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

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 **"Bill Gates"** Twitter resmi hesabı kullanılmıştır. Twitter'dan aldığımız API'key ile 21 Günlük **"Bill Gates"** verileri incelenmiştir.

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 madenciliğinde 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.

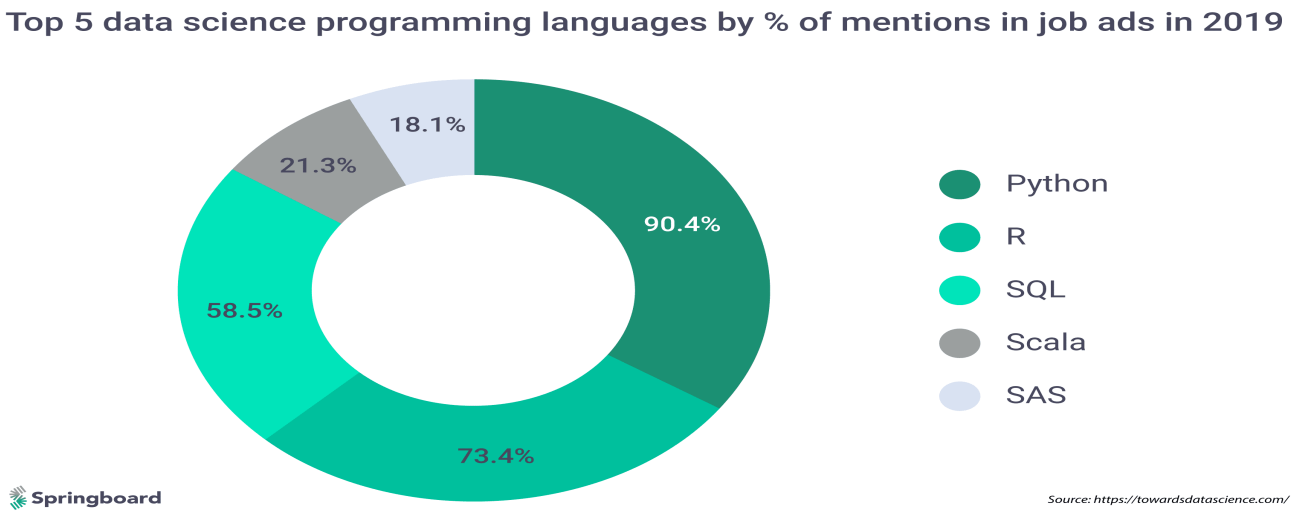
Veri Analizi Süreçleri;

- 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.

2019 yılında yapılmış bir ankete göre, analitik araçlar, veri madenciliği ve veri bilimi yazılım kullanıcılarının en çok tercih ettiği program **Python** ve **R** programlama 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 İçerik Analizi için kullanılan Paketler

(“twitteR”): Paketin amacı çeşitli analizler yapılabilmesi adına Twitter verisinin çeşitli alt gruplarını almasını sağlayan Twitter API’sine yani uygulama programlama ara yüzüne, R programı ile erişimini sağlar. Böylelikle elde edilen metinler üzerinden analiz yapma imkanına erişilir.

(“ROAuth”): Twitter hesabımıza R üzerinden erişebilmek için kullanılan bağlantı paketidir.

(“tm”): Metin analizi için kullanılan bir pakettir.

(“wordcloud”): Kelime havuzu veya kelime bulutu hazırlamak için kullanılan pakettir.

(“ggplot2”): Verilerin görselleştirilmesi için kullanılan bir pakettir.

(“RColorBrewed”): Yapılan çalışmalarımızda renk paletlerini oluşturmak için kullanılan pakettir.

(“stringr”): Karakterden oluşan verilerin manipülasyonu için kullanılır.

(“dplyr”): Veri manipülasyonu için kullanılır.

(“tidytext”): Düzenli metin araçlarını kullanabilmemizi sağlayan pakettir.

(“readr”): csv ya da **tsv** gibi dikdörtgen şeklindeki veri tablolarının okunması için kullanılır.

3.R’DA METİN ANALİZİ UYGULAMASI

Twitter web sitesinde **“Bill Gates”** resmi twitter 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")       #Twitter'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("dplyr")         #Ortak sorun kümeleri için (split-apply-combine işlemleri için kullanırız.)
install.packages("tidytext")      #Düzenli metin araçlarını barındırır.
install.packages("readr")         #csv ya da tsv gibi dikdörtgen şeklindeki veri tablolarının okunması için kullanılır.


# YÜKLEDİĞİMİZ KÜTÜPHANELERİ AKTİFLEŞTİRELİM.

library(ROAuth)
library(twitteR)
library(tm)
library(wordcloud)
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(dplyr)
library(tidytext)
library(readr)
```

"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"
>
```

