T. C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ

VERİ MADENCİLİĞİ

TWİTTER İÇERİK ANALİZİ

HAZIRLAYAN

BETÜL ÜSTÜN -57046617938

BİLECİK,2022

İÇİNDEKİLER

ÖNSÖZ

Bu projenin ortaya çıkmasına vesile olan, ortaya koyduğu özveriyle bana programlamayı tekrar sevdiren kıymetli hocam Dr. Nur Kuban TORUN'a teşekkürü kendime borç bilirim.

ÖZET

Veri tabanları veya dosyalarda bulunan verilerin belirli istatistik yönetmeleri kullanılarak kullanılabilir hale getirilmesi işlemine veri madenciliği denir. Veri madenciliği tüm şirketler için son derece önem arz etmektedir. Müşterilerin tepkilerini, müşterilerin almış oldukları hizmetlerden duydukları memnuniyet durumlarını sosyal medya platformlarından paylaşması işletmelerde müşteri odaklı faaliyetlerin artmasına neden olmuştur. Veri madenciliği ve Twitter API aracılığıyla metinler çekilerek insanların duyguları ve en çok kullanılan kelimeler analiz edip çıkarımlarda bulunulabilir. Bu projede "R Studıo" yardımıyla twitter üzerinden veriler çekerek onlar üzerinde metin analizi, duygu analizi yapacağım. Bu proje de konu olarak #kitap etiketini kullanacağım. İnsanların kitap kelimesi hakkında duygularını ve ne gibi çıkarımlarda bulunduğu inceleyeceğim.

1.GİRİŞ

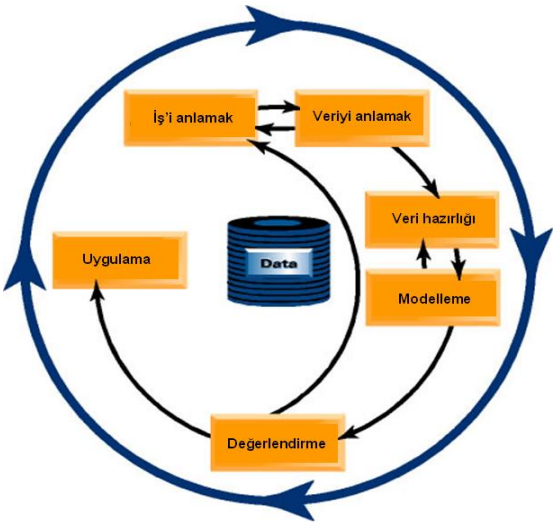
1.1 Veri Madenciliği

Veri madenciliği, büyük miktardaki veri kümesi içerisinden gelecekle ilgili tahmin yapmamızı sağlayacak ilişki ve kuralların aranmasıdır. Kısaca kaynaklardan toplanan geniş hacimde verilerin toplanması ve saklanmasının ve bir sonucu üzerinde işlem yapılarak anlamlı bilgilerin çıkarılmasıdır. Özel sektör ve kamu kuruluşlarında birçok şekilde kullanılmaktadır.

Büyük çapta metin içeren Twitter üzerinden metin madenciliği çalışmaları sıklıkla yapılmaktadır. Pazarlama faaliyetlerinde, dizilerin-filmlerin reyting skorlamalarında, satış tahminlerinde Twitter metin verileri üzerine yapılan analizler büyük önem taşır ve reklam şirketlerince kullanılır.

1.2 Veri Madenciliği Süreçleri

1. aşama: Problemin Tanımlanması
2.aşama: Veriyi Anlama
3.aşama: Verinin Hazırlanması
4.aşama: Modelleme
5. aşama: Değerlendirme
6. aşama: Yayılım



Veri Madenciliği Süreçleri Tablosu

2. METİN MADENCİLİĞİ (TEXT MINING)

Teknikler	İşlem Adımları	Kütüphaneler
	Veri setlerinin oluşturulması Metin ön işleme • Metni küçük harfe dönüştürme • Gereksiz karakterlerin ve kelimelerin kaldırılması ○ Retweet ○ @kullanıcı adı ○ Noktalama işaretleri ○ Rakamlar ○ Linkler ○ Tabs ○ Cümle başındaki ve sonundaki boş alanlar ○ Stop words	Library (twitteR) Library (ROAuth) Library (tm) Library (stringi)
Metin madenciliği (Text mining)	Kelime bulutu (Word cloud)	Library (RColorBrewer) Library (wordcloud)

Tablo 1: Veri analizleri için kullanılan teknikler, işlem adımları ve kütüphaneler.

Metin madenciliği, veri kaynağı metin olan veri madenciliği (data mining) çalışması olup yapısal olmayan verinin yapısal hâle dönüştürülmesini sağlamaktadır. Diğer bir deyişle metin madenciliği, metin veri setinin sayısal veri setine çevrilmesi sırasında uygulanan doğal dil işleme tekniklerinin tümünü ifade eder.

Bu projede Twitter'dan KİTAP ile ilgili Twitter API aracılığıyla veriler çekilerek metin veri setleri oluşturulmuştur. Daha sonra metin içeriğinin yapısal hâle dönüştürülmesi için bir takım metin ön işleme adımları uygulanır. Gereksiz kelimelerin temizlenmesi, Rt ifadelerinin kaldırılması, URL linklerinin temizlenmesi, Hastag kullanımlarının kaldırılması Türkçe Stopwords kelimler kaldırılması vb. işlemler yapılarak veri seti temiz hale getirilir.

Metin ön işleme işlemleri tamamlanır. Böylece metin madenciliği işlem adımları tamamlanarak metin veri setleri kelime bulutu (word cloud), duygu analizi (sentiment analysis) ve otomatik linear model (automatic linear modeling) yöntemleri için hazır hâle getirilmiş olur.

3. R ile Metin Madenciliği

3.1 İçerik Analizi

Bu çalışmada Twitter üzerinde KİTAP ile ilgili yapılan paylaşımlar ve retweetler dikkate alınarak veri incelemeleri yapılmıştır.

R üzerinde twitterdan veri çekmek için ilk olarak bir geliştirici hesaba ihtiyaç vardır. Twitter Developer ile iletişime geçerek bir geliştirici hesap temin edilir.

Geliştirici hesap alındıktan sonra Developer hesabına giriş yapılarak bir uygulama oluşturulur. Ve bu uygulamaya bir isim verilir. Detaylı açıklama yapıldıktan sonra twitter üzerinden kişiye özel tanımlanan TOKEN'lar verilir.

Tanımlanan rastgele kodlar **"Api_key", " Api_key_secret", "access_token", "Access_token_secret"** bunlardır. R ile Twitter arasında bağ kurmak için bu kodlara ihtiyacımız vardır.

