T.C.

Bilecik Şeyh Edebali Üniversitesi

İktisadi ve İdari Bilimler Fakültesi

Yönetim Bilişim Sistemleri

Twitter Veri Analizi – Duygu Analizi – Metin Analizi Uygulaması

Hazırlayan

Sefanur Pınar – 17567766522

BİLECİK, 2021

İÇİNDEKİLER

İÇİNDEKİLER 2
ÖNSÖZ 3
ÖZET 4
1.GİRİŞ 5
1.1 Veri Analizi – Veri Bilimi Nedir? 5
1.2 R Programlama Dili Nedir? 5
2. R PROGRAMLAMA DA METİN ANALİZİ (TEXT MINING) 6
2.1 Twitter ile Veri Analizi 6
2.2 Twitter Veri Analizi için kullanılan Paketler (twitteR) 7
3.R'DA METİN ANALİZİ UYGULAMASI 7
3.1 Veriyi Temizleme 10
3.2 Analiz: 11

ÖNSÖZ

Üniversite hayatım boyunca birçok proje üzerinde çalıştım ve kendimi geliştirdim. 4. Sınıfta aldığımız VERİ MADENCİLİĞİ dersi ile birlikte böyle kapsamlı bir projeyi hazırlıyorum. Eminim ki bu projenin iş hayatımda birçok yerde işe yarayacağını ve kendimi bu alanda geliştirmeme yardımcı olduğunu biliyorum. Bu projede emeği geçen Sayın Nur Kuban Torun hocamıza destekleri ve emekleri için teşekkürlerimi sunuyorum.

ÖZET

Günümüzde teknoloji şirketleri ve kurumlar büyük veriler üzerine çalışmaktadır. Büyük bir veri yığınından yararlı bilgiyi çekip çıkarabilmek ise oldukça zahmetli bir iştir. Madencilik sonucunda edinilen kazanımları göz önünde bulundurursak şirketler için sadece sahip oldukları verileri değil dışarıdan alınan verileri de koruyabilmek ve işleyebilmek son derece hassas bir konu haline gelmiştir. Çalışmam da "R" programlama dili ile birlikte Twitter'dan aldığım verileri metin analizi, duygu analizi yapacağım. Çalışmam da Türkiye Cumhurbaşkanlığı twitter resmi hesabı kullanılmıştır. 1000 'tweet' R programlama ile çekilip analizler yapılmıştır.

1.GİRİŞ

1.1 Veri Analizi – Veri Bilimi Nedir?

Kurumlardaki büyük ölçekli olarak tanımlanan ve milyonlarca veriye sahip yazılım sistemlerinden, ihtiyacı karşılayacak değerli verilerin elde edilmesi işlemine veri madenciliği denir. Bu sayede veriler arasındaki ilişkileri ortaya koymak ve gerektiğinde ileriye yönelik doğru tahminlerde bulunmak mümkün hale gelmektedir. Veri madenciliği'nde milyarlarca veri üzerinde çalışılabilir. Madenciliğin temel amacının kurumlardaki karar destek mekanizmaları olarak adlandırılan sistemler için değerli olan veriyi belirli yöntemler ve işlem süreçleri sonrası ortaya çıkarmaktır.

Veri analizi, doğru verilerle ve yöntemlerle yapıldığında, firmaların stratejik ve kritik kararlarında yapılabilecek birçok hatanın önüne geçilmesini sağlayabilmektedir. Bankacılık, finans, perakende, sağlık gibi birçok sektör veri analizlerini müşteri memnuniyetini ölçmek ve artırmak amacıyla kullanılmaktadır.

Veri analizi, Veri madenciliği ve Business Intelligence (BI) temel bir bileşenidir ve işletme kararlarını yönlendiren iç görü kazanmada anahtar rol oynar. Kuruluşlar, büyük veri yöntemi çözümlerini ve verileri işlemeye uygun iç görülere dönüştürmek için veri analizini kullanan müşteri deneyimi yönetimi çözümlerini kullanarak çok sayıda kaynaktan gelen verileri analiz eder ve kullanıcıya sunar.