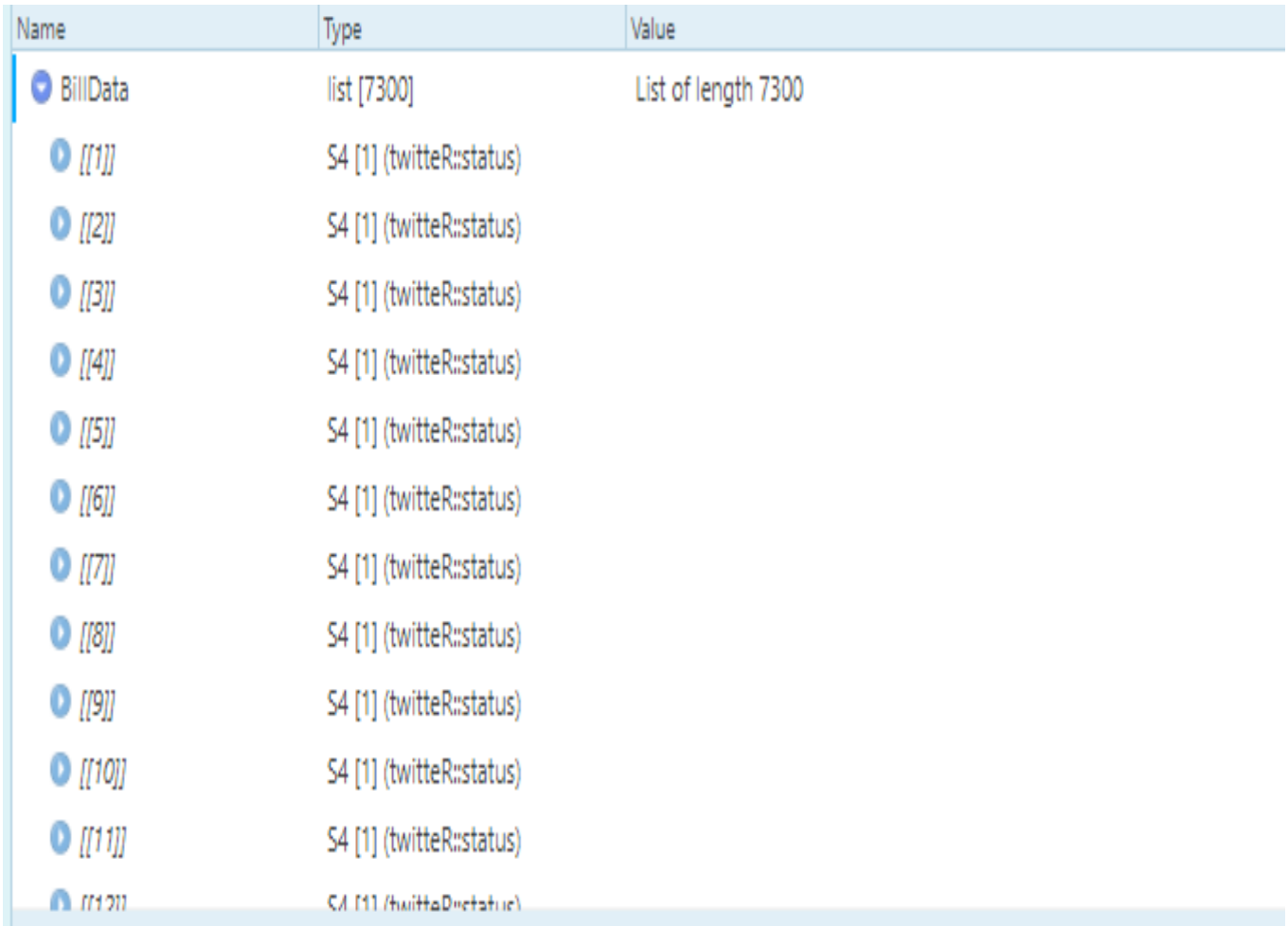
“TwitteR” paketinde searchTwitter() fonksiyonu ile öncelikle veri olarak “Bill Gates” resmi twitter sayfasını ele aldık ve burada atılan yorumları, tweetlerin 21 günlük yani tweet çektik ve listeledik.

```
# Twitter da, "twitteR" paketindeki searchTwitter() fonksiyonu ile
# "BillGates" hesabının 21 günlük verileri çekilmiştir ve incelemeleri yapılmıştır.

BillData <- searchTwitteR('#BillGates', n = 30000, lang="en")  # '#BillGates' adı altındaki verileri çekiyoruz.
```

Listelendi.

```
> save(BillData, file="BillData.RData") #Data Kaydedilir.
> length(BillData)
[1] 7300
>
```

Resimde göründüğü gibi 7300’e kadar devam ediyor.



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.

Çektiğimiz metinler oldukça düzensizdir ve metin dışında birçok karakter vardır. Bunu önlemek için **sapply()** fonksiyonu ile **BillData.text** isimli yeni bir vektör oluşturduk ve ardından yeni bir **Corpus** oluşturarak verilerimizi temizleme işlemine geçiyoruz

```
#Corpus Oluşturma İşlemi

BillData.text <- sapply(BillData, function(x) x$getText())
mycorpus <- Corpus(VectorSource(BillData.text))
```

- **RemoveURL :** URL temizleme işlemi için kullanılır.
- **RemoveNumPunct:** İngilizce olmayan harfleri ve boşluklar için kullanılır.
- **PlainTextDocument:** Sade metin belgesine dönüştürme işlemine yarar.
- **StripWhitespace:** Kelimeler aralarında boşluklar eşitlenir.
- **ToLower:** Tüm kelimeleri küçük harfe dönüştürür.
- **RemoveWords:** Gereksiz tekrar eden kelime grupları kaldırılır.
- **RemovePunctution:** Noktalama işaretleri ve sayılar kaldırılır.

Stopwords: Türkçede kullandığımız etkisi kelimeler grubunun listesini bilgisayarımızdan çekiyoruz ve projemize ekliyoruz.

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

Örnek vermek gerekirse: Acaba, Ama, Ancak, Aslında, Elbette, da, de, gibi, gene, yine, az gibi kelime gruplarını etkisiz olarak ele alarak yeni bir dosya oluşturuyoruz.