R sisteminde yapacağımız işlemlere geçtiğimizde ilk olarak paket kurulumu yapacağız. Paket kurulumu için **"install.packages()"** komutunu kullanacağız. Yüklediğimiz paketleri **"library()"** komutuyla aktif hale getiriyoruz. Hata almamak için paketin çalışıp çalışmadığını kontrol etmek son derece önemlidir.

```r
#Tüm paketlerin yüklenmesi
install.packages("twitteR") #Twitter veri çekme için kullanılır.
install.packages("ROAuth") #Twitter'da ki uygulamaya giriş yapmak ve iletişim kurmak için kullanilirr.
install.packages("openssl") #imzalar ve sertifakalar için araç seti
install.packages("httpuv") #HTTP ve WebSocket Sunucu Kitapligi
install.packages("tm") #veri madenciliÄŸi iÅŸin kullanÄ±lÄ±r.
install.packages("readxl") #ecxel dosyalarini okur
install.packages("tidytext")
install.packages("wordcloud") #kelime bulutu oluşturmada kullanılır.
install.packages("ggplot2") #oluşturacagimiz grafiklerini görüntülemek için kullaniriz.
install.packages("stringr") #String verilere yani metinsel verilere manipülasyon için kullanilir.
install.packages("writexl") #verileri excel formatına aktarmak için kullanılır.

library(twitteR)
library(ROAuth)
library(openssl)
library(httpuv)
library(stringi)
library(stringr)
library(tm)
library(readxl)
library(tidytext)
library(dplyr)
library(ggplot2)
library(wordcloud)
library(writexl)

```

Görselde de görüldüğü üzere paketlerimiz yükleyip çalıştırıyoruz.

Sonrasında Twitter'da ki uygulamaya giriş yapıp ve iletişim kurulur. Bunu için de Twitter Developer hesabından bizim adımıza tanımlanan TOKEN'lar girilir.

```r
api_key
api_key_secret
access_token
access_token_secret


setup_twitter_oauth(api_key,api_key_secret,access_token,access_token_secret)
```

Kimlik doğrulama ve bağlantı kurma başarılıdır ibaresini aldıktan sonra veri çekme işlemine başlarız. Bunun için **"Using direct authentication"** mesajını almanız gerekir.

```
> setup_twitter_oauth(api_key,api_key_secret,access_token,access_token_secret)
[1] "Using direct authentication"
```

```r
tweets<- searchTwitteR('#kitap', n=8165,  locale = "tr_TR")
```

```r
tweets.df <- twListToDF(tweets)
tweet_clean <- tweets.df

tweet_clean$text <- stri_enc_toutf8(tweet_clean$text)
```

searchTwitteR () paketi kullanarak tweet araması yapılmaya başlanır. Konuyla alakalı tweetler çekilir. Bu çalışmada günlük olarak **"#kitap"** hashtagi ile atılmış son 8165 adet tweeti incelemek istiyorum.

```
#RT ifadelerinin kaldirilmesi

tweet_clean$text <- ifelse(str_sub(tweet_clean$text,1,2) == "RT",
                           substring(tweet_clean$text,3),
                           tweet_clean$text)
```

Çekilen tweetler incelendiğinde bazılarının rt(retweet) olduğu gözükmektedir. Bu yüzden rt ifadelerini temizlemek için yukarıda ki görselde ki kodu kullanırız.

```
#URL linklerinin kaldirilmasi

tweet_clean$text <- str_replace_all(tweet_clean$text, "http[^[:space:]]*", "")

#hastag "#" ve "@" isaretlerinin temizlenmesi
tweet_clean$text <- str_replace_all(tweet_clean$text, "#\\S+", "")
tweet_clean$text <- str_replace_all(tweet_clean$text, "@\\S+", "")

#noktalama isaretlerinin kaldirilmesi
tweet_clean$text <- str_replace_all(tweet_clean$text, "[[:punct:][:blank:]]+", " ")

#tum harflerin kucuk harfe donusturulmesi
tweet_clean$text  <- str_to_lower(tweet_clean$text, "tr")


#Rakamlarin temizlenmesi
tweet_clean$text <- removeNumbers(tweet_clean$text)

#ASCII formatina uymayan karaketerlerin temizlenmesi
tweet_clean$text <- str_replace_all(tweet_clean$text, "[<].*[>]", "")
tweet_clean$text <- gsub("\uFFFD", "", tweet_clean$text, fixed =  TRUE)
tweet_clean$text <- gsub("\n", "", tweet_clean$text, fixed =  TRUE)

#Alfabetik olmayan karakterlerin temizlenmesi
tweet_clean$text <- str_replace_all(tweet_clean$text, "[^[:alnum:]]"," " )

#etkisiz kelimelerin stopwords analizinden cikarilmasi
Turkish_Stopwords <- read.csv("Turkish-Stopwords.csv")
head(Turkish_Stopwords)
```

Tweetlerin içerisinde hala; rakamlar, URL linkler, “@” ve “#” sembolleri, emojiler ve yabancı içerikleri gösteren unicode ifadeler gibi temizlenmesi gereken bir çok içerik bulunmaktadır. Bunlar içinde yukarı da olan kodlar bütününü kullandım.

```
#Excel formatinda export alma
#install.packages("writexl")
library("writexl")

write_xlsx(tweet_clean, "D:/Veriler/temizveri_8ocak.xlsx")
```

Çektiğim verileri temizleme işleminden sonra excel formatında kaydetmek için **“writexl”** paketini kullandım. Excel için “.xlsx” formatında kaydedilmeli.

3.2 Tidytext İle İçerik Analizi

Data frame haline getirdiğimiz metin henüz düzenli bir metin formatında değil. Elimizdeki metin birleşik kelimelerden (cümle) oluştuğu için metin içerisindeki en sık ortaya çıkan kelimeleri bulamayız.
Bu yüzden, cümleleri kelimelere parçalamamız gerekiyor. Bunu yaparken de her bir döküman için satır başına bir kelime olacak şekilde dönüştürmeliyiz.

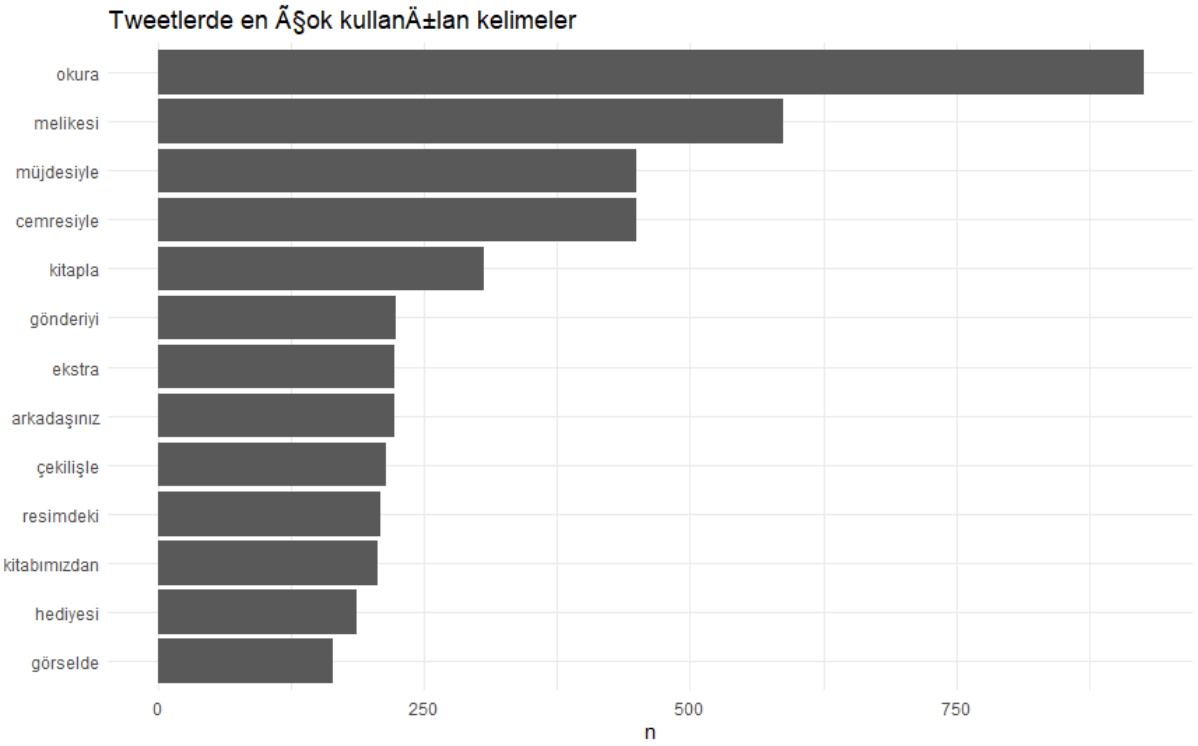
Tokenization işlemi metni kelimelere ayırma işlemidir. Analizlerimizi yapabilmemiz için metni kelimelere bölmemiz gerekiyor.
Metni hem birer kelimelere ayırmak hem de toplu bir şekilde (ikişer, üçer …) ayırmak için tidytext kütüphanesinden unnest_tokens() fonksiyonu kullanılır.

```
#tidytext analizi
#install.packages("tidytext")
library(tidytext)
library(dplyr)
library(ggplot2)

tidy_tweets <- tweet_clean %>% select(text) %>%
  mutate(linenumber = row_number()) %>% unnest_tokens(word, text)
tidy_tweets <- tidy_tweets %>% anti_join(Turkish_Stopwords, by=c("word"="STOPWORD"))
head(tidy_tweets)


tidy_tweets %>%
  count(word, sort = TRUE) %>%
  filter(n > 500) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() + theme_minimal() +
  ggtitle("Tweetlerde en cok kullanilan kelimeler")
```

Çektiğimiz ve temizleme işlemi gerçekleştirdiğimiz veri setinde en çok kullanılan kelimeleri bulmak adına yukarıda bulunun görselde ki kodları yazıyoruz. En çok kullanılan kelimeleri **“ggplot2”** paketiyle görselleştiririz. Bu çalışmada inceleme adına en çok kullanılan 500 kelimeyi tablo halinde çağırıyoruz.



Tablo 2: En Çok Kullanılan Kelimler Bar Grafiği

En sık kullanılan 500 kelimeye bakıldığında görüyoruz ki KİTAP ile beraber, hediyesi, okur, melikesi, müjdesiyle, cemresiyle, kitapla, gönderiye, ekstra, arkadaşınız, çekilişle, resimdeki, kitabımızdan ve görselde gibi kavramlar sıklıkla kullanılmış. Demek oluyor ki, bu kelimeler kitapla yakın alakalı, yakın ilişkili veya analizi riske atacak yanıltacak kelimelerden oluşuyordur.

Bunlara tek tek bakarsak eğer son zamanlarda çıkan Hüzün Melikesi adlı kitapla ilgili kelimelerin sık kullanıldığını görüyoruz bu duruma bakarak Hüzün Melikesi adlı kitabın bu dönem popüler kitap olduğu çıkarımında bulunabiliriz.

Diğer kelimelere baktığımızda ise kitabın çekilişle reklam amaçlı veya gerçek bir hediye için kullanıldığını gözlemleyebiliyoruz.

Analiz işlemini bozan (“görselde, ”ekstra”, “resimdeki”) kelimelerin Türkçe stopwords (etkisiz kelimler) dosyası içine ekleyerek tablomuz içerisinden çıkarıyoruz. Temiz bir analiz yapmak istiyorsak eğer etkisiz kelimelerin çıkarılması gerekir. Analiz işlemini riske atabilir ve yanıltabilir.

3.3 Kelime Bulutu Oluşturma

Kelime bulutu, metin veri setleri için kullanılan veri görselleştirme yöntemidir. Kelime bulutu frekansa dayalı bir görselleştirme yöntemi olduğundan metinlerin görsel bir özetini oluşturur.8 Kelime bulutunu oluşturmak için word cloud ve R Color Brewer kütüphaneleri ile word cloud () fonksiyonu kullanılmıştır. Kelime bulutları en sık kullanılan 100 kelime üzerinden oluşturulmuştur.

```
105
106 #Kelime bulutu haline getirme
107 tidy_tweets %>%
108   count(word) %>%
109   with(wordcloud(word, n, max.words = 100))
110
111
```

En çok kullanılan kelimeleri wordcloud() komutu ile kelime bulutu haline getiriyoruz.

Tablo 3: Kelime Bulutu

Kelime bulutunda çıkan diğer kelimelere baktığımızda ise kitapla alakalı “Stefan”, “Semerkand”, “Dostoyevski”, “Zweig”, “Editörlük”, “Yayınevi” vb. birbiriyle bağlantılı ve alakalı kelimler olduğunu görüyoruz.

Bunu yanı sıra bir o kadar da “retweet”, “beğenip”, ”hakkı”, “başlayalım”, “tıklayın”, “örneğ” gibi analizi bozan alakasız kelimelerde bulunmaktadır. Bu yüzden verilerin temizlenmesi kısmı analiz için mühim bir öneme sahiptir.

4. 21 GÜNLÜK VERİ SETİ İLE İÇERİK ANALİZİ

Temizlemiş olduğumuz verileri birleştirerek yeni bir veri seti oluşturarak en çok kullanılan kelimelere ve kelime bulutuna bakacağız.

```
#21 günlük birleştirilmiş olan veri setinin içeriye aktarılması "temizveri.csv" dosyası
temizveri =read.table(file.choose(), header = T, sep = ";")
```

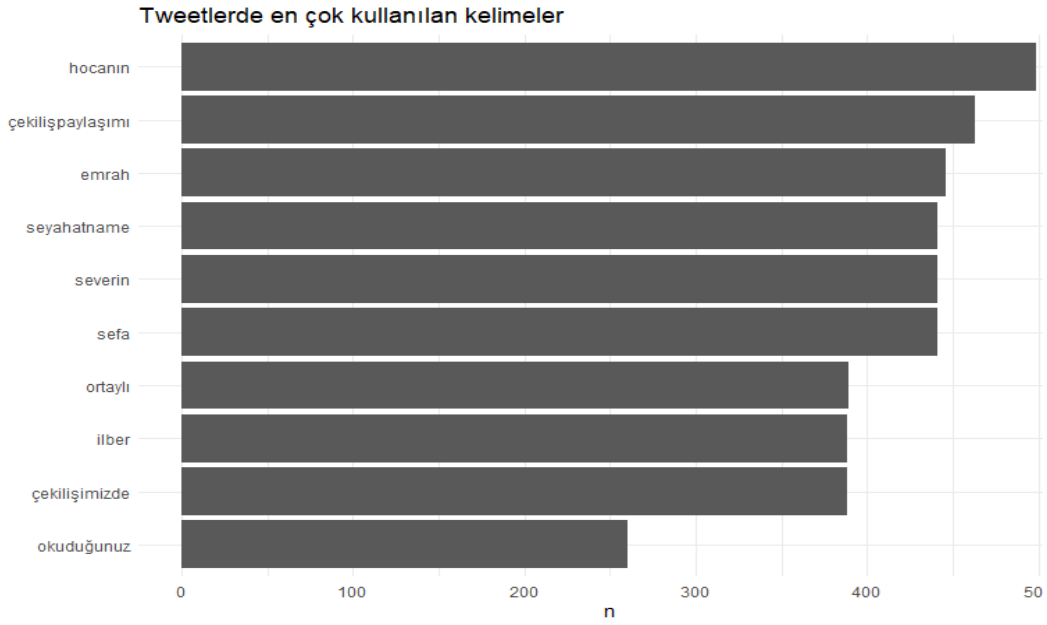
Veri setini yukarıda olan kod bütünüyle çekiyoruz. Birleştirdiğimiz veri setini cvs virgülle ayrılmış hale getirmemiz gerekiyor.

```
#21 günlük verilerden etkisiz kelimelerin çıkartılması
temiz_twit <- temizveri %>% select(text) %>%
  mutate(linenumber = row_number()) %>% unnest_tokens(word, text)
temiz_twit <- temiz_twit %>% anti_join(Turkish_Stopwords, by=c("word"="STOPWORD"))
head(temiz_twit)
```

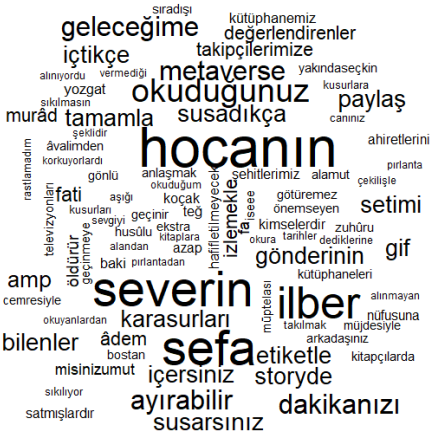
Veri setimiz de bulunan etkisiz kelimeleri yukarı da bulunan kod ile çıkartıyoruz. Analiz yaparken etkisiz kelimeler analiz işlemini bozabilir.

```
#Twitlerde toplamda 250'den daha fazla kullanılan kelimelerin listelenmesi
temiz_twit %>%
  count(word, sort = TRUE) %>%
  filter(n > 250) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() + theme_minimal() +
  ggtitle("Tweetlerde en çok kullanılan kelimeler")
```

Veri setimizin içerisinde geçen 250 kelimeyi ve kelime bulutu grafiğini oluşturuyoruz.



```
#Kelime bulutu
temiz_twit %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



Tablo4: Kelime Bulutu Grafiği

21 günlük veri setimiz de “#Kitap” için en çok kullanılan kelimeler ve kelime bulutu grafiklerine baktığımızda karşımıza “İlber” ve “Ortaylı” kelimelerinin çıktığını görüyoruz buradan 21 gün içerisinde İlber Ortaylı’nın popüler bir yazar olduğu varsayımında bulunabiliriz. Ve hakkında yorumlarda bulunulduğu tespitine varıyoruz.

Diğer çıkan kelimelere bakacak olursak “Emrah”, “Sefa”, “Hoca” kelimeleri hakkında çıkarımda bulunacak olursak Emrah Sefa Gürkan adlı tarihçinin yazdığı kitap ile karşılaşıyoruz. Ve çekiliş olarak tarih kitaplarının verildiğini gözlemleyebiliyoruz.

“Seyahatname” kelimesinde ise Evliya Çelebinin eseriyle karşılaşıyoruz. Yine 21 günlük ver setinde popüler kitaplar arasındadır.

5.SAAT BAŞINA ATILAN TWEETLER

```
#En önce ve en son atılan twitlerin gösterilmesi
tweets.df %>% pull(created) %>% min()
tweets.df %>% pull(created) %>% max()
```

İlk ve son atılan twitlerin zamanının gösterilmesi yukarı da bulunan kodlarla gösterilir.