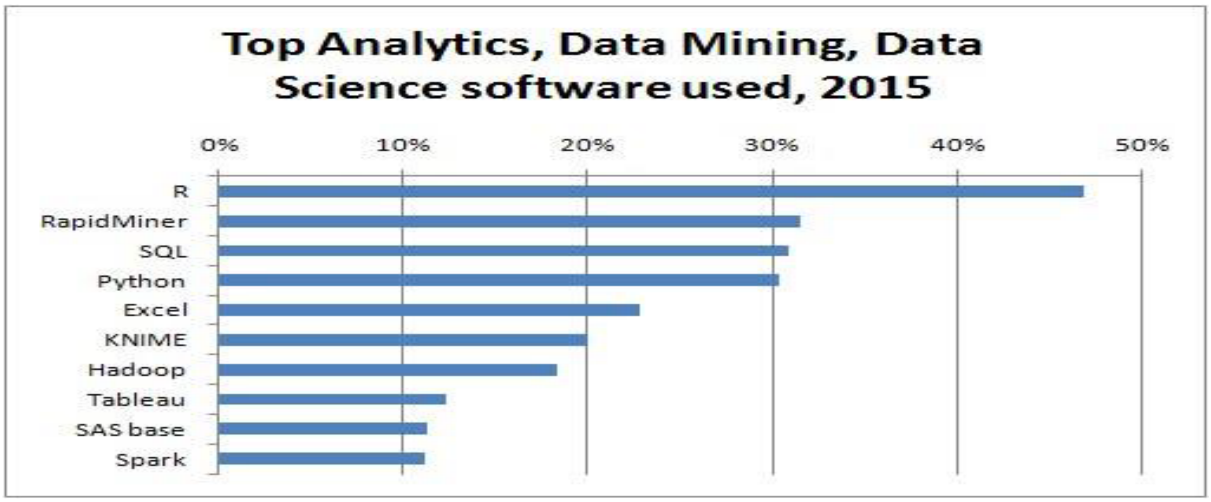
Veri Analizi Süreçleri;

- Problem Tanımlama
- Veri Anlama
- Veri Hazırlama
- Modelleme
- Değerlendirme
- Yayılım

1.2 R Programlama Dili Nedir?

R programlama dilini kısaca anlatmak gerekirse; istatiksel hesaplama, veri analizi ve bilimsel araştırmalarda verileri temizlemek, analiz etmek, görselleştirmek ve anlamlı hale getirmek için istatistikçiler, veri bilimciler, veri analistleri, araştırmacılar ve pazarlamacılar tarafından yaygın olarak kullanılan bir programlama dilidir. Dünya çapında birçok analist ve veri bilimci, kurumlar için temel bir araç haline gelen R programlama dilini finanstan tutun üretim, e-ticaret, sağlık, banka, kapsamlı pazarlamaya kadar uzanan alanlardaki en zor sorunlarıyla ilgilenmek için kullanılmaktadır ve son zamanlarda güncellenen kütüphaneler ile birlikte birçok veri işleme alanında bu dil kullanılmakta ve adresleme yapılabilmektedir. Örnek vermek gerekirse Twitter, Facebook, Amazon, Mozilla, Microsoft, Google, Bank Of America, Merck, Ford Motor Company, National Weather Service gibi birçok büyük kuruluşlar aktif olarak kullanmaktadır.

2015 yılında yapılmış bir ankete göre, analitik araçlar, veri madenciliği ve veri bilimi yazılım kullanıcılarının en çok tercih ettiği program R olduğu gözükmektedir.



2. R PROGRAMLAMA DA METİN ANALİZİ (TEXT MINING)

Metin analizi, Veri Madenciliği olarak da adlandırılır. Veri tabanları veya veri madenciliği araçlarını kullanarak büyük veri kümelerinin içinde bir model, desen veya anlamlı ilişki keşfetmek için kullanılan veri analizi yöntemlerinden biridir. Ham verilerden yararlı bilgiler elde etmek ve dönüştürmek için kullanılır. Genel olarak, verileri ortaya çıkartmak, incelemek, desenler türetmek ve verilerin yorumlanması için bir yol sunar.

Niteliksel veriler, rakamlarla ölçülemez nitelikte tanımlayıcı verilerdir. Genellikle renk, doku ve yazılı açıklama gibi görünür özellikleri içerir. Bilindiği gibi kantitatif veriler yapılandırılmış verilerdir. Bununla birlikte nitel ve nicel veriler arasında bir kayma söz konusu olur. Niteliksel verilere her gün kullanmaya alışık olduğumuz e-postalar, mailler, gazete ve web makaleleri, sosyal medya, telefon görüşmelerinin transkriptleri, blog yazıları veya diğer veriler olmak üzere örnek verebiliriz. Web ve sosyal medya aracılığıyla günde 8 milyondan fazla web sayfası metni deponuza günlük olarak eklenebilir düzeydedir.

2.1 Twitter ile Veri Analizi

Twitter, kullanıcılara okunması için kısa mesajlar (“tweet”) yayınlamasına imkan veren popüler bir hizmettir. Günümüzde bir çok sosyal medya kullanıcısının twitter hesaplarını aktif olarak kullandıkları gözlenmektedir. 2020 verilerine göre günde 850 milyondan fazla tweet atılmakla birlikte ayda 500 milyondan fazla aktif kullanıcı vardır. Bu sosyal medya platformu yıllar içinde yalnızca standart sosyal medya kullanım amacına değil, duyarlılık analizi gibi çeşitli veri madenciliği çalışmalarına da katkı sağlayan değerli bir araç haline gelmiştir. Sosyal araştırmalar için klasik anket yönetimi kullanmasından ziyade, interaktif bir ortamdan direkt olarak istenilen veriye ulaşılabiliyor olmak çok daha avantajlı ve pratik olarak karşımıza gelir. Gerek iş gücü, gerek ulaşılan örneklem ve elde edilen bilginin güncelliği açısından verimli bulunduğu için veri madenciliğinde kullanılan bir ortam haline yıllar içinde yer almıştır. Bilgi kirliliğini önlemek içinde çalışmalar güncel olarak twitter bünyesinde yapılmaya devam ediliyor.

2.2 Twitter Veri Analizi için kullanılan Paketler (twitteR)

Paketin amacı çeşitli analizler yapılabilmesi adına Twitter verisinin çeşitli alt gruplarını almasını sağlayan Twitter API'sine yani uygulama programlama arayüzüne, R programı ile erişimini sağlar. Böylelikle elde edilen metinler üzerinden analiz yapma imkanına erişilir.

3.R'DA METİN ANALİZİ UYGULAMASI

Twitter web sitesinde Türkiye Cumhuriyeti Cumhurbaşkanlığı hesabı ile ilgili insanlar tarafından yazılmış cümleler baz alınarak çeşitli analizler yapılacaktır.

Bu işleme başlamadan önce ilk olarak Twitter Developer Sayfası başlığı altında olan siteden API geliştirici hesabı almanız lazım. Bu hesap olmadan metin analizi yapamazsınız.

Öncelikle Twitter developer hesabımıza giriş yapıyoruz ve ardından uygulamalarım bölümünden yeni bir uygulama oluşturuyoruz. Bu uygulamaya bir ad veriyoruz ve bu uygulamayı ne için kullanacağımızı belirtiyoruz. Ardından Twitter API tarafından bizlere "TOKEN" adresleri veriliyor ve böylelikle bu tokenleri R programlama da kullanarak Twitter ile r programlama arasında ki bağlantıyı gerçekleştiriyoruz.

Bu uygulamayı oluşturmanın amacı Twitter'dan veri alabilme yetkisine sahip olmamızdır. Bu hesabı açtıktan sonra Twitter tarafından bizlere kullanmamız için, **Consumer Key, Consumer Secret Key, Access Token, Access Secret Token** olmak üzere 4 adet random key veriyor.

Daha sonrasında twitter'dan yetki işlemlerini aldıktan sonra R Studio programında yapılacak işlemlere geçiyoruz. Öncelikle Twitter'dan alacağımız veriler için belirli paketler indirmemiz gerekecek. Bu paketleri **"install.packages()"** ile indirebiliriz. **"Library()"** fonksiyonu ile indirdiğimiz paketleri aktif hale getirip kullanıma hazırlayabiliriz.

- **Resimde göründüğü gibi ilk olarak veri analizi için paketlerimizi yükledik.**

```
# İLK ÖNCELİKE VERİ ANALİZİ YAPMAMIZ İÇİN İNDİRMEMİZ GEREKEN PAKETLERİ İNDİRİP VE TANIYALIM.

install.packages("ROAuth")        #Twitter'daki uygulamaya giriş yapmak ve iletişim kurmak için kullanırız.
install.packages("twitteR")       #Twiiter'dan veri almak için kullanırız.
install.packages("tm")            #Metin analizi için kullanırız.
install.packages("wordcloud")     #Kelime bulutu hazırlamak için kullanırız.
install.packages("ggplot2")       #Oluşturacağımız grafikleri görüntülemek için kullanırız.
install.packages("RColorBrewed")  #Oluşturulacak renk paletleri için kullanırız.
install.packages("stringr")       # 'string' verilere yani metinsel verilere manipülasyon için kullanırız.
install.packages("plyr")          #Ortak sorun kümeleri için (split-apply-combine işlemleri için kullanırız.)

# YÜKLEDİĞİMİZ KÜTÜPHANELERİ AKTİFLEŞTİRELİM.

library(ROAuth)
library(twitteR)
library(tm)
library(wordcloud)
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(plyr)
```