Temizlenmiş tweetlerimizi metin belirteçlerine bölerek hangi kelimenin kaç kere tekrar ettiğini görebileceğimiz fonksiyonu yazalım.

```
#Gereksiz yere tekrar eden verileri kaldırmak için kullanığımız
#kelimeleri ekliyoruz.

rmv_word <- c(stopwords("en"))
```

Stopwords Kelimeleri:

```
 [33] "should"     "could"     "ought"     "i'm"       "you're"    "he's"      "she's"     "it's"      "we're"
 [64] "they're"    "i've"      "you've"    "we've"     "they've"   "i'd"       "you'd"     "he'd"      "she'd"
 [73] "we'd"       "they'd"    "i'll"      "you'll"    "he'll"     "she'll"    "we'll"     "they'll"   "isn't"
 [82] "aren't"     "wasn't"    "weren't"   "hasn't"    "haven't"   "hadn't"    "doesn't"   "don't"     "didn't"
 [91] "won't"      "wouldn't"  "shan't"    "shouldn't" "can't"     "cannot"    "couldn't"  "mustn't"   "let's"
[100] "that's"     "who's"     "what's"    "here's"    "there's"   "when's"    "where's"   "why's"     "how's"
[109] "a"          "an"        "the"       "and"       "but"       "if"        "or"        "because"   "as"
[118] "until"      "while"     "of"        "at"        "by"        "for"       "with"      "about"     "against"
[127] "between"    "into"      "through"   "during"    "before"    "after"     "above"     "below"     "to"
[136] "from"       "up"        "down"      "in"        "out"       "on"        "off"       "over"      "under"
[145] "again"      "further"   "then"      "once"      "here"      "there"     "when"      "where"     "why"
[154] "how"        "all"       "any"       "both"      "each"      "few"       "more"      "most"      "other"
[163] "some"       "such"      "no"        "nor"       "not"       "only"      "own"       "same"      "so"
[172] "than"       "too"       "very"      "ve"        "veya"      "kim"       "https"     "http"      "de"
[181] "ama"        "amaç"      "a"         "acaba"     "altı"      "altmış"    "ama"       "ancak"     "arada"
[190] "artık"      "asla"      "aslında"   "aslında"   "ayrıca"    "az"        "bana"      "bazen"     "bazı"
[199] "bazıları"   "belki"     "ben"       "benden"    "beni"      "benim"     "beri"      "beş"       "bile"
[208] "bilhassa"   "bin"       "bir"       "biraz"     "birçoğu"   "birçok"    "biri"      "birisi"    "birkaç"
[217] "birşey"     "biz"       "bizden"    "bize"      "bizi"      "bizim"     "böyle"     "böylece"   "bu"
[226] "buna"       "bunda"     "bundan"    "bunlar"    "bunları"   "bunların"  "bunu"      "bunun"     "burada"
[235] "bütün"      "çoğu"      "çoğunu"    "çok"       "çünkü"     "da"        "daha"      "dahi"      "dan"
[244] "de"         "defa"      "değil"     "diğer"     "diğeri"    "diğerleri" "diye"      "doksan"    "dokuz"
[253] "dolayı"     "dolayısıyla" "dört"    "e"         "edecek"    "eden"      "ederek"    "edilecek"  "ediliyor"
[262] "edilmesi"   "ediyor"    "eğer"      "elbette"   "elli"      "en"        "etmesi"    "etti"      "ettiği"
[271] "ettiğini"   "fakat"     "falan"     "filan"     "gene"      "gereği"    "gerek"     "gibi"      "göre"
[280] "hala"       "halde"     "halen"     "hangi"     "hangisi"   "hani"      "hatta"     "hem"       "henüz"
[289] "hep"        "hepsi"     "her"       "herhangi"  "herkes"    "herkese"   "herkesi"   "herkesin"  "hiç"
[298] "hiçbir"     "hiçbiri"   "i"         "ı"         "için"      "içinde"    "iki"       "ile"       "ilgili"
[307] "ise"        "işte"      "itibaren"  "itibariyle" "kaç"      "kadar"     "karşın"    "kendi"     "kendilerine"
[316] "kendine"    "kendini"   "kendisi"   "kendisine" "kendisini" "kez"       "ki"        "kim"       "kime"
[325] "kimi"       "kimin"     "kimisi"    "kimse"     "kırk"      "madem"     "mi"        "mı"        "milyar"
[334] "milyon"     "mu"        "mü"        "nasıl"     "ne"        "neden"     "nedenle"   "nerde"     "nerede"
[343] "nereye"     "neyse"     "niçin"     "nin"       "nın"       "niye"      "nun"       "nün"       "o"
[352] "öbür"       "olan"      "olarak"    "oldu"      "olduğu"    "olduğunu"  "olduklarını" "olmadı"  "olmadığı"
[361] "olmak"      "olması"    "olmayan"   "olmaz"     "olsa"      "olsun"     "olup"      "olur"      "olur"
[370] "olursa"     "oluyor"    "on"        "ön"        "ona"       "önce"      "ondan"     "onlar"     "onlara"
[379] "onlardan"   "onları"    "onların"   "onu"       "onun"      "orada"     "öte"       "ötürü"     "otuz"
[388] "öyle"       "oysa"      "pek"       "rağmen"    "sana"      "sanki"     "sanki"     "şayet"     "şekilde"
[397] "sekiz"      "seksen"    "sen"       "senden"    "seni"      "senin"     "şey"       "şeyden"    "şeye"
[406] "şeyi"       "şeyler"    "şimdi"     "siz"       "siz"       "sizden"    "sizden"    "size"      "sizi"
[415] "sizi"       "sizin"     "sizin"     "sonra"     "şöyle"     "şu"        "şuna"      "şunları"   "şunu"
[424] "ta"         "tabii"     "tam"       "tamam"     "tamamen"   "tarafından" "trilyon"  "tüm"       "tümü"
[433] "u"          "ü"         "üç"        "un"        "ün"        "üzere"     "var"       "vardı"     "ve"
[442] "veya"       "ya"        "yani"      "yapacak"   "yapılan"   "yapılması" "yapıyor"   "yapmak"    "yaptı"
[451] "yaptığı"    "yaptığını" "yaptıkları" "ye"       "yedi"      "yerine"    "yetmiş"    "yi"        "yı"
```

İçerik Analizimiz için çektiğimiz tweetleri temizleme işlemlerini yapıyoruz.

Verileri temizleme işlemleri:

```
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)
```

Verilerimizin temizlenmemiş ve daha sonra temizlenmiş hali alt ki görseldedir. Temizlenmemiş halinin ne kadar karışık ve düzensiz olduğunu görebiliyoruz.

```
> BillData.text[10]
[1] "RT @AlexGeezy13: MUST WATCH #ArrestBillGates #BillGates https://t.co/8arNc1Dead"
> clean_corp[[10]][1]
$content
[1] " alexgeezy must watch arrestbillgates billgates tcoarncdead"

> ""
```

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.

```
#Terim Dökümantasyon Değerleri
BillData_tdm <- TermDocumentMatrix(clean_corp)
BillData_tdm
```

```
> #Terim Dökümantasyon Değerleri
> BillData_tdm <- TermDocumentMatrix(clean_corp)
> BillData_tdm
<<TermDocumentMatrix (terms: 7307, documents: 7314)>>
Non-/sparse entries: 67850/53375548
Sparsity           : 100%
Maximal term length: 45
Weighting          : term frequency (tf)
>
```

```
> #matrix oluşturma
> BillData_m <- as.matrix(BillData_tdm)
> dim(BillData_m)
[1] 7307 7314
```

Resimde göründüğü üzere BillData_tdm isimli bir terim doküman matrisi oluşturduk. İçerisinde 7307 adet terim ve 7314 adet doküman olduğunu görüyoruz. Oluşturduğumuz bu matris de 7307 satır ve 7314 sütun olduğunu görmekteyiz.

2. Sık Kullanılan Kelimeler

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

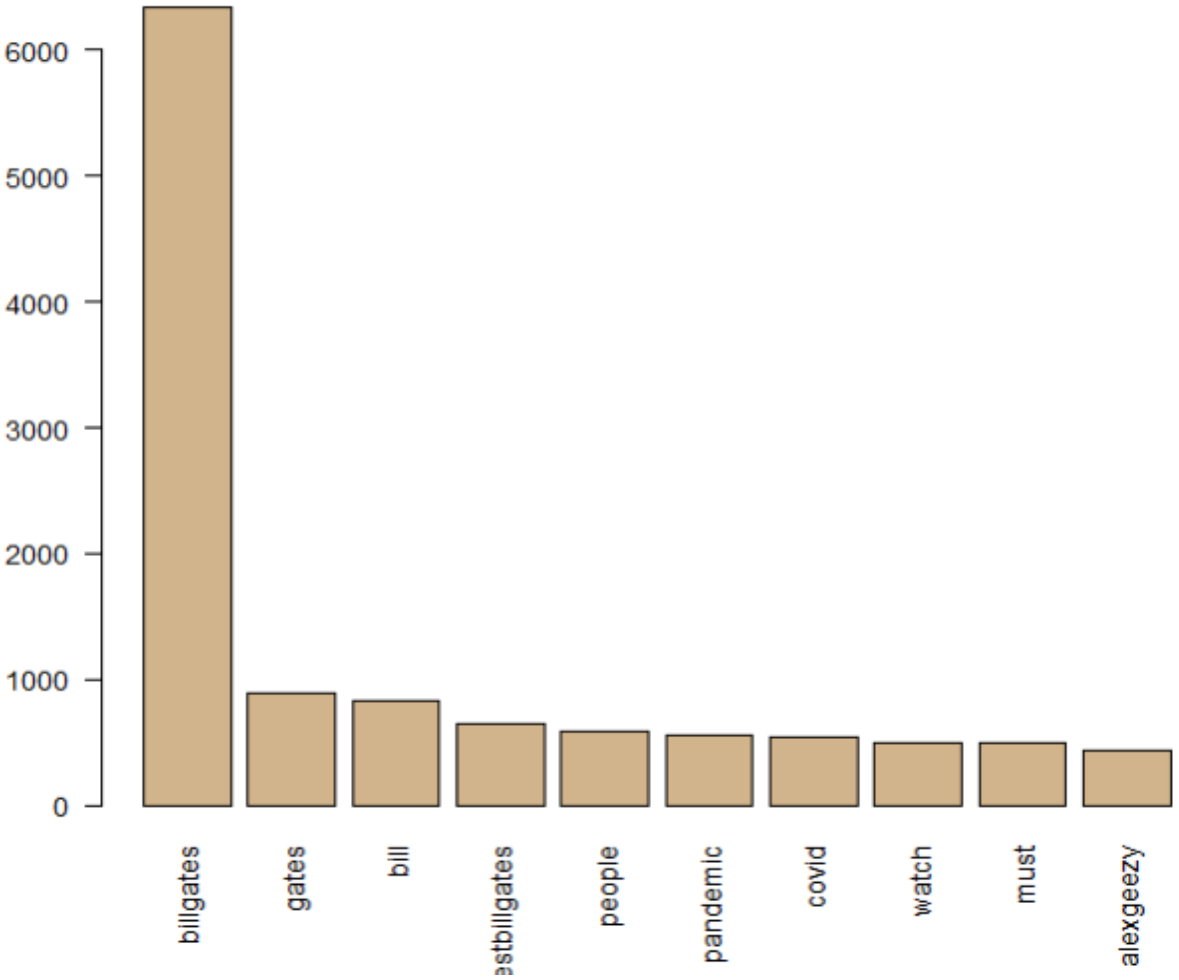
```
#Frequency Kullanılan kelimelerin sayılarının azalan sıraya göre sıralamak.
term_freq <- rowSums(BillData_m)
term_freq <- sort(term_freq, decreasing = TRUE)

term_freq[1:10] #10 tane veriyi getirdik.
```

10 Tane en çok tekrar eden veriyi getiriyoruz.

```
> term_freq <- sort(term_freq, decreasing = TRUE)
> term_freq[1:10] #10 tane veriyi getirdik.
    billgates         gates          bill arrestbillgates        people         covid      pandemic         watch          must
         6385           864           841           668           585           552           548           513           504
   alexgeezy
          449
>
```

Bu listenin bar grafiğini oluşturalım.

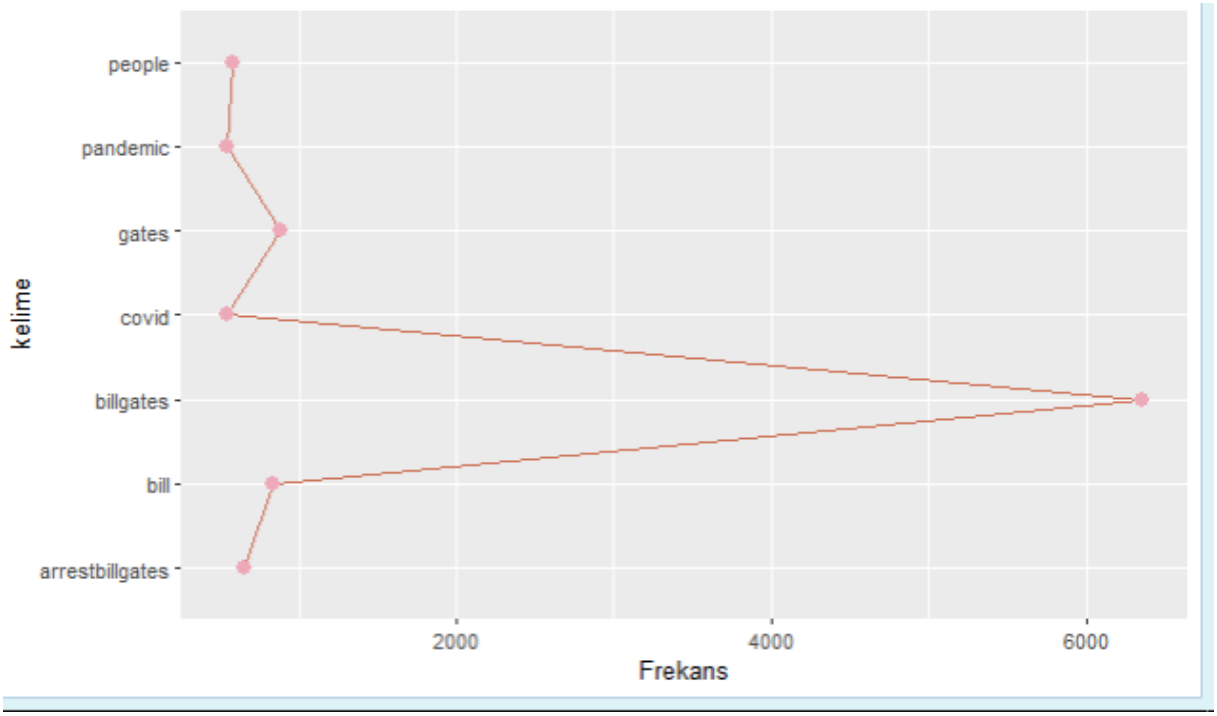
```
#Grafik halinde görelim
barplot(term_freq[1:10], col="tan", las=2)
```



Kelime Frekansı 500 ve üzeri olan farklı bir grafik görelim.

```
#Frekansı 500 ve daha üzeri olan kelimelerin grafiği
term_freq <- subset(term_freq, term_freq >=500)
term_freqs_df <- data.frame(term= names(term_freq), freq= term_freq)
ggplot(term_freqs_df, aes(x=term, y=freq)) +
            geom_line(aes(group=1), colour="salmon3")+
            geom_point(size=3, colour="pink2") + xlab("kelime") + ylab("Frekans") + coord_flip()
```

Burada da uygulamada ise 500 ve üzeri frekansa sahip kelimelerin **ggplot** grafiği oluşturuldu. Öncelikle 40 ve üzeri frekansa sahip kelimeler **term_freq** değişkeninde kümelendi. Daha sonra bu değerler **term_freq_df** adında bir veri çerçevesine atandı. Grafiği oluşturmak için ise "**ggplot2"** paketindeki **ggplot** fonksiyonundan yararlanıldı. Bu fonksiyon içeriği **geom_line( )** ve **geom_point( )** ile güçlendirildi ve renklendirildi.



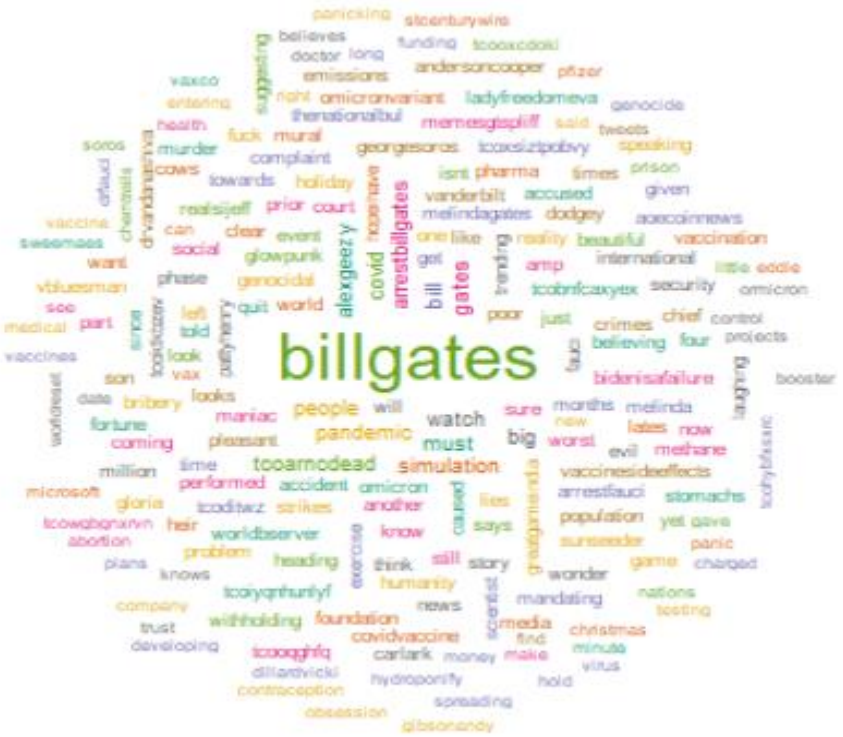
Kelime tekrarı 500 ve üzeri olan kelimelerin frekansları yukarıdaki gibidir.

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.

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

```
#Kelime Bulutu Olusturma

wordcloud(clean_corp, min.freq = 2, scale =c(2,0.5),colors = brewer.pal(8,"Dark2"),
          random.color = TRUE, random.order = FALSE, max.words = 200)
```



Bill Gates Verilerini “Covid” salgınıyla birlikte eşleştirdik ve çıkan sonuçlar: exercise, prior, performed gibi sonuçlarla ilişkilendirdik.

```
#İlişkilendirmeler

#BillGates'i covid salgınıyla ilgili ilişkilendirdik.
#Korelasyon limiti en az %50 olanları getirdik.
#Ve ilişkili kelimeler, "exercise","performed","prior" gibi kelimeler gelmiştir.


findAssocs(BillData_tdm, terms="covid",corlimit = 0.5)
```

```
> findAssocs(BillData_tdm, terms="covid",corlimit = 0.5)
$covid
   performed        prior   simulation worldbserver     exercise      reality       months        event
        0.58         0.58         0.58         0.58         0.57         0.57         0.56         0.54

>
```

4. Duygu Analizi – Semantic Analysis

Oluşturduğumuz kelime verileri üzerinden duygu analizi yapacağız. Bu analizin amacı kelimelerin anlamlarına göre olumlu, olumsuz gruplara ayrılmasını sağlamaktır. Twitter'dan aldığımız **"#BillGates"** ifadeli paylaşılan tweetleri**, "Anger", "Anticipation", "Disgust", "Fear", "Joy", " Sadness", "Surprise", "Trust", "Negative", "Pozitive"** duygu durumlarıyla ilgili analizlerimizi gerçekleştireceğiz.

Duygu Analizi süreci boyunca kullanacağımız paketler:

```
Source on Save
1 library(twitteR)  #Twitter Bağlantı Paketi paketimiz.
2 library(graphics) #Grafiklerimizi oluşturmak kullanılan paket.
3 library(purrr)    #Tidyverse ile kullanılır eksik bölümleri doldurur.
4 library(stringr)  #Karakter dizisi işlemleri için kullanılır.
5 library(tm)       #Metin Analizi için kullanılan bir pakettir.
6 library(syuzhet)  #Metinlerden duygu ve duyguya dayalı kelimelerin çıkarılması için bir pakettir.
7
8
```

- **TwitteR:** Twitter ile bağlantı paketidir.
- **Graphics:** Grafik Paketidir.
- **Purrr:** Eksik bölümleri tamamlayan pakettir.
- **Stringr:** Karakter dizisi işlemleri için kullanılır.
- **Tm:** Metin analizi için kullandığımız bir pakettir.
- **Syuzhet:** Metinlerden duygu ve duyguya dayalı kelimelerin çıkarılması için bir pakettir.

Paketlerimizi yükledikten ve kütüphanede aktif ettikten sonra veri işlemlerine geçebiliriz.

İlk Öncelikle Twitter API 'ye bağlantımızı kuruyoruz ve ardından **"#BillGates"** terimiyle ilgili 21 günlük toplam verileri çekiyoruz.

```
#API Bağlantısı:

api_key<-
api_secret <-
access_token <-
access_token_secret <-
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
```

“BillGates” ifadeli terimleri çekiyoruz ve listeye dönüştürüyoruz. Daha sonra çektiğimiz verileri bilgisayarımıza kaydediyoruz.

```
#Tweet Çekim İşlemi:

bill_tweets <- searchTwitteR("#BillGates", n = 20000)


tweets.df <- twListToDF(bill_tweets)

#Çektiğimiz verileri "csv" uzantılı olarak kaydediyoruz.
write.csv(tweets.df, file="tweetsBill.csv", row.names=FALSE)

#Verileri okuma işlemi
Read in data:

#Bağımsız değişken olarak yeni bir değişken tanımladık.
  setwd("C:/Users/Sefa/Documents/")
tweets.df<-read.csv("tweetsBill.csv")
```

Çektiğimiz veriler:

Çektiğimiz 10 adet veriyi getirdik ve bu verilerin analiz yapılmayacak kadar anlamsız ve karışık olduğunu görebiliyoruz. Bu yüzden verilerimizi temizleyelim ve daha saf ve temiz bir görüntüye ulaşmasını sağlayalım.

```
 [1] RT @GLOWPUNK: #BillGates strikes me as a pleasant genocidal maniac.

 [2] Einer der großen Profiteure dieser #plandemie \n\n#BillGates  #zwangsimpfung #Gates https://t.co/AtQXcmzQws

 [3] You mean #BillGates who thinks the population should be killed off through vaccinessss?  That #BillGates?\nSomeone n… https://t.co/tb1op1KD
nO
 [4] Genosse #Stegner kämpft heute gegen #BillGates und seine Atom-Trolle und dafür, dass der Glaube an eine #Bots-Versc… https://t.co/zgTzO89LE
z
 [5] RT @PattyHenry76: #arrestbillgates #BillGates #ArrestFauci #BidenIsAFailure #VaccineSideEffects Quit believing the lies! https://t.co/DITW
z…
 [6] @JuCourvoisier Tout était prévu et écrit par notre bon @jattali \nCes gens sont à gerber il faut les combattre… https://t.co/rFOHKL7GaE

 [7] RT @fuelcellsworks: . @Forbes - a Chat With @Enapter_'s Vaitea Cowan on the #GreenHydrogen Solution That Got #BillGates' Attention -- In t
h…
 [8] #stevenjoop ve #BillGates biyografisi #iPhone in başlaması ve büyümesi <U+270C><U+270C><U+270C>

 [9] RT @fuelcellsworks: . @Forbes - a Chat With @Enapter_'s Vaitea Cowan on the #GreenHydrogen Solution That Got #BillGates' Attention -- In t
h…
[10] If You Are Interested In Affiliate Marketing &amp;amp; Want to Make $1k/daYFire Then Check Out My BIO <U+270C> Follow Me… https://t.co/DZMo9nXu
2d
```

Verilerimizi Temizleme İşlemleri:

- **RemoveURL :** URL temizleme işlemi için kullanılır.
- **RemoveNumPunct:** İngilizce olmayan harfleri ve boşluklar için kullanılır.
- **PlainTextDocument:** Sade metin belgesine dönüştürme işlemine yarar.
- **StripWhitespace:** Kelimeler aralarında boşluklar eşitlenir.
- **ToLower:** Tüm kelimeleri küçük harfe dönüştürür.
- **RemoveWords:** Gereksiz tekrar eden kelime grupları kaldırılır.
- **RemovePunctution:** Noktalama işaretleri ve sayılar kaldırılır.

Stopwords: Türkçede kullandığımız etkisi kelimeler grubunun listesini bilgisayarımızdan çekiyoruz ve projemize ekliyoruz.

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

Örnek vermek gerekirse: Acaba, Ama, Ancak, Aslında, Elbette, da, de, gibi, gene, yine, az gibi kelime gruplarını etkisiz olarak ele alarak yeni bir dosya oluşturuyoruz.

Temizlenmiş tweetlerimizi metin belirteçlerine bölerek hangi kelimenin kaç kere tekrar ettiğini görebileceğimiz fonksiyonu yazalım.

```
#Verilerimizi Temizleme işlemini gerçekleştiriyoruz:

twitterCorpus <-Corpus(VectorSource(tweets.df$text)) #listeyi text dosyasına dönüştürdük.
inspect(twitterCorpus[1:10])                         #10 Adet veriyi getiriyoruz.
twitterCorpus<- tm_map(twitterCorpus, content_transformer(tolower)) #Bütün harfleri küçük harfe çeviriyoruz.
twitterCorpus<- tm_map(twitterCorpus,removeWords,stopwords("en")) #Tekrar eden verileri çıkartıyoruz.
twitterCorpus<- tm_map( twitterCorpus,removeNumbers)      # Rakamları ve özel karakterleri çıkartıyoruz.
twitterCorpus<- tm_map( twitterCorpus,removePunctuation)  # Noktalama işaretlerini ve özel karakterleri çıkartıyoruz.

removeURL<- function(x) gsub("http[[:alnum:]]*", "", x)
twitterCorpus<- tm_map(twitterCorpus,content_transformer(removeURL)) #URL kaldırır.

removeURL<- function(x) gsub("edua[[:alnum:]]*", "", x)   #Link ve dışındaki verileri kaldırır.
twitterCorpus<- tm_map(twitterCorpus,content_transformer(removeURL))


removeNonAscii<-function(x) textclean::replace_non_ascii(x)
twitterCorpus<-tm_map(twitterCorpus,content_transformer(removeNonAscii)) #Ascii karakterleri kaldırılır.

twitterCorpus<- tm_map(twitterCorpus,removeWords,c("amp","ufef",
                                                   "ufeft","uufefuufefuufef",
                                                   "uufef","s","rt","ufuf")) #Belirtilen tekrarlar çıkartılmıştır.

twitterCorpus<- tm_map(twitterCorpus,stripWhitespace) #Gereksiz Boşlukları sileriz.
```

Verileri yukarıdaki görseldeki gibi daha saf ve temiz hale getirmek için kullandığımız kodlar.

Temizleme işleminden sonra verilerimizi görelim.

Kontrol amaçlı 10 adet veri getiriyoruz.

```
2  inspect(twitterCorpus[1:10]) #Temizlenmiş verileri görelim.
3
```

Temizleme işlemi yaptığımız veri setinin yeni halini görüyoruz. Sade ve gereksiz kelimeler, gereksiz işaretler çıkartılmış durumdadır.

```
 [1] glowpunk billgates strikes pleasant genocidal maniac
 [2] einer der grossen profiteure dieser plandemie billgates zwangsimpfung gates tcoatqxcmzqws
 [3] mean billgates thinks population killed vaccinessss billgates someone n... tcotbopkdno
 [4] genosse stegner kampft heute gegen billgates und seine atomtrolle und dafur dass der glaube eine botsversc... tcozgtzolez
 [5] pattyhenry arrestbillgates billgates arrestfauci bidenisafailure vaccinesideeffects quit believing lies tcoditwz...
 [6] jucourvoisier tout etait prevu et ecrit par notre bon jattali ces gens sont a gerber il faut les combattre... tcorfohklgae
 [7] fuelcellsworks forbes - chat enapter' vaitea cowan greenhydrogen solution got billgates' attention th...
 [8] stevenjoop billgates biyografisi iphone baslamasi buyumesi ccc
 [9] fuelcellsworks forbes - chat enapter' vaitea cowan greenhydrogen solution got billgates' attention th...
[10] interested affiliate marketing want make kdayfire check bio c follow ... tcodzmonxud
```

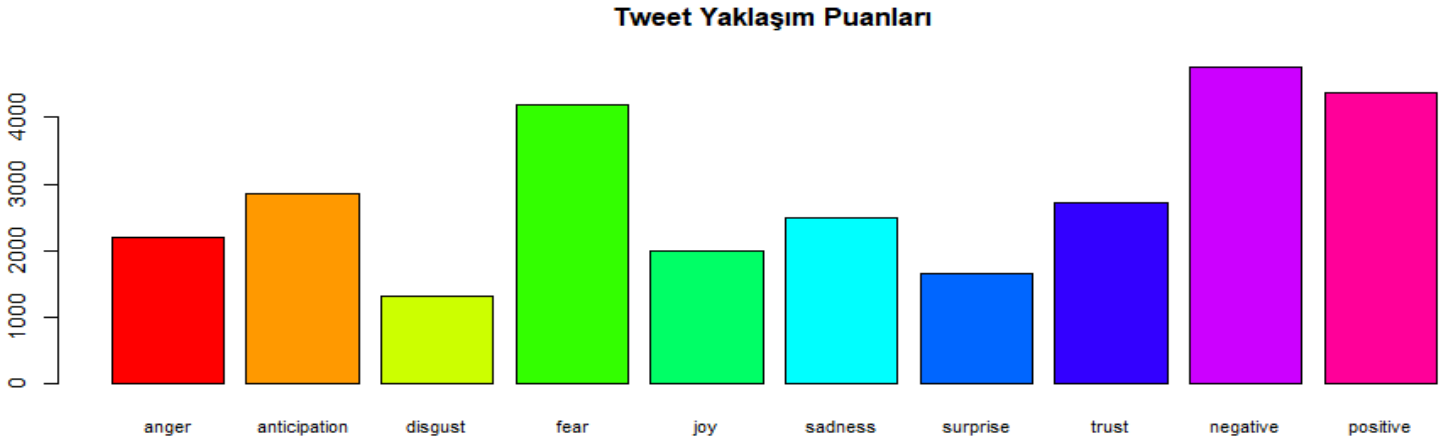
Çektiğimiz verileri yukarıda göründüğü gibi oldukça sade ve temizdir. Duygu analizi değerlendirmesi yapalım. Bunun için **"get_nrc_sentiment"** paketini kullanıyoruz. Bu kelime paketi içerisinde 10 Adet duygu durumunu barındırıyor. Bu değerlerden bahsedelim.

- **Anger:** Öfke
- **Anticipation:** Tahmin, Beklenti
- **Disgust:** Bıkkınlık, Sıkılma
- **Fear:** Korku
- **Joy:** Neşe
- **Sadness:** Üzüntü
- **Surprise:** Sürpriz
- **Trust:** Güven
- **Negative:** Negatif
- **Pozitive:** Pozitif

Oluşturduğumuz kod bloğunu incelediğimiz de ilk olarak get_ncr_sentiment veri sözlüğünden yararlandık ve paket içerisinde var olan 10 duygu türü ile çektiğimiz **"twitterCorpus"** verilerini inceledik.

```
#Burada get_ncr_sentiment sözlüğünden duygu kelimelerini alıyoruz.
#Bu paketin içerisinde 10 adet duygu değeri vardır.
#Bu değerler: Öfke, Tahmin, Bıkkınlık, Korku, Neşe, Pozitif, Negatif,
# Üzüntü, Güven, Sürpriz bu kelimelere göre analizler yapılarak
#değerler belirlenmiştir.
emotions<-get_nrc_sentiment(twitterCorpus$content)
barplot(colSums(emotions),cex.names = .8,
        col = rainbow(12),
        main = "Tweet Yaklaşım Puanları"
)
```

Kod bloğunu çalıştırdıktan sonra karşımıza gelen grafiği inceliyoruz.



İncelediğimiz **"Bill Gates"** ile alakalı paylaşılan 21 günlük tweetlerin duygu durumları yukarıdaki görseldedir. Görseli yorumladığımız da paylaşılan tweetlerin çoğunluğu Negative veriler olduğunu görebiliyoruz. Bunun ardından dünyada oluşan pandemiyle dönemiyle birlikte korku ve pozitif değerler neredeyse birbirlerine yakın. Korku değerinin yüksek çıkmasının Bill Gates için vurulan aşılarda çip olduğuna dair söylentiler var. En düşük gözlemlediğimiz duygu durumu ise bıkkınlık olarak karşımıza geliyor.

Kaynakça

- https://www.springboard.com/library/data-science/data-mining/
- https://leventcan.github.io/blog/twitter-ile-i%C3%A7erik-analizi/
- http://varianceexplained.org/r/trump-tweets/
- https://marketreading.com/tr/scraping-twitter-data-with-the-r-language.html
- https://github.com/kaveai/veribilimiyazokulu/blob/main/Python%20ve%20Veri%20Bilimi%20%C3%96rnekleri/Twitter%20Verisi%20ile%20Duygu%20Analizi.ipynb
- https://medium.com/@elifarslan099/r-dplyr-paketi-7aea18a40e2d
- https://bookdown.org/content/2096/ra-veri-yukleme.html
- http://gulsahsemiz.com/tr/veri-analizi-icin-olmazsa-olmaz-r-paketleri/