



TÜRKİYE CUMHURİYETİ

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

ÇAĞLA YURTSEVEN

DOÇ. DR. NUR KUBAN TORUN

Veri Madenciliği

Mevzular Açık Mikrofon | 6284 Özel

Bilecik

26.12.2024

İçindekiler

ÖNSÖZ	I
ÖZET	II
1. GİRİŞ	5
1.1. Metin Madenciliği	5
1.2. Duygu Analizi	5
2. YÖNTEM	5
2.1. Kapsam ve Kısıtlar	5
2.2. Projenin Amacı ve Önemi	5
2.3. YouTube API İşlemleri	5
2.4. Veri Çekme İşlemi	6
2.5. R’da Kullanılan Kütüphaneler	7
2.6. R Programlamada Kullanılan Kodlar ve Analiz Aşamaları	7
3. BULGULAR	12
3.1. Kelime Bulutu	12
3.2. Frekans Tablosu	12
3.3. Kelime Ağı Grafiği	13
3.4. Duygu Analizi	13
4. SONUÇ VE TARTIŞMA	15
5. KAYNAKÇA	16

ÖNSÖZ

Bu çalışmayı 4.sınıfta almış olduğum Veri Madenciliği dersi için **Mevzular Açık Mikrofon | 6284 Özel** adlı videosunu analiz etmek için yapmış bulunmaktadı. Ana hedefim videoda bulunan yorumları analiz ederek insanların video hakkındaki düşüncelerini gün yüzüne çıkartmaktır.

Bu rapor hazırlarken Youtube, R ve Python gibi araçlar kullanılarak hazırlanmıştır. Bu projeyi gerçekleştirmek, veri madenciliği konusunda edindiğim bilgileri pratiğe dökerek hem teknik hem de analitik becerilerimin gelişmesinde katkı sağlanmıştır. Elde ettiğim deneyimler doğrultusunda aldığımız eğitimin gelecekte kariyerimde önemli bir yol gösterici olacağına inanıyorum.

Projeyi hazırlarken verdiği proje fikriyle, öğretim ve destekleriyle projeyi başarılı bir şekilde tamamlamama yardımcı olan Doç.Dr.Nur Kuban Torun'a teşekkürümü borç bilirim. Bu zamana kadar verdiği emekleri için teşekkür ediyorum.

25.12.2024
Çağla Yurtseven

Özet

Bu çalışma doğrultusunda **Mevzular Açık Mikrofon | 6284 Özel** adlı videosuna yapılan yorumları analiz ederek izleyicilerin video hakkındaki görüşlerini anlamayı ve yorumlamayı amaçlanmıştır. Veri madenciliği, Metin madenciliği, ve duygu analizi yöntemlerini kullanarak, Youtube platformundan elde edilen yorumları incelenmiştir. Analiz sürecinde R ve Python programlama dillerinden yararlanılmıştır.

1.GİRİŞ

1.1 Metin Madenciliği

Büyük miktardaki metin verisinden anlamlı bilgiler ve desenler çıkarılması diyebiliriz. Bu süreçte dil işleme teknikleri ya da istatistiksel yöntemleri kullanarak metin içeriklerini analiz edebilir hale getirmeyi amaçlar. Metin madenciliği kullanıcı yorumları, sosyal medya paylaşımları ve bir çok kaynaktan elde ettiği verileri analiz edilmesini sağlar.

Metin madenciliği teknikleri yüksek boyutlu metin içeriklerinden gözle görülmeyecek içeriklerin, ilişkilerin ve örüntülerin çıkartılarak bunların yeni iş fırsatlarında ve süreçlerinde kullanılmasını sağlar. Metin madenciliği zaman ve kaynak tasarrufu sağlar.

1.3 Duygu Analizi

Duygu analizi bir metnin duygusal tonunu yani verilerde hangi duyguları ifade ettiğini otomatik olarak belirlemeye yönelik hesaplama işlemidir. Bir metnin duygusu genellikle pozitif, negatif ve nötr olarak analiz edilir. Duygu analizi genellikle sosyal medya, müşteri geri bildirimleri ya da benimde kullandığım yorumlar için oldukça faydalıdır.

