu
mng
cüzdan
çekilisi
indirimlerinin
twiti
slam
sayfamizi
çikarabilmeniz
indirimleri
epic
efsanesi
fm
açilin
tweet
trenin
torrent
war
god
hediyeniz
gönderiyi
kalmayin
yasakladi
game
pc
takipçimize
games
oyunlardan
ödüllü
eni
random
giveaway
christmas
seçil..r..k
n..f..r..
ed..n
asagidaki
amazon
..n..n..vi
kullanin
dekabri
esprisi
krizine
gülme
sokar
etraftakileri
antrenmanda
takipçilerimizden
begen
çeki
projemiz
is
future
eko
safe
metaverse
deck
pubg
reef
moon
playfab
stem
kurunu
indirimi
0
2000
4000
6000
n

Daha sonra ise en fazla 100 kelime içerecek şekilde bir kelime bulutu oluşturuyoruz. En fazla tekrar eden kelime en büyük, en az tekrar eden kelime ise en küçük olacak şekilde oranlanmıştır.



Duygu Analizi

Duygu analizi yapmak için öncelikle farklı duygu grupları olması gerekir. Bu projede sadece pozitif, negatif ve nötr duyguları bulunmaktadır. Pozitif ve negatif kelimelerin bulunduğu txt dosyalarını projeye ekliyoruz. Özet bilgilerine summary() komutu ile erişebiliyoruz. İçeriklerinden birkaçını görmek için ise dput() komutunu kullanabiliriz.

```
178 pos.words <- scan('C:/Users/samet/Desktop/twitter_steam_text_mining/pozitifkelimeler.txt', what = 'character', comment.char = ';', skipNul = TRUE
179 dput(pos.words[1289:1293])
180 neg.words <- scan('C:/Users/samet/Desktop/twitter_steam_text_mining/negatifkelimeler.txt', what = 'character', comment.char = ';', skipNul = TRUE
181 dput(neg.words[1967:1971])
```

```
> summary(pos.words)
   Length     Class      Mode
     1293 character character
```

```
> dput(pos.words[1289:1293])
c("zenginlik", "zerafet", "zevk", "zevkle", "zevkli")
```

```
> summary(neg.words)
   Length     Class      Mode
     1981 character character
```

```
> dput(neg.words[1967:1971])
c("zindan", "ziyan", "zor", "zoraki", "zorba")
```

Ardından duygu skoru oluşturmak için bir kod bloğu yazıyoruz. Kendi içinde temizleme işlemleri barındırıyor. Duygu analizi için en önemli kısım olan match() komutu ise tweetlerde bulunan kelimeleri pozitif ve negatif kelimeler sözlüğümüzle kıyaslamasıdır. Ancak bu komut bize eşleşen kelimelerin pozitif ve negatif kelimeler sözlüğünde kaçıncı kelimelere denk geldiğinin sonucunu veriyor ve cümlelere duygu analizi yapabilmek için sözlükten kaçıncı kelimeye denk geldiğinden ziyade sözlükten bir kelimeye eşleşip eşleşmediğini bulmamız gerekir. Bu sebeple true/false içeren bir kod eklemesi yapıyoruz. Ardından sum() komutunu kullanarak bir skor formülü oluşturuyoruz. Burada sum() komutu true/false değerlerini 1 ve 0 olarak değerlendirecek. Yani pozitifle eşleşen ve negatifle eşleşmeyen bir cümle skor = 1-0 sonucundan pozitif, pozitifle eşleşmeyen ve negatifle eşleşen bir cümle skor = 0-1 sonucundan negatif, pozitifle eşleşen ve negatifle eşleşen bir cümle veya her ikisiyle de eşleşmeyen bir cümle de skor = 1-1, skor = 0-0 sonuçlarından nötr olacaktır.

```r
score.sentiment <- function(sentences, pos.words, neg.words, .progress='none')

{
  require(plyr)
  require(stringr)

  scores <- laply(sentences, function(sentence, pos.words, neg.words)

  {

    # clean up sentences with R's regex-driven global substitute, gsub() function:
    sentence <- gsub('https://','',sentence)
    sentence <- gsub('http://','',sentence)
    sentence <- gsub('[^[:graph:]]', ' ',sentence)
    sentence <- gsub('[[:punct:]]', '', sentence)
    sentence <- gsub('[[:cntrl:]]', '', sentence)
    sentence <- gsub('\\d+', '', sentence)
    sentence <- str_replace_all(sentence,"[^[:graph:]]", " ")
    # and convert to lower case:
    sentence <- tolower(sentence)

    # split into words. str_split is in the stringr package
    word.list <- str_split(sentence, '\\s+')
    # sometimes a list() is one level of hierarchy too much
    words <- unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches <- match(words, pos.words)
    neg.matches <- match(words, neg.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches <- !is.na(pos.matches)
    neg.matches <- !is.na(neg.matches)

    # TRUE/FALSE will be treated as 1/0 by sum():
    score <- sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words, .progress=.progress )

  scores.df <- data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

Bu kod bloğunun ardından analizi hazırlamak ve duygu skor frekans tablosunu oluşturmak geliyor.

```
228
229 analysis <- score.sentiment(tweet_clean$text, pos.words, neg.words)
230 # sentiment score frequency table
231 table(analysis$score)
232
```

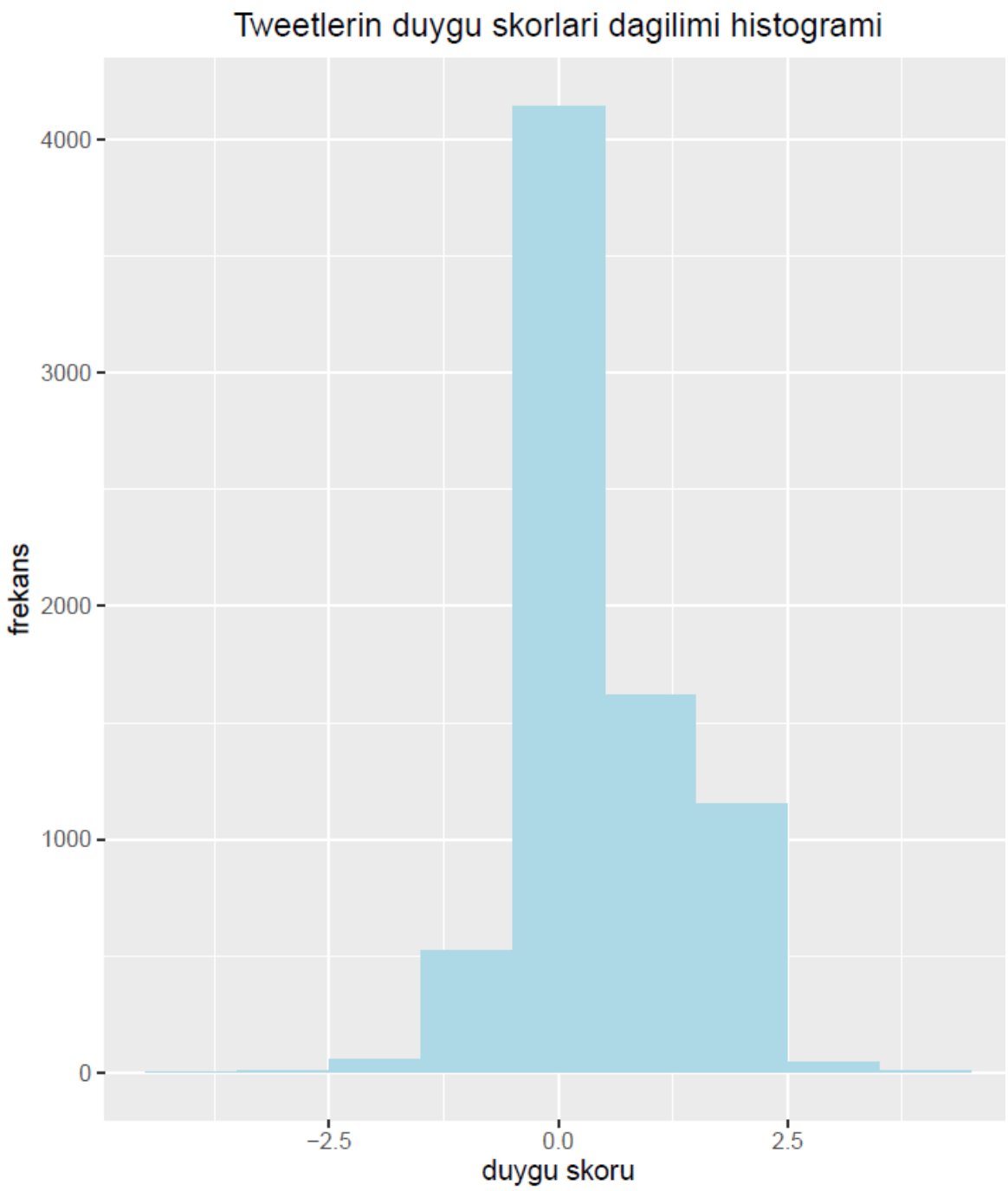
Yukarıdaki skor formülüne göre bir tablo oluşturursak en düşük skor, yani en negatif yorumların skorları -4 çıkmış ve bu skora sahip 2 adet yorum bulunuyor. En yüksek, yani en pozitif yorumların skorları ise 4 çıkmış ve bu skora sahip 6 adet yorum bulunuyor. 4145 adet ise nötr yorum bulunmaktadır.

```
> table(analysis$score)

  -4   -3   -2   -1    0    1    2    3    4
   2    7   57  522 4145 1617 1152   44    6
```

Bu skor dağılımlarını daha iyi görmek için bir histogram grafiğinde gösterelim. Histogram grafiğini oluştururken ggplot paketini kullanıyoruz.

```
233
234 analysis %>%
235   ggplot(aes(x=score)) +
236   geom_histogram(binwidth = 1, fill = "lightblue")+
237   ylab("frekans") +
238   xlab("duygu skoru") +
239   ggtitle("Tweetlerin duygu skorlari dagilimi histogrami") +
240   ggeasy::easy_center_title()
241
```


Tweetlerin duygu skorlari dagilimi histogrami
4000
3000
2000
1000
0
frekans
−2.5
0.0
2.5
duygu skoru

Son olarak bu skorların 0'ın üzeri pozitif, 0'ın altı negatif ve 0'a eşit olanlarını nötr olarak gruplandırmak kalıyor. Bu aşamada artık pozitif ve negatif değerlerde 0'ın ne kadar üzerinde veya ne kadar altında olduğunun bir önemi kalmıyor. Son grafiği de ggplot paketiyle oluşturuyoruz. Artık 7500 adet tweet'in duygu tipi bar grafiği de oluşmuş oluyor.

```
242
243 neutral <- length(which(analysis$score == 0))
244 positive <- length(which(analysis$score > 0))
245 negative <- length(which(analysis$score < 0))
246 Sentiment <- c("pozitif","notr","negatif")
247 Count <- c(positive,neutral,negative)
248 output <- data.frame(Sentiment,Count)
249 output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
250 ggplot(output, aes(x=Sentiment,y=Count))+
251   geom_bar(stat = "identity", aes(fill = Sentiment))+
252   ggtitle("7500 tweetin duygu tipi grafigi")
253
```

Duygu analizinin sonucunda 7500 adet tweet içerisinden büyük çoğunluğu nötr tweetlerin oluşturduğunu görmekteyiz. Yaklaşık 4250 adet nötr tweet bulunuyor. Nötr tweet'lerin ardından ise pozitif tweet'ler geliyor. Yaklaşık 2750 adet pozitif tweet bulunuyor. Son sırada ise negatif tweet'ler bulunuyor. Yaklaşık 600 adet negatif tweet bulunmaktadır. Nötr tweet'ler bütün tweet'lerin yaklaşık %56'sını, pozitif tweet'ler bütün tweet'lerin yaklaşık %36'sını ve negatif tweet'ler ise bütün tweet'lerin yaklaşık %8'ini oluşturmaktadır.