“ROAuth” paketindeki setup_twitter_oauth() fonksiyonunu kullanarak daha Twitter API hesabından aldığımız 4 adet TOKEN Şifremizi girerek Twitter arasında bağlantı sağlarız. “Using direct authentication“ mesajını aldıktan sonra işlemimiz başarılı bir şekilde gerçekleşmiştir.

```
# "ROAuth" paketindeki setup_twitter_oauth() fonksiyonu kullanılarak twitter
# API hesabımızla iletişimi sağlıyoruz.
# ilk öncelikle API'Lerimizi giriyoruz.

api_key <- [redacted]
api_secret_key <- [redacted]
access_token <- [redacted]
access_token_secret <- [redacted]

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

Bağlantı işlemi başarılı olduktan sonra “Using direct authentication” bildirimini alıyoruz.

```
[1] "Using direct authentication"
>
```

“TwitteR” paketinde searchTwitter() fonksiyonu ile öncelikle veri olarak Türkiye Cumhuriyeti sayfasını ele aldık ve burada atılan yorumları, tweetlerin 1000 adet çektik ve listeledik.

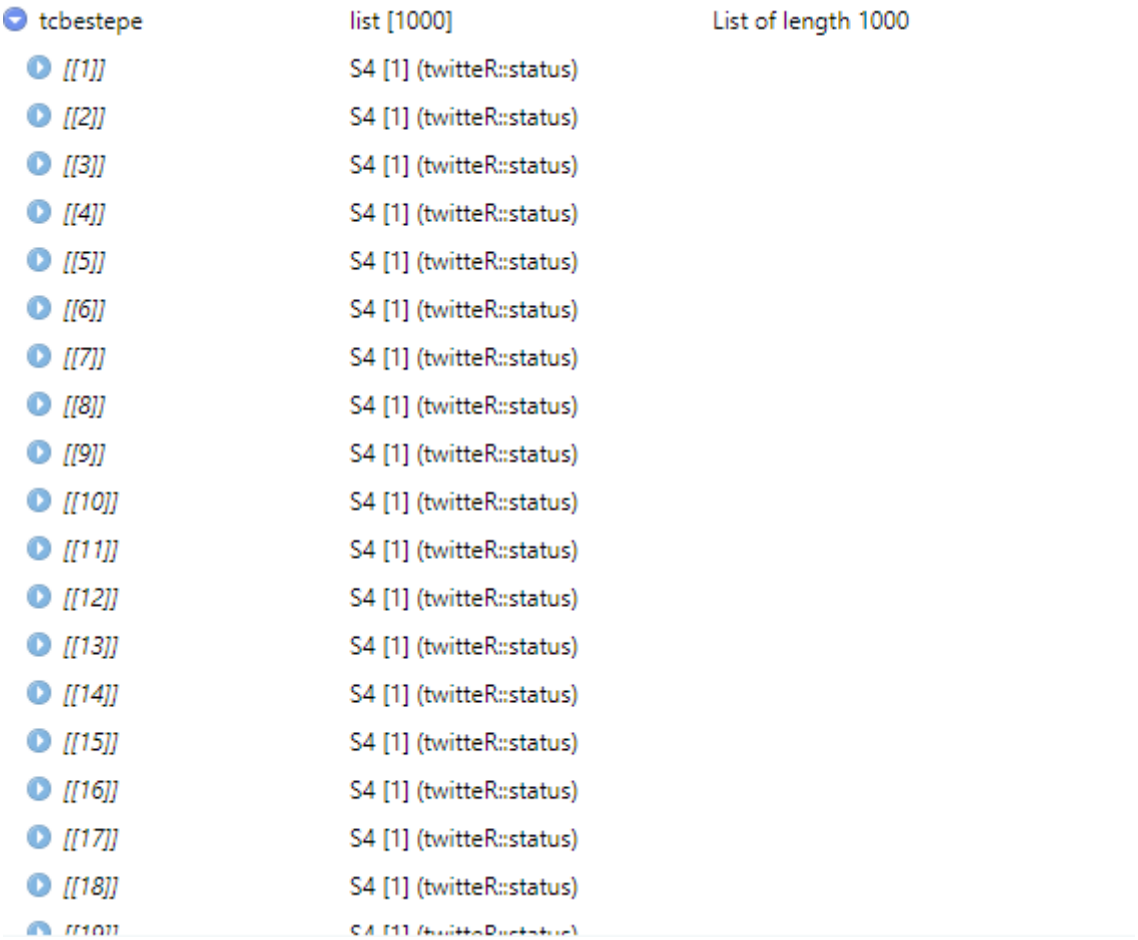
```
tcbestepe <- searchTwitteR('tcbestepe', n = 1000, lang = "tr") # verileri çekeriz.
save(tcbestepe, file="tc.RData") # verileri kaydederiz.

length(tcbestepe)
```

Listelendi.

```
> tcbestepe <- searchTwitteR('tcbestepe', n = 1000, lang = "tr") # verileri çekeriz.
> save(tcbestepe, file="tc.RData") # verileri kaydederiz.
> length(tcbestepe)
[1] 1000
>
```

Resimde göründüğü gibi 1000'e kadar devam ediyor.



tcbestepe	list [1000]	List of length 1000
[[1]]	S4 [1] (twitteR::status)	
[[2]]	S4 [1] (twitteR::status)	
[[3]]	S4 [1] (twitteR::status)	
[[4]]	S4 [1] (twitteR::status)	
[[5]]	S4 [1] (twitteR::status)	
[[6]]	S4 [1] (twitteR::status)	
[[7]]	S4 [1] (twitteR::status)	
[[8]]	S4 [1] (twitteR::status)	
[[9]]	S4 [1] (twitteR::status)	
[[10]]	S4 [1] (twitteR::status)	
[[11]]	S4 [1] (twitteR::status)	
[[12]]	S4 [1] (twitteR::status)	
[[13]]	S4 [1] (twitteR::status)	
[[14]]	S4 [1] (twitteR::status)	
[[15]]	S4 [1] (twitteR::status)	
[[16]]	S4 [1] (twitteR::status)	
[[17]]	S4 [1] (twitteR::status)	
[[18]]	S4 [1] (twitteR::status)	

"tcbestepe" isimli oluşturduğumuz vektöre atadığımız veriler, metin dışında birçok değişken içermektedir. Metin analizinde sadece metin içeren değerler kullanılacağı için **"sapply()"** fonksiyonu ile **"tcbestepe.txt"** isimli yeni bir vektör oluşturuyoruz ve **Corpus** adı verilen yapıyı oluşturmak için ise, **"tm"** paketinden yararlanıyoruz. **Corpus() ve VectorSource()** **"tm"** paketinden yararlanılarak yapıldı.

```
tcbestepe.txt <- sapply(tcbestepe, function(x) x$getText())

mycorpus <- corpus(VectorSource(tcbestepe.txt))
```

Bu kodun açıklamasını yapmak gerekirse, çektiğimiz verilerin .txt uzantılı dosyaya yazdırılmasıdır.

3.1 Veriyi Temizleme

Metin(Text) analizi için twitter'dan elde ettiğimiz veriler içerisinde metinler çeşitli sembollerle, büyüklü küçüklü harflerle, sayılarla ve özel karakterle doludur. İyi bir uygulama için bu metinlerin temizlenmesi ve analize hazır hali getirmemiz gerekecektir.

"tm" yani **"Text Mining"** paketini yükledik. Bu pakette temizleme sürecinde kullanacağımız etkili fonksiyonlar vardır. Bu veri çalışmasında elde ettiğimiz metinler bir sosyal medya web sitesinden alındığı için oldukça fazla web sitesi linki **(URL)** ve çeşitli semboller barındırdığından dolayı bizler de çektiğimiz verileri daha temiz sade bir şekilde ayrıt edeceğimizden dolayı bu kütüphaneyi kullanacağız.

```
removeURL <- function(x) gsub("http[[:alnum:]]*","",x) # Bu işlem ile çektiğimiz verilerde ki URL kısımlarını temizlemek için yapıyoruz.

removeSpace <- function(x) gsub("[^[:alpha:][:space:]*","",x) # Bu fonksiyonda ise boşlukları ve türkçe karekter dışındaki verileri sildik.
```

Çektiğimiz verilerimizi URL ve gereksiz boşluk ve karakterlerden kurtardık.

Metin analizinde önemli kelimelere ulaşabilmek için cümlelerde geçen en sık kullanılan öznelerin verilerden arındırılması gerekir. **"stopwords"** paketini kullanarak bulacağımız özneleri belirleyelim.

```
lists <- c(stopwords,"erdoğan","ekonomi","dolar","recep","tayyip","ankara","iyi","kötü")
lists
```

Tüm bu yaptığımız adımlardan sonra temizleme işlemlerini hızlandırabiliriz. R'da bahsedilen **"tm"** paketi içerisindeki **"tm_map()"** fonksiyonu temizleme işlemleri için oldukça kullanışlı ve işlevseldir. Bu işlemleri yaparken işlemlerin sırası çok önemlidir. Bu sıra metinden metine farklılık gösterebilir. Düzenli veya kurallı değildir.

Bazı fonksiyonlar kullanılarak kelime ve metin analizi üzerindeki değerler temizlendi ve daha sade hali aldı.

- **PlainTextDocument:** Sade metin belgesine dönüştürme işlemine yarar.
- **Content_transformer(removeURL):** URL dosyaları kaldırdık.
- **StripWhitespace:** Kelimeler aralarında boşluklar eşitlenir.
- **Content_transformer(tolower):** Tüm kelimeler küçük harfe dönüşür.
- **RemoveWords:** Gereksiz kelimeler kaldırıldı.
- **RemovePunctution:** Noktalama işaretleri ve sayılar kaldırıldı.

Aşağıda ki resimde göründüğü gibi temizleme işlemlerimizi yaptık.

```
clean_corp <- tm_map(myCorpus,PlainTextDocument) # Oluşturulan yapının içeriğini PlainTextDocument işlemi ile sade metin belgesine dönüştürdük.
#ardından yen clean_corp adı altında yeni bir yapıya dönüştürdük.
clean_corp <- tm_map(clean_corp, content_transformer(removeURL)) #URL Kaldırma işlemini gerçekleştirdik.
clean_corp <- tm_map(clean_corp, stripWhitespace)               #Kelimeler arasındaki boşlukları kaldırdık.
clean_corp <- tm_map(clean_corp, content_transformer(tolower))   #Tüm harfleri küçük harfe çevirdik.
clean_corp <- tm_map(clean_corp, removeWords, lists)            #ihtiyaç olmayan kelimeler, karakterler kaldırıldı.
clean_corp <- tm_map(clean_corp, content_transformer(removeSpace)) #Farklı semboller ve sayılar kaldırıldı.
clean_corp <- tm_map(clean_corp, removePunctuation)
```

3.2 Analiz:

Verilerin temizlenme işleminden sonraki süreç, terim – doküman matrisi oluşturma, sık kullanılan kelimeler, kelime bulutu(wordcloud) oluşturma ve ilişkilendirmeler olmak üzere 4 farklı süreç vardır.

1. **Terim – Doküman Matrisi Oluşturma**

Doküman terim matrisi veya belge matrisi bir gel topluluğunda oluşan terimlerin sıklığını tanımlayan matematiksel bir matristir. Bu matrislerin oluşturulma amacı terimleri ve aralarındaki ilişkileri incelemek ve görselleştirme işlemini kolaylaştırmak için yapılır. Çalışmamızda terim-doküman matrisi oluşturacağımız için **"tm"** paketi aracılığıyla **TermDocumentMatrix()** fonksiyonu kullanıldı.

```
tcbestepe_tdm <- TermDocumentMatrix(clean_corp)
tcbestepe_tdm

tcbestepe_m <- as.matrix(tcbestepe_tdm)
dim(tcbestepe_m)
```

```
> tcbestepe_tdm <- TermDocumentMatrix(clean_corp)
> tcbestepe_tdm
<<TermDocumentMatrix (terms: 1860, documents: 1000)>>
Non-/sparse entries: 12146/1847854
Sparsity           : 99%
Maximal term length: 48
Weighting          : term frequency (tf)
> tcbestepe_m <- as.matrix(tcbestepe_tdm)
> dim(tcbestepe_m)
[1] 1860 1000
>
```

Resimde göründüğü üzere tcbestepe_tdm isimli bir terim doküman matrisi oluşturduk. İçerisinde 1860 adet terim ve 1000 adet doküman olduğunu görüyoruz. Oluşturduğumuz bu matris de 1860 satır ve 1000 sütun olduğunu görmekteyiz.

2. Sık Kullanılan Kelimeler

İlk olarak term_frequency isimli bir liste oluşturuyoruz. Bu liste de tcbestepe_m matrisindeki toplam satır sayılarının yani kullanılan kelime sayılarının, azalan sıraya göre sıralıyoruz.

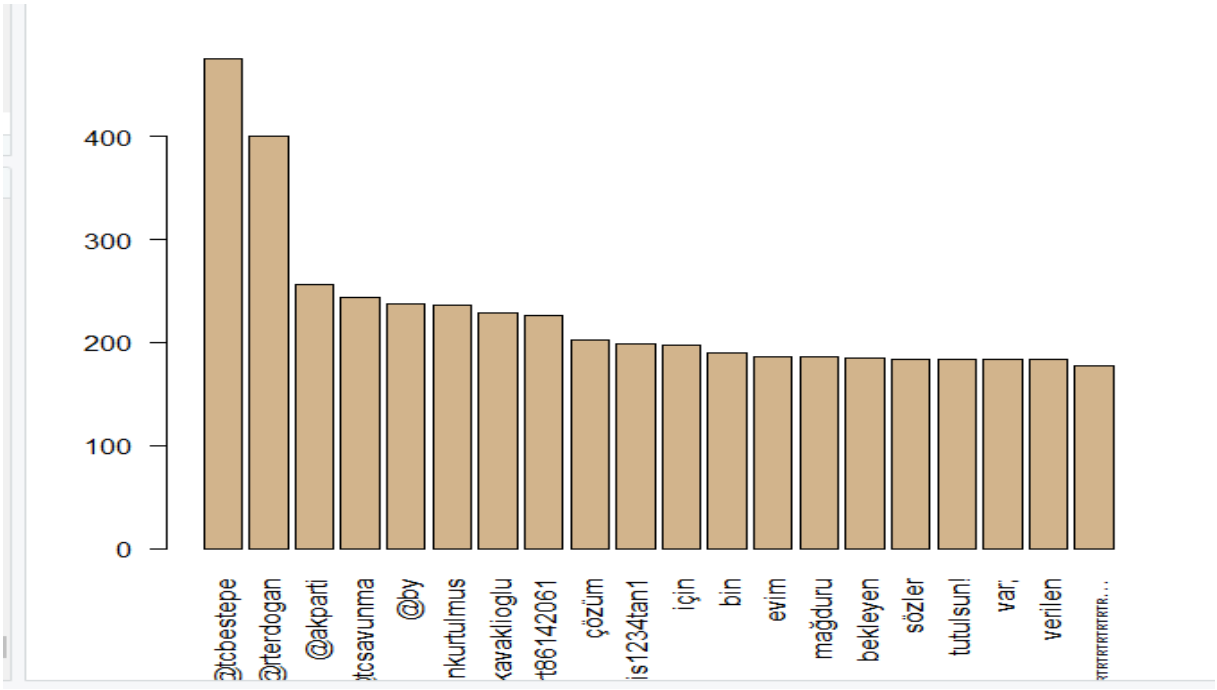
```
frequency_terms <- rowSums(tcbestepe_m)
frequency_terms <- sort(frequency_terms, decreasing = TRUE) #kullanılan verileri azalan sıraya göre sıraladık.
frequency_terms[1:10]
```

En çok tekrar edenler…

```
> frequency_terms <- rowSums(tcbestepe_m)
> frequency_terms <- sort(frequency_terms, decreasing = TRUE) #kullanılan verileri azalan sıraya göre sıraladık.
> frequency_terms[1:10]
     @tcbestepe      @rterdogan        @akparti       @tcsavunma             @by  @numankurtulmus @alp_kavaklioglu @bozkurt86142061           çözüm
            475             400             256             244             237             236             229             226             202
    @is1234tan1
            199
>
```

Bu listenin bar grafiğini oluşturalım.

```
barplot(frequency_terms[1:20], col="tan", las=2)
```

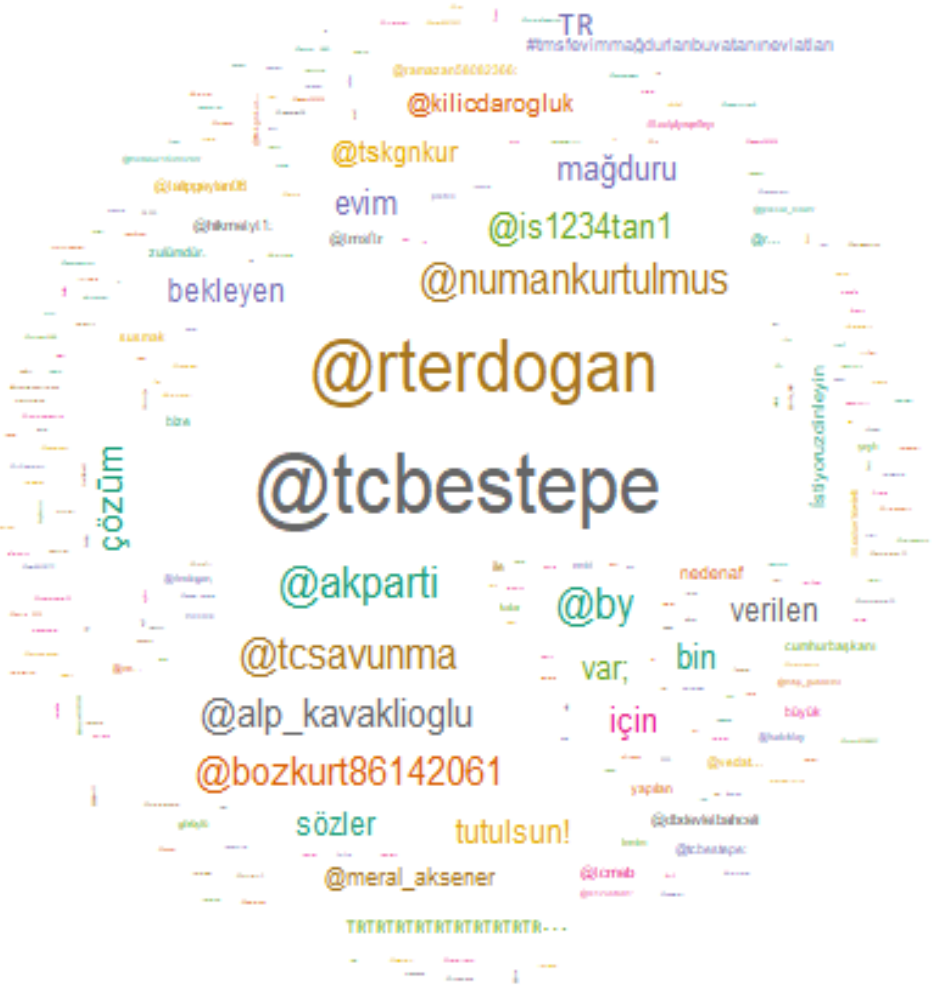


3. Kelime Bulutu (WordCloud) Oluşturma

Temizlenmiş yapı olan clean_corp üzerinden sık kullanılan kelimeleri görselleştirmek amacıyla kelime bulutu oluşturuyoruz. Bu işlem için **"wordcloud()"** paketini kullanıyoruz.

```
wordcloud(clean_corp,min.freq = 2, scale = c(2,0,5),colors = brewer.pal(8,"Dark2"),
random.color = TRUE, random.order = FALSE, max.words = 250)
```

En çok kullanılan resimler şekillerde daha çok değerlendirilmiştir ve daha büyük oranda görünüyor.



4. Duygu Analizi – Semantic Analysis

Oluşturduğumuz kelime dataları üzerinden duygu analizi yapacağız. Bu analizin amacı kelimelerin anlamlarına göre olumlu, nört, olumsuz gruplara ayrılmasını sağlamaktır. Biz Türkçede iyi ve kötü anlamlarına gelen kelime gruplarını bir txt.dosyasının içerisine koyacağız ve öyle devam edeceğiz. Hazırladığımız ve bilgisayarda hazır olan metin dosyalarını okutup pozitif kelimelerin pos değişkenine, negatif değişkenlerin ise neg değişkenine karakter olarak atayalım.

```
findAssocs(tcbestepe_tdm, terms = "erdogan", corlimit = 0.5)

pos <- scan('C:/Users/Sefa/Desktop/UZEM 4.SINIF DERSLER/Veri Madenciliği/DÖNEM ÖDEVİ/positive.txt', what='character')

neg <- scan('C:/Users/Sefa/Desktop/UZEM 4.SINIF DERSLER/Veri Madenciliği/DÖNEM ÖDEVİ/negative.txt', what='character')

list(pos[1:10])  # iyi ve kötü kelimeleri yükledik ve sistemimize aldık.
```

```
> pos <- scan('C:/Users/Sefa/Desktop/UZEM 4.SINIF DERSLER/Veri Madenciliği/DÖNEM ÖDEVİ/positive.txt', what='character')
Read 12 items
> neg <- scan('C:/Users/Sefa/Desktop/UZEM 4.SINIF DERSLER/Veri Madenciliği/DÖNEM ÖDEVİ/negative.txt', what='character')
Read 11 items
> list(pos[1:10])
[[1]]
 [1] "guzel"    "basarili" "gayet"    "iyi"      "mukemmel" "super"    "iyi"      "cok"      "begendim" "begendim"

>
```

Daha sonra kelimelerin duygu skorlarını belirleyen sentiment_scores isimli bir fonksiyon oluşturuyoruz.