```
> #En önce ve en son atılan twitlerin gösterilmesi
> tweets.df %>% pull(created) %>% min()
[1] "2022-01-11 22:52:18 UTC"
> tweets.df %>% pull(created) %>% max()
[1] "2022-01-20 13:28:21 UTC"
```

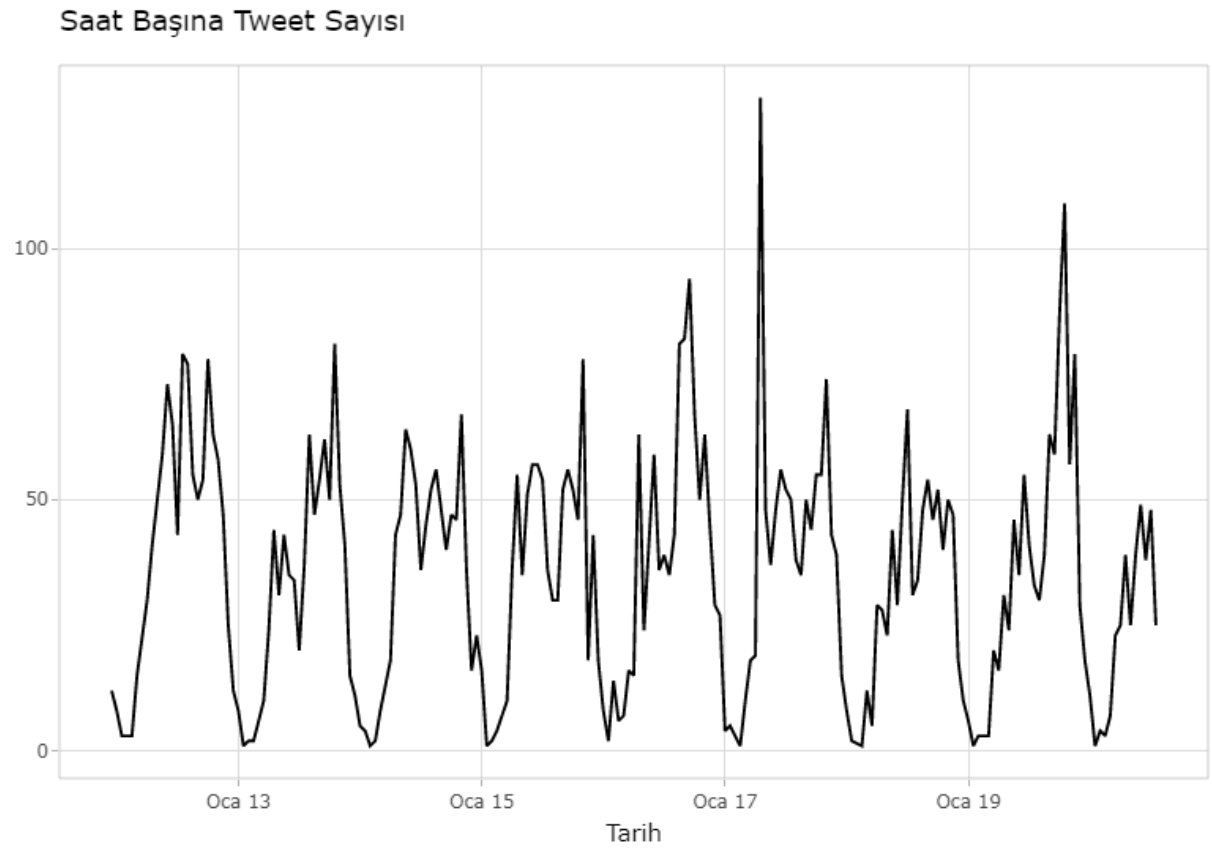
Çıktı olarak yukarıda ki saatleri görmekteyiz.

```r
cleanText <- clean.text(tweets.txt)
# bos sonuçları kaldır (varsa)
idx <- which(cleanText == " ")
cleanText <- cleanText[cleanText != " "]

tweets.df %<>%
  mutate(
    created = created %>%
      # Sıfırları kaldırın.
      str_remove_all(pattern = '\\+0000') %>%
      #Ayrıştırma tarihi..
      parse_date_time(orders = '%y-%m-%d %H%M%S')
  )

tweets.df %<>%
  mutate(Created_At_Round = created%>% round(units = 'hours') %>% as.POSIXct())
#En önce ve en son atılan twitlerin gösterilmesi
tweets.df %>% pull(created) %>% min()
tweets.df %>% pull(created) %>% max()
#Atılan twitlerin saatlere göre sayısınının grafik halinde gösterilmesi
plt <- tweets.df %>%
  dplyr::count(Created_At_Round) %>%
  ggplot(mapping = aes(x = Created_At_Round, y = n)) +
  theme_light() +
  geom_line() +
  xlab(label = 'Tarih') +
  ylab(label = NULL) +
  ggtitle(label = 'Saat Başına Tweet Sayısı')
```

Yukarıda ki kod bütünü ile saat başına atılan tweetleri grafik halinde görebiliriz.



Tablo 5: Saat Başına Atılan Tweetler Grafiği

6.DUYGU ANALİZİ

Genel olarak pozitif, negatif ve nötr dilin ölçümü anlamına gelen duygu analizi, fikir madenciliği (opinion mining) olarak da anılır. Nitel araştırmanın bir yönünü oluşturan bu analizle ürünlerden, reklamlardan, lokasyonlardan, reklamlardan ve hatta rakiplerden ortaya çıkan müşteri düşüncelerini açığa çıkarılabilmektedir. Müşterilerin hoşlandığı veya hoşlanmadığı ürünlerin tespitinde bu analiz önemli rol oynayarak ürün gamı ve ürünün kalitesi şekillendirilebilmektedir. Diğer bir deyişle, firmalar müşterilerin ne hissettiğini ve düşündüğünü bilerek müşteri beklentilerini daha iyi karşılayabilir. Sağlık sektöründe de benzer durum söz konusu olabilir. Hasta beklentileri, algıları ve yönetiminin analizleri ile sağlık krizlerinde fikirlerin analizinde duygu analizlerinin yoğun bir şekilde kullanıldığı literatürden yakinen bilinmektedir. Literatürde aynı zamanda duygu analizlerinden önce kelime bulutlarının oluşturulduğu da görülmektedir.

Tablo 6: Duygu Analizi Kullanım Alanları

Tablo 7: Duygu Analizinde Kullanılan Duygu Sözcükleri

Duyarlılık puanı frekans tablosu:

```
# duyarlılık puanı frekans tablosu
table(analysis$score)

analysis %>%
```

Çıktı olarak:

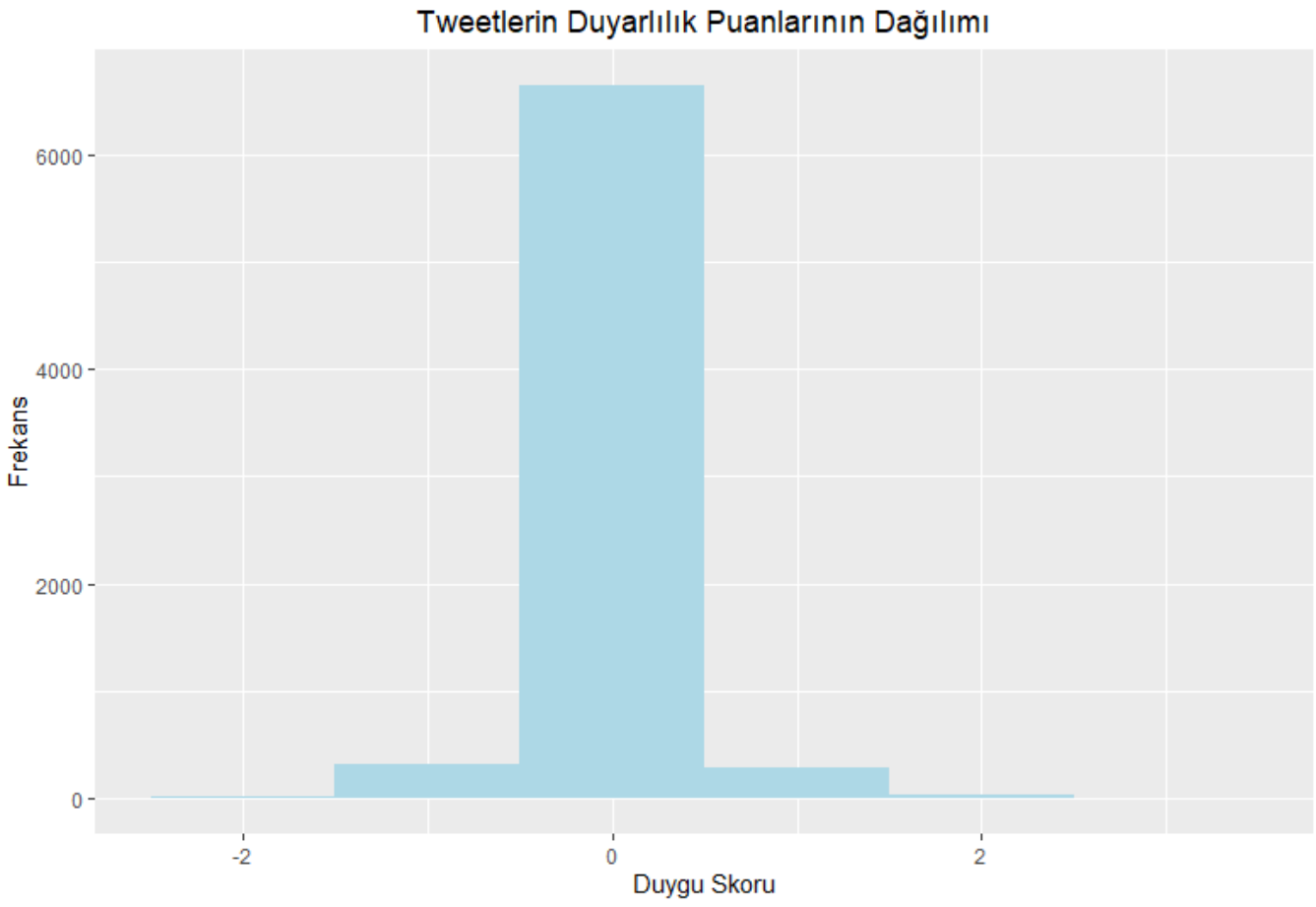
```
> # duyarlılık puanı frekans tablosu
> table(analysis$score)

  -2   -1    0    1    2    3
  18  319 6647  287   26    3
> analysis %>%
```

Duyarlılık Puanlarının grafik halinde gösterilmesi aşağıdaki gibidir.

```
analysis %>%
  ggplot(aes(x=score)) +
  geom_histogram(binwidth = 1, fill = "lightblue")+
  ylab("Frekans") +
  xlab("Duygu Skoru") +
  ggtitle("Tweetlerin Duyarlılık Puanlarının Dağılımı") +
  ggeasy::easy_center_title()
```

Grafiği oluştumak için yukarıda bulunan kodlar kullanılır.



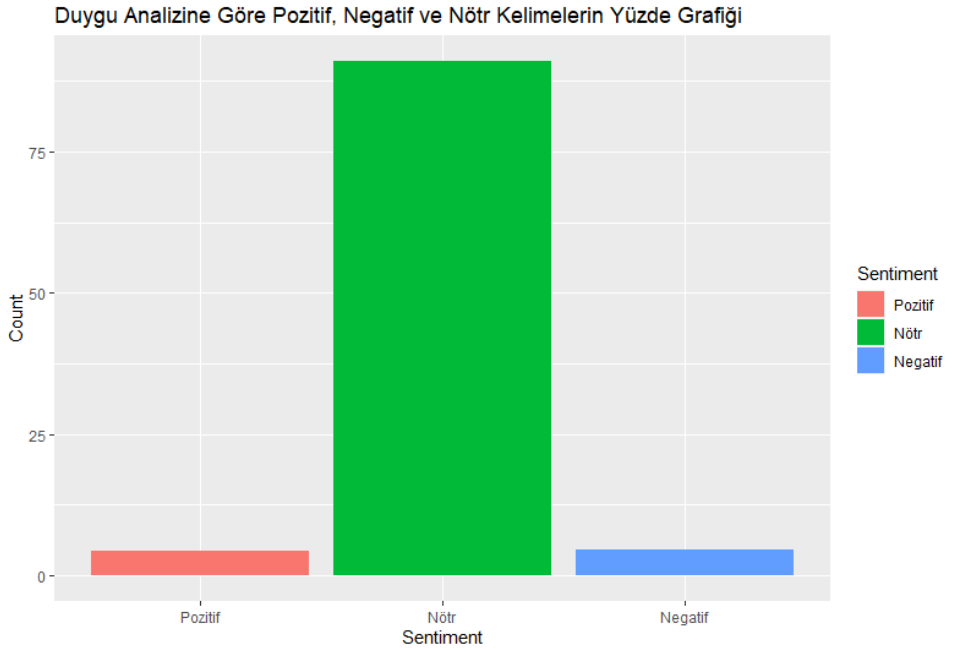
Tablo 8: Tweetlerin Duyarlılık Puanlarının Dağılımı Grafiği

```
neutral <- length(which(analysis$score == 0))
positive <- length(which(analysis$score > 0))
negative <- length(which(analysis$score < 0))
toplam=positive+neutral+negative
Sentiment <- c("Pozitif","Nötr","Negatif")

Count <- c((positive/toplam)*100,(neutral/toplam)*100,(negative/toplam)*100)
output <- data.frame(Sentiment,Count)
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
ggplot(output, aes(x=Sentiment,y=Count,))+

  geom_bar(stat = "identity", aes(fill = Sentiment ))+
  ggtitle("Duygu Analizine Göre Pozitif, Negatif ve Nötr Kelimelerin Yüzde Grafiği")
head((positive/toplam)*100,"Pozitif")
head((neutral/toplam)*100 ,"Nötr")
head((negative/toplam)*100 ,"Negatif")
```

Duygu analizinde olan pozitif, negatif, nötr olan kelimelerin yüzedilikli şekilde grafik halinde gösterimi yukarıda bulunan kodlarla yapılır. Çıktı olarak ise aşağıdaki grafiktedir.



Tablo 9: Duygu Analizine Göre Pozitif, Negatif ve Nötr Kelimelerin Yüzde Grafiği

Yukarıda bulunan grafik incelendiğinde kitap hakkında olan düşüncelerin daha çok nötr olduğunu görüyoruz. %91 üzerinde nötr, %4 pozitif ve %4 negatif olduğunu görebiliriz. Türkiye genelinde kitap okuma ve kitap hakkında fazla düşünceye sahip olunmadığını bu analiz ile de kanıtlayabiliriz.

Duygu skorlarının oransal olarak gösterimi:

```
head((positive/toplam)*100,"Pozitif")
head((neutral/toplam)*100 ,"Nötr")
head((negative/toplam)*100 ,"Negatif")
```

```
> head((positive/toplam)*100,"Pozitif")
[1] 4.328767
> head((neutral/toplam)*100 ,"Nötr")
[1] 91.05479
> head((negative/toplam)*100 ,"Negatif")
[1] 4.616438
>
```

Yüzdelik değerleri bu şekilde.

SONUÇ

Bu analizde insanların kitaplar ile ilgili tepkileri ölçülmüştür. Ve sonuç olarak Türkiye de bulunan insanların kitaplara karşı Nötr bir tutum sergilediğini görmekteyiz.
Türkiye'nin 2021 yılı verilerine göre okuma oranının 180 ülke orasında 140. Sırada olduğu açıklandı. Türkiye'de kitap okuma oranı gün geçtikte azalırken, en çok okunan kitaplar ise satılan kitaplar üzerinden değerlendiriliyor. %65 oranında aşk kitapları, %24 siyaset ve %13 oranında düşünce/ felsefe kitapları okunuyor. Türkiye de kitap okuma oranın düşük olmasından kaynaklı bu sonucun doğruluğunu da teyit etmemiz mümkündür. Ve genel olarak popüler romanların okunduğu tespitine ulaşıyoruz. Kitap okumayı seven kesimin ise çekiliş yardımıyla ve kendi okuma gruplarıyla okuduğu kitaplar hakkında tweetler yazdığını ve popüler olan kitapları ayrıca popüler olan kitaplarının yazarları hakkında tweetler atıldığını da görmekteyiz. Gündem her konuda olduğu kitap konusunda da etkili olmaktadır.

EKLER

1.Projede Kullanılan Kodlar

1.1İçerik Analizi İçin Kullanılan Kodlar

```
install.packages("twitteR")

install.packages("ROAuth)

install.packages("openssl")

install.packages("httpuv")

install.packages("tm")

install.packages("readxl")

install.packages("tidytext")

install.packages("wordcloud")

install.packages("writexl")

library(twitteR)

library(ROAuth)

library(openssl)

library(httpuv)

library(stringi)

library(stringr)

library(tm)

library(readxl)

library(tidytext)

library(dplyr)

library(ggplot2)

library(wordcloud)

library("writexl")

options(httr_oauth_cache=T)

consumer_key <- "------------------"

consumer_secret <- "------------------"

access_token <- "--------------------"

access_secret <- "--------------------"

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

```
tweets <- searchTwitteR("#kitap", n=10000,  locale = "tr_TR")
tweets.df <- twListToDF(tweets)
tweet_clean <- tweets.df
tweet_clean$text <- stri_enc_toutf8(tweet_clean$text)
tweet_clean$text <- ifelse(str_sub(tweet_clean$text,1,2) == "RT",
                           substring(tweet_clean$text,3),
                           tweet_clean$text)
tweet_clean$text <- str_replace_all(tweet_clean$text, "http[^[:space:]]*", "")
tweet_clean$text <- str_replace_all(tweet_clean$text, "#\\S+", "")
tweet_clean$text <- str_replace_all(tweet_clean$text, "@\\S+", "")
tweet_clean$text <- str_replace_all(tweet_clean$text, "[[:punct:][:blank:]]+", " ")
tweet_clean$text  <- str_to_lower(tweet_clean$text, "tr")
tweet_clean$text <- removeNumbers(tweet_clean$text)
tweet_clean$text <- str_replace_all(tweet_clean$text, "[<].*[>]", "")
tweet_clean$text <- gsub("\uFFFD", "", tweet_clean$text, fixed =  TRUE)
tweet_clean$text <- gsub("\n", "", tweet_clean$text, fixed =  TRUE)
tweet_clean$text <- str_replace_all(tweet_clean$text, "[^[:alnum:]]"," " )
Turkish_Stopwords <- read.csv("Turkish-Stopwords.csv")
head(Turkish_Stopwords)
tidy_tweets <- tweet_clean %>% select(text) %>%
  mutate(linenumber = row_number()) %>% unnest_tokens(word, text)
tidy_tweets <- tidy_tweets %>% anti_join(Turkish_Stopwords, by=c("word"="STOPWORD"))
head(tidy_tweets)
tidy_tweets %>%
  count(word, sort = TRUE) %>%
  filter(n > 150) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() + theme_minimal() +
```

```
  ggtitle("Tweetlerde en çok kullanılan kelimeler")
tidy_tweets %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
write_xlsx(tweet_clean, "C:/Yeni klasör/temizveri_13ocak.xlsx")
temizveri =read.table(file.choose(), header = T, sep = ";")
temiz_twit <- temizveri %>% select(text) %>%
  mutate(linenumber = row_number()) %>% unnest_tokens(word, text)
temiz_twit <- temiz_twit %>% anti_join(Turkish_Stopwords, by=c("word"="STOPWORD"))
head(temiz_twit)
temiz_twit %>%
  count(word, sort = TRUE) %>%
  filter(n > 250) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() + theme_minimal() +
  ggtitle("Tweetlerde en çok kullanılan kelimeler")
temiz_twit %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

1.2 Duygu Analizinde Kullanılan Kodlar

```
install.packages("twitteR")
install.packages("ROAuth")
install.packages("ROAuth")
install.packages("hms")
install.packages("lubridate")
install.packages("tidytext")
install.packages("tm")
install.packages("wordcloud")
```

```
install.packages("igraph")
install.packages("glue")
install.packages("networkD3")
install.packages("rtweet")
install.packages("plyr")
install.packages("stringr")
install.packages("networkD3")
install.packages("ggplot2")
install.packages("ggeasy")
install.packages("plotly")
install.packages("dplyr")
install.packages("hms")
install.packages("magrittr")
install.packages("tidyverse")
install.packages("janeaustenr")
install.packages("widyr")
library(twitteR)
library(ROAuth)
library(hms)
library(lubridate)
library(tidytext)
library(tm)
library(wordcloud)
library(igraph)
library(glue)
library(networkD3)
library(rtweet)
library(plyr)
library(stringr)
library(ggplot2)
library(ggeasy)
```

```
library(plotly)
library(dplyr)
library(hms)
library(magrittr)
library(tidyverse)
library(janeaustenr)
library(widyr)
options(httr_oauth_cache=T)
api_key <-
api_key_secret <-
access_token <-
access_token_secret <-

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
tweets <- searchTwitteR("#paribu", n=10000,  locale = "tr_TR")
n.tweet <- length(tweets)

tweets.df <- twListToDF(tweets)
tweets.txt <- sapply(tweets, function(t)t$getText())
tweets.txt <- str_replace_all(tweets.txt,"[^[:graph:]]", " ")
clean.text = function(x)
{
  x = tolower(x)
  x = gsub("rt", "", x)
  x = gsub("@\\w+", "", x)
  x = gsub("[[:punct:]]", "", x)
  # sayıların kaldırılması
  x = gsub("[[:digit:]]", "", x)
  x = gsub("http\\w+", "", x)
  x = gsub("[ |\t]{2,}", "", x)
  x = gsub("^ ", "", x)
```

```
  x = gsub(" $", "", x)
  x = gsub('https://','',x)
  x = gsub('http://','',x)
  x = gsub('[^[:graph:]]', ' ',x)
  x = gsub('[[:punct:]]', '', x)
  x = gsub('[[:cntrl:]]', '', x)
  x = gsub('\\d+', '', x)
  x = str_replace_all(x,"[^[:graph:]]", " ")
  return(x)
}
cleanText <- clean.text(tweets.txt)
idx <- which(cleanText == " ")
cleanText <- cleanText[cleanText != " "]
tweets.df %<>%
  mutate(
    created = created %>%
      # Sıfırları kaldırın.
      str_remove_all(pattern = '\\+0000') %>%
      #Ayrıştırma tarihi..
      parse_date_time(orders = '%y-%m-%d %H%M%S')
  )
tweets.df %<>%
  mutate(Created_At_Round = created%>% round(units = 'hours') %>% as.POSIXct())
tweets.df %>% pull(created) %>% min()
tweets.df %>% pull(created) %>% max()
plt <- tweets.df %>%
  dplyr::count(Created_At_Round) %>%
  ggplot(mapping = aes(x = Created_At_Round, y = n)) +
  theme_light() +
  geom_line() +
  xlab(label = 'Tarih') +
```

```
  ylab(label = NULL) +
  ggtitle(label = 'Saat Başına Tweet Sayısı')
plt %>% ggplotly()
positive <- read_csv("positive-words.csv")
negative <- read_csv("negative-words.csv")
pos.words =
c(positive,'güzel','iyi','mükemmel','beğendim','olmuş','özgün','doğru','etkin','hatasız','etkiliyeci','büyül
eyici','sürükleyici','tatlı','sevimli','heycanlı')
neg.words = c(negative,'Bearish','berbat', 'yanlış' ,'az', 'beğenmedim' ,'sat' ,'alçak', 'destek', 'güvensiz'
,'olumsuz' ,
          'alınmaz' ,"düştü",'zarar','beğenmedim','kötü','iyi
değil','delist','olmaz','sabırsız','yapmaz','düşüş','pahalı','uzun','sıkıcı'
          ,'dondu')score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  require(plyr)
  require(stringr)
  scores = laply(sentences, function(sentence, pos.words, neg.words) {
    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:]]", " ")
    sentence = tolower(sentence)
    word.list = str_split(sentence, '\\s+')
    words = unlist(word.list)

    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)
```

```
    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)
}
analysis <- score.sentiment(cleanText, pos.words, neg.words)
table(analysis$score)
analysis %>%
  ggplot(aes(x=score)) +
  geom_histogram(binwidth = 1, fill = "lightblue")+
  ylab("Frekans") +
  xlab("Duygu Skoru") +
  ggtitle("Tweetlerin Duyarlılık Puanlarının Dağılımı") +
  ggeasy::easy_center_title()
neutral <- length(which(analysis$score == 0))
positive <- length(which(analysis$score > 0))
negative <- length(which(analysis$score < 0))
toplam=positive+neutral+negative
Sentiment <- c("Pozitif","Nötr","Negatif")
Count <- c((positive/toplam)*100,(neutral/toplam)*100,(negative/toplam)*100)
output <- data.frame(Sentiment,Count)
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
ggplot(output, aes(x=Sentiment,y=Count,))+
  geom_bar(stat = "identity", aes(fill = Sentiment ))+
  ggtitle("Duygu Analizine Göre Pozitif, Negatif ve Nötr Kelimelerin Yüzde Grafiği")
head((positive/toplam)*100,"Pozitif")
head((neutral/toplam)*100 ,"Nötr")
head((negative/toplam)*100 ,"Negatif")
```

KAYNAKÇA

Twitter Text Mining Erişim: htttps://www.veribilimiokulu.com/twitter-text-mining/

memekanseri-with-cover-page-v2.pdf

Temizhan, Ebru, Mendeş, Mehmet, ,Çanakkale Onsekiz Mart Üniversitesi, Ziraat Fakültesi, Zootekni Bölümü Biyometri ve Genetik ABD, Çanakkale, TÜRKİYE, 2021;13(2):185-200

R ile metin madenciliği (2018-08-11) Erişim: https://www.veribilimiokulu.com/r-ile-metin-madenciligi-bolum-1/

Twitter İçerik Analizi (2018-03-31) Erişim: https://leventcan.github.io/blog/twitter-ile-i%C3%A7erik-analizi/

R'da Duygu Analizi Üzerine Vaka Çalışmaları: Case Studies on Sentiment Analysis in R (2021-02-27) Erişim: https://tevfikbulut.com/2021/02/27/rda-duygu-analizi-uzerine-vaka-calismalari-case-studies-on-sentiment-analysis-in-r/

2021 yılımda okunan kitaplar ve yazarlar açıklandı (2021-10-23) Erişim: http://onedio.com/haber/2021-yilinda-en-cok-okunan-kitaplar-ve-yazarlar-aciklandi-okuma-oraniyla-turkiye-180-ulke-arasindan-140-inci-1011368#:~:text=T%C3%BCrkiye'nin%202021%20y%C4%B1l%C4%B1%20verilerine,oran%C4%B1nda%20d%C3%BC%C5%9F%C3%BCnce%2F%20felsefe%20kitaplar%C4%B1%20okunuyor.