2. YÖNTEM

2.1 Kapsam ve Kısıtlar

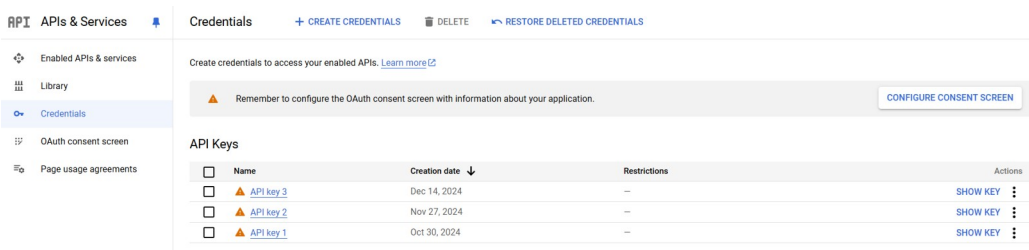
Bu çalışmamda metin madenciliği ve duygu analizi tekniklerini Türkçe metinler üzerinden analizler yaparak incelemiş bulunmaktadır. Burada yapmak istediğim Youtube üzerinden kullanıcı yorumlarını alındı. Bu Youtube videosunda 8334 adet veri çekilmiştir. Bu veriler son olarak 17.12.2024 tarihinde çekildiğinin altını çizirim. Böylelikle zaman kısıtını beraberinde getirdi.

2.2 Projenin Amacı

Youtube kullanıcılarının 6284 adlı yasa ve ülkemizde yaşanan kadın cinayetlerine ait verdikleri tepkileri anlamak ve yorumlamak ayrıca video analiz edilmiştir. Bu analizle birlikte, kullanıcıların bu konu hakkındaki duygu ve düşüncelerini belirlenmesi ve videonun izlenmeden de anlaşılıp anlaşılmadığını öğrenmek.

2.3 Youtube Api İşlemleri

Google Cloud'un APIs&services kısmını kullanarak API Keys oluşturacağız. Burada verilen API'yi hangi yazılım dili ile çekmek istiyorsak oraya yazacağız. Credentials satırında yer alan "+CREATE CREDENTIALS" kısmından API Key tıklayarak elde edebiliriz.



Name	Creation date	Restrictions	Actions
API key 3	Dec 14, 2024	—	SHOW KEY
API key 2	Nov 27, 2024	—	SHOW KEY
API key 1	Oct 30, 2024	—	SHOW KEY

2.4 Veri Çekme İşlemi

Verileri Youtube üzerinden çekmek için python dilini kullanıldı. Veri çekerken Youtube api oluşturmak için öncelikle hesap açıp ardından kullanmak istenildi. Şekil 1.0 video da bulunan “video id” ile gerçekleştirildi. Bunları çekerken Python da bulunan hazır kütüphaneleri kullandım. Ardından bunları bir csv dosyası formatından kaydedildi.

Veri çekme işlemi yaparken kullanılan kütüphane “googleapiclient.discovery.build” biridir bu kütüphane Google API istemcisi ile etkileşim kurmayı sağlıyor. Bu kod Youtube API ile çalışmak için kullanıldığını söyleyebiliriz. Diğer bir kütüphanemiz ise “csv” bu da dosyaları okuma ve yazma işlemleri için kullanacağız. Bu kodda, Youtube API ile çektiğimiz yorumları bir csv dosyasına kaydedildi. Kod ekranında bahsettiğim gibi Api istemcisi oluşturma, yorum çekme ve csv dosyasını kaydederek işlemleri tamamlandı.

```
C > Users > cyurt > OneDrive > Masaüstü > Veri Madenciliği Final Çağla Yurtseven > finaldev.py
1
2 #VERİLERİ ÇEKME İŞLEMİ
3 # Youtube Data API ile video yorumları çekme işlemi
4 from googleapiclient.discovery import build # Youtube API ile etkileşim kurmayı sağlar.
5 import csv # Elde ettiğimiz verileri CSV dosyasına kaydetmeye yardımcı kütüphane
6 # API anahtar ve video ID'si
7 api_key = "GvHmMfIVLom" # Youtube API anahtar buraya yapıştırıldı.
8 video_id = "GvHmMfIVLom" # Video ID'si
9 # Youtube API istemcisi
10 youtube = build('youtube', 'v3', developerKey=api_key) #buil ile youtube api istemcisi oluşturuldu ve sınıflı belirttim
11 # Çekilen yorumları saklamak için Python listesi oluşturduk
12 all_comments = []
13 # İlk sayfa
14 request = youtube.commentThreads().list( #request ile api iletişimi yapılır.
15     part="snippet", #yorum üzerisinin temel bilgilerini alacağımızı belirtiyor.
16     videoId=video_id, #Belirtilen video ID ait yorumları getirir.
17     maxResults=100 #her istekte max 100 yorum alıyoruz
18 )
19 response = request.execute() #istediğimiz ilk sayfanın yorumlarını al
20 # Tüm yorumları çek yani bu döngü ile tüm yorumları çekme işlemi yapıyoruz
21 while response: #API den gelen veriler olduğu sürece döngü devam ediyor.
22     for item in response['items']:
23         comment = item['snippet']['topLevelComment']['snippet']['textDisplay'] #textDisplay demek yorumun metin kısmının alınması.
24         all_comments.append(comment) #yorumu listeye ekle
25     # Sonraki sayfa token'i kontrol et
26     if 'nextPageToken' in response:
27         request = youtube.commentThreads().list(
28             part="snippet",
29             videoId=video_id,
30             maxResults=100, #Buradaki ama 100den fazla yorum varsa sonraki sayfa için token alınır
31             pageToken=response['nextPageToken']
32         )
33         response = request.execute()
34     else:
35         break #sayfa bittiğinde döngü sonlandır
36 # Yorumları bir CSV dosyasına kaydet
37 with open('vericekme.csv', 'w', newline='', encoding='utf-8') as file: #yeni csv dosyası açılır
38     writer = csv.writer(file) #csv yazma yapılır
39     writer.writerow(['Comment']) #1.satır başlık ekleme
40     for comment in all_comments:
41         writer.writerow([comment])
42     print(f"Toplam {len(all_comments)} yorum indirildi ve 'vericekme.csv' dosyasına kaydedildi.") #işlemin tamamlandığına dair en son haber verir.
43
```

Şekil 1: Yorumları Çekme

2.5 R' da Kullanılan Kütüphaneler

Kullanılan kütüphaneleri kısaca özetleyecek olursam çalışmada veri manipülasyonu, analizi ve görselleştirme işlemleri için R kütüphaneleri kullanılmıştır. Veri manipülasyonu ve temizleme işlemlerinde “dplyr,tidyverse ve tidyr” kullanıldı. Metin manipülasyonu ve temizleme işlemi için “stringr, textclean” kullanılmıştır. Metin madenciliği ve analizinde “tm, tidytext, syuzhet ve topicmodels” kullanıldı.

Veri görselleştirme araçlarında ise “ggplot2, wordcloud2 ve ggwordcloud” kullanıldı. Etkileşimli ağ grafiğini oluşturmak içinde “igraph, ggraph ve visNetwork” tercih edilmiştir.

```
2 library(dplyr)#veri manipulasyonu icin
3 library(ggraph)#grafik gorsellestirme
4 library(syuzhet)#duygu analizi yapmak
5 library(tidyverse)#veri bilimi icin daha genis bir kutuphane
6 library(tidyr)#verileri temizlemek
7 library(stringr)#karakter ve metin manipulasyonu
8 library(textclean)#verileri temizlemek
9 library(topicmodels)#LDA modelleme yapmak
10 library(tidytext)#metin verilerini analiz etmek
11 library(tm)#metin madenciligi icin kullanan
12 library(igraph)#ag analizi ve gorsellestirme
13 library(knitr)#dinamik rapor olusturup kod blogu ile birlestirme
14 library(ggplot2)#veri gorsellestirme
15 library(wordcloud2)#kelime bulutu olusturma
16 library(ggwordcloud)#ggplot2 üzerinden daha detayli gorsellestirme
17 library(visNetwork) #Interaktif ag grafikleri olusturmak icin kullanan paket. Web tabanlı olarak etkileşimli
```

2.6 R Programda Kullanılan Kodlar ve Analiz Aşamaları

Python ile oluşturduğum (Şekil 1: Yorumları Çekme) yorumları R' a getirilmiştir.(Şekil 2: CSV dosyasını getirme)

```
yorumlar <- read.table (file.choose(),header=T,sep=";") #Python'da olusturdugum vericekme.csv dosyasini getirmek
```

(Şekil 2: CSV dosyasını getirme)

Çektiğim yorumları daha anlaşılabilir hale getirip analize uygun şekilde düzeltildi. Analizi anlaşılır bir hale getirmek için yaptıklarımı sıralayacak olursam html ve url temizleme, özel karakter temizliği yapma, fazla boşlukları kaldırma gibi düzenleme işlemi yapmak amaçlanmıştır.

```
21 #Verileri Temizleme
22 yorumlar$Comment <- as.character(yorumlar$Comment)#yorumlar karakter formatina cevrilir
23 yorumlar$Comment <- str_remove_all(yorumlar$Comment, "<[>]+>")# HTML temizleme
24 yorumlar$Comment <- str_remove_all(yorumlar$Comment, "http\\S+|www\\S+")#url temizleme silme
25 yorumlar$Comment <- replace_emoji(yorumlar$Comment, replacement = "")# Emojileri kaldırma
26 yorumlar$Comment <- str_remove_all(yorumlar$Comment, "[^[:alpha:][:space:]]")# Özel karakter temizliği
27 yorumlar$Comment <- str_squish(yorumlar$Comment) # Fazla bosluklar?? temizler
28 yorumlar$Comment <- tolower(yorumlar$Comment) # Tum harfleri kucuge ceviriir
29 yorumlar <- yorumlar %>% filter(Comment != "" & !is.na(Comment)) # Bos satirlari sil
30 write.csv(yorumlar, "temizlenmis_yorumlar.csv", row.names = FALSE)
31
32 # Sadece dolu olan satirlari secme
33 yorumlar <- yorumlar %>% filter(Comment != "" & !is.na(Comment) & str_squish(Comment) != "")
34 head(yorumlar)# ilk birkac satira goz atma
35 dim(yorumlar)# Satir ve sutun sayisini gorme
```

(Şekil 3: Yorumları Temizleme)

Burada Türkçe de gereksiz tekrar eden kelimeleri yorumlar kısmından temizleyerek kelime bulutu oluştururken daha iyi sonuç vermesini sağlamayı amaçlandı.

```

37 Sys.setlocale("LC_ALL", "Turkish")
38 turkce_durkelime <- c(
39   "yani", "bize", "fazla", "quot", "umarım", "kendim", "benim", "eden", "acaba", "hangi", "fakat", "onun", "diğer", "büyük", "ya", "da", "ki", "olmuş",
40   "çok", "bir", "de", "yada", "hale", "seni", "iyi", "hep", "olarak", "ö?nce", "varsa", "kaç", "ele", "yere", "varsa", "yeni", "başka", "olacak", "önemli",
41   "dolu", "işte", "nasıl", "söyle", "tane", "ilgili", "olup", "diyor", "sadece", "hepsi", "süreçe", "once", "teşekkürler", "lütfe", "gerekten",
42   "herkes", "ben", "bana", "sonra", "eğer", "artık", "bence", "kişi", "ilk", "sana", "bu", "şu", "o", "ve", "en", "nasıl", "olduğu", "zorunda",
43   "veya", "ama", "için", "bizi", "tek", "yine", "cok", "sizi", "hic", "hi??", "peki", "çünkü", "ki", "de", "da", "herkesin", "yerine", "ediyorum",
44   "ile", "mi", "ne", "bile", "tüm", "belki", "??uan", "evet", "olan", "iki", "son", "adım", "size", "hala", "olsun", "zaten", "gerçekten", "beni",
45   "dedi", "hayır", "ben", "sen", "o", "biz", "siz", "onlar", "bunu", "böyle", "çok", "az", "buna", "biri", "durak", "bizim", "doğru",
46   "daha", "gibi", "ise", "hem", "her", "hi??", "de", "da", "iyi", "ne", "hic", "şey", "var", "gibi", "ne", "bu", "bir", "içinde", "kimse", "versin",
47   "gün", "ben", "için", "gibi", "var", "şeyler", "en", "şimdi", "var", "yok", "nasıl", "neden", "değil", "diye", "etti", "gibi", "önce", "olmak",
48   "hiç", "ama", "var", "ne", "o", "ya", "ay", "hiçbir", "an", "şu", "neye", "kadar", "asla", "keşke", "göre", "nerede", "kim", "olsa", "devam",
49   "bütün", "birkac", "bazı", "için", "şey", "sizin", "lazım", "aynı", "şeyi", "boş", "onu", "geçen", "ancak", "olur", "olması", "oldu", "gerekten",
50   "bende", "demek", "kesinlikle", "tarafından", "tekrar", "olmalı", "hatta", "diyen", "bu", "kadar", "karşında", "olur")

```

(Şekil 4: Kelime Temizleme)

En çok tekrar eden kelimeleri tekrar boyutuna göre ayarlayarak görselleştirme yapıyoruz.

```

53 #yorumlari kelimelere bolme ve frekans islemi yapma(bir kelimeden kac tane var gibi)
54 kelime_frekans_duzenli <- yorumlar %>%
55   unnest_tokens(word, Comment) %>%
56   count(word, sort = TRUE) %>%
57   filter(!word %in% turkce_durkelime) %>% # Stopwords'leri cikar
58   filter(nchar(word) > 2) # Kisa kelimeleri filtrele
59
60 kelime_frekans_duzenli <- kelime_frekans_duzenli %>% slice_max(order_by = n, n = 50) # En sik gecen 50 kelimeyi secme
61 #burasi gorsellestirme kelimelerin tekrarına göre boyut ayarlama
62 wordcloud2(
63   kelime_frekans_duzenli,
64   size = 1.5,
65   color = c("darkred", "black", "gray"), # Renkleri ayarlama
66   backgroundColor = "lightpink" # Arka plan rengi
67 )

```

(Şekil 5: Kelime Bulutu Oluşturma)

Kelime bulutunda elde ettiğimiz verilere karşılık bu sefer de bir yatay tablo halinde en çok tekrar eden 20 kelimeyi görmek için (Şekil 6: En çok tekrar eden 20 kelime) şeklinde bakabiliriz.

```

70 ##En cok tekrar eden 20 kelimeyi secme
71 top20_kelime <- kelime_frekans_duzenli %>%
72   top_n(20, n)
73
74 # Tekrar eden o 20 kelimeyi gorsellestirme
75 ggplot(top20_kelime, aes(x = reorder(word, n), y = n)) +
76   geom_bar(stat = "identity", fill = "pink") +
77   coord_flip() +
78   labs(
79     title = "En cok Tekrar Eden 20 Kelime",
80     x = "Kelime",
81     y = "Frekans"
82   ) +
83   theme_minimal()

```

(Şekil 6: En çok tekrar eden 20 kelime)

Buradaki amaç kelimelerin birbirleriyle olan bağlantısını öğrenmek ve bu bağlantılarla keşfetmek istediğimiz kelime ağlarına ulaşmayı hedefliyoruz.

```

85 ##Kelime Ağı Grafiği
86 turkce_stop_words <- c("ve", "i", "o", "45", "fazla", "daha", "hiçbir", "en", "azından", "şey", "39", "diken",
87   "dediği", "gibi", "ama", "bu", "çok", "bir", "4", "saat", "her", "da", "de", "ile",
88   "kadar", "olan", "şuan", "şu", "an", "var", "değil", "mu", "mi", "yüzden")
89 bigramlar <- yorumlar %>%
90   unnest_tokens(bigram, Comment, token = "ngrams", n = 2) %>% #yorum metinleri analiz edilmesi ve 2 kelimelik gruplar cikar
91   separate(bigram, into = c("kaynak", "hedef"), sep = " ") %>% #iki sutuna ayrılma olur kaynak ve hedef olarak
92   filter(!kaynak %in% turkce_stop_words, !hedef %in% turkce_stop_words) %>% #gereksiz kelime temizliği
93   filter(!is.na(kaynak), !is.na(hedef)) %>% #gecersiz bigraflar cikar
94   count(kaynak, hedef, sort = TRUE)
95
96 bigramlar <- bigramlar %>% filter(n > 3) #3 veya daha fazla tekrar eden
97 bigramlar <- bigramlar %>% slice(1:50) #sadece en çok tekrar eden 50 bigrafi secer
98 bigram_graf <- graph_from_data_frame(bigramlar) # ve Bigram ağını olusturulur
99 head(bigramlar) # ilk birkac satıra göz atma
100 #Sade gorsellestirme
101 ggraph(bigram_graf, layout = "fr") + # Gorsellestirme kısmi
102   geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, edge_width = 0.5) +
103   geom_node_point(color = "pink", size = 5) + # Dugumleri tek renk yapıyoruz
104   geom_node_text(aes(label = name), repel = TRUE, size = 3) +
105   theme_void() +
106   labs(title = "Kelime Ağı Grafiği", color = "Dugum Kategorisi")

```

(Şekil 7: Kelime Ağı Oluşturma)


```

#Özelleştirilmiş Gorselleştirme
# Düğüm ve kenar veri çerçeveleri
nodes <- data.frame(
  id = V(bigram_graf)$name, #benzersiz kimlik (her dugumun)
  label = V(bigram_graf)$name, #dugum ustundeki isim
  value = degree(bigram_graf), # dugumun büyüklüğü belirleme
  group = ifelse(degree(bigram_graf) > 5, "High", "Low") # 5den fazla olanlar high, 5 veya daha az low
)

edges <- data.frame(
  from = bigramlar$kaynak,
  to = bigramlar$hedef,
  value = bigramlar$n #kenar kalınlığı tekrar sayısına göre
)

# Interaktif ağ görselleştirme
visNetwork(nodes, edges, width = "150%", height = "900px") %>%
  visEdges(arrows = "to", smooth = TRUE) %>% #kenarlık yonu ve kenarların düzgün gözükmesi
  visNodes(color = list(highlight = "red", border = "black")) %>%
  visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%
  visPhysics(stabilization = TRUE) %>%
  visLayout(randomSeed = 42)

```

(Şekil 7.1: Kelime Ağı Oluşturma)

Polarite analizi için öncelikle kelimeleri Türkçe'den İngilizce'ye çevireceğiz çünkü Türkçe kelimeler ile kullanılan paketin doğru bir sonuç vermeyeceğini bilindiği için bu yola başvuruldu. Bunun sonucunda kelimeleri Python da kelimeleri çevirecek bir döngü yazıldı ve her bir CSV olarak toplandı bunun sonucunda başka bir kod yazıp hepsini aynı aynı dosya içerisinde yazarak birleştirildi.

```

import pandas as pd #csv dosyalarını okumak için
from googletrans import Translator #metinli farklı dile çevirmek için
import pandas as pd
from googletrans import Translator

# Çevirici oluşturma
translator = Translator()

#csv dosyasını okuyup karakter kodlaması yapılır
tablo = pd.read_csv('kelime_frekans_polarite.csv', encoding='utf-8')

# toplam satır sayısını öğrenme
total_rows = len(tablo)

batch_size = 50 #bir seferde kaç satır işleneceğini yazarız burada 50 satır

start_from = 6740 #başlangıç noktasını belirliyoruz

# Çevirme işlemi
def ceviri(row):
    try:
        return translator.translate(row, src='tr', dest='en').text #türkçeden ing çevirme
    except Exception as e:
        return str(e)

for start in range(start_from, total_rows, batch_size): #döngü start_from başlar ve total_rows değerine kadar her adımda 50 satır kadar ilerler
    end = start + batch_size #batch bitiş satır numarasını hesaplar
    tablo_sample = tablo.iloc[start:end] #start ve end arasındaki satırları seçer

    # Kelimeleri çevir
    tablo_sample['ingilizce_kelime'] = tablo_sample['word'].apply(ceviri) #word sütunundaki her kelime için ceviri fonk uygular
    batch_filename = f'cevirilmis_yorumlar_batch_{start + 1}_{end}.csv' #dosya kaydetme işlemi
    tablo_sample.to_csv(batch_filename, index=False, encoding='utf-8') #dosya karakter kodlaması
    print(f"Batch {start + 1} - {end} için çeviri tamamlandı ve {batch_filename} olarak kaydedildi.") #tamamlandığını bildirmek

```

(Şekil 8: Python kelimeleri ingilizceye çevirme)

```

import pandas as pd
import glob #belirtilen dosya yoluna uyan dosyaların listesini alma

# dosya yolunu bulmak için
batch_files = glob.glob('C:/Users/cyurt/OneDrive/Masaüstü/VeriMadenciliği/batch_files/cevrilmis_yorumlar_batch_*.csv')

if not batch_files:
    print("Hiç batch dosyası bulunamadı. Lütfen dosyaları kontrol edin.") #dosya boşsa uyarı ver
else:
    print("Batch dosyaları bulundu:", batch_files) #boş değilse bulunan dosyaları ekrana yaz

    # Dosyaları birleştirme işlemi
    all_batches = [] #birleştirilecek tüm dosyalar için dataframe saklamak için liste oluşturma
    for file in batch_files:#dosya için döngü başlatılıyor
        df = pd.read_csv(file, encoding='utf-8')
        all_batches.append(df) #listeye ekleme

    # Birleştirilmiş DataFrame
    final_result = pd.concat(all_batches, ignore_index=True) #tek bir dataframe olarak birleştirilip yeni indeks oluşturma işlemi

    # Sonuçları kaydet
    final_result.to_csv('final_cevrilmis_yorumlar.csv', index=False, encoding='utf-8')
    print("Batch dosyaları başarıyla birleştirildi ve final_cevrilmis_yorumlar.csv olarak kaydedildi.")

```

(Şekil 9: Çevrilen kelimelerin Dosyaya Toplanması)

```

1 polarite_kelimeleri <- read.table (file.choose(),header=T,sep=";") #pythonda olusturdugum ing kelimeleri getirdim(cevirdigim)
2 sum(is.na(polarite_kelimeleri$ceviri)) # NA de?erlerinin sayisini kontrol et
3 polarite_kelimeleri <- polarite_kelimeleri %>% filter(!is.na(ceviri)) # NA olan satirlari cikar
4 # ozel karakterleri temizlemek için örnek
5 polarite_kelimeleri$ceviri <- gsub("[^[:alnum:]][:space:]", "", polarite_kelimeleri$ceviri..)
6 polarite <- get_sentiment(polarite_kelimeleri$ceviri, method = "syuzhet")
7 #syuzhet paketi ile kelimelerin duygu skorlari her yorum için polarite skoru
8 polarite_kelimeleri$Polarite <- polarite#hesaplanan polarite skorlarini ekleme
9 # Kelime Frekansi ile Polariteyi Birlestirme
10 tablo_2 <- polarite_kelimeleri %>%
11 unnest_tokens(word, ceviri) %>%#tidytext paketinin fonksiyonu ile "tek tek kelimelere ayirma" her yorumu
12 count(word, sort = TRUE) %>% #her kelime kac kez gecti ve azalan sirada sirala
13 filter(nchar(word) > 2) %>% #kelime karakterlerini 2 den fazla olanlari al
14 mutate(sentiment = get_sentiment(word, method = "syuzhet"))#kelime polaritesini hesaplama ve yeni sutun olusturma

```

(Şekil 9: Polarite Oluşturma)

Polarite analizi kullanarak matematiksel analiz oluşturuldu. Buradaki amaç matematiksel olarak bu çıktıların negatif, pozitif ve nötr sonuçlarına ulaşarak yorumlamaktır. Burada ki amacımız Python'da yapılan İngilizce kelimeleri R'da getirerek polarite analizi tamamlamak.

```

16 #Polarite Grafigi
17 ggplot(tablo_2, aes(x = n, y = sentiment, color = sentiment)) +
18   geom_point(size = 2) + # noktaların boyutlari ayarlama
19   scale_color_gradient2(low = "blue", mid = "purple", high = "orange",
20     midpoint = 0) + # Renk ayarlama
21   geom_hline(yintercept = mean(tablo_2$sentiment, na.rm = TRUE),
22     color = "red", size = 0,65, linetype = "dashed") + # Polarite cizgisi
23   labs(title = "Duygu Analiz Sonucu",
24     x = "Kelime Frekansi",
25     y = "Skor",
26     color = "Ortalama Polarite") + # Renklerin degisim gostergesi
27   theme_minimal()
28 #Matematiksel degerleri hesaplama
29 duygu_ozet <- polarite_kelimeleri %>%
30   summarise(
31     mean = mean(Polarite, na.rm = TRUE), # Ortalama
32     se.mean = sd(Polarite, na.rm = TRUE) / sqrt(n()), # Ortalama Standart Hata
33     CI.mean.0.95 = qt(0.975, df = n() - 1) * (sd(Polarite, na.rm = TRUE) / sqrt(n())), #%95 guven araligi
34     var = var(Polarite, na.rm = TRUE), # Varyans
35     std.dev = sd(Polarite, na.rm = TRUE), # Standart Sapma
36     coef.var = (sd(Polarite, na.rm = TRUE) / mean(Polarite, na.rm = TRUE)) * 100 # Varyasyon Katsayisi
37   )
38 #Tabloyu yazdirma
39 print(duygu_ozet)
40 #Tablonun daha duzenli gorunumu için kable paketi
41 kable(duygu_ozet, caption = "Duygu Analizi Matematiksel Sonucu", digits = 5)

```

(Şekil 10: Polarite Matematiksel Oluşturulması)

Türkçe kelimeler ile “Pozitif ve Negatif” kelimeleri öğrenmek istedim. Bunu yaparken en başta oluşturduğum kelime frekans kullandım yani kelimeleri İngilizceye çevirmeden “türkçe_dur” kelimelerinden ayırarak pozitif ve negatif şekilde Duygu analiz kelimeleri uygulandı.

```
1  turkce_dur_frekans <- c("ve", "i", "o", "45", "fazla", "daha", "hiçbir", "en", "azından", "şey", "39", "diken", "dediği", "gibi", "ana", "bu", "çok", "bir", "4", "saat",
2  "her", "da", "de", "ile", "bunu", "böyle", "kadar", "olan", "şuan", "şu", "an", "var", "değil", "ve", "mi", "mi", "yüzden", "olsun", "ancak", "vs",
3  "fakat", "gelen", "bize", "ki", "ben", "sen", "biz", "siz", "onlar", "kendisi", "diye", "nasıl?", "şimdi", "yine", "iki", "ona", "orda", "varsa",
4  "hem", "göre", "kaç", "olmalı", "onu", "bütün", "sadece", "aslında", "herhangi", "suanda", "gerektiğinde", "yapılan", "gerek", "neden", "yapmak",
5  "bunlar", "yeni", "belirli", "olmaz", "bazı", "yapma", "herkes", "görüntü", "birçok", "sürekli", "şunu", "tabii", "bişey", "uygulama", "alışan", "ne",
6  "için", "ya", "bile", "bi", "tek", "çok", "gün", "aynı", "bana", "uğur", "e", "şey", "dile", "hemen", "dedim", "sey", "peki", "çoğu", "hocam", "kez", "ay", "oy", "hic",
7  "artık", "oldu", "tv", "rağmen", "biri", "olarak", "önce", "veya", "olun", "için", "hala", "kim", "belki", "son", "başka", "gerekiyor", "asla", "sanki",
8  pozitif_kelime_listesi <- c("mutlu", "güzel", "harika", "umut", "eşit", "kesinlikle", "iyiki", "sevgi", "başarı", "mutlu", "iyi", "mükemmel", "şahane", "teşekkürler",
9  "teşekkür", "tebrik" )
10 negatif_kelime_listesi <- c("üzgün", "cinayetleri", "şikayet", "ağladım", "cezalar", "cinayeti", "yok", "korkunç", "utanıyorum", "suçlu", "taciz", "öldürülen", "kötü", "çirkin", "nefret",
11 "üzgünüm", "suç", "dayak", "zarar", "maalesef", "şiddetin", "acı", "maalesef", "yazıklar", "kaygı", "hüzün", "başarısız", "yanlış",
12 "utanmak", "berbat", "cinayet", "sabır", "zor", "olum", "şiddet", "şiddet", "şiddet", "korkuyorum", "üzüldüm", "şiddeti", "yazık")
13 kelime_frekans_duzen_uygu <- kelime_frekans_duzenli %>% slice_max(order_by = n, n = 200) # En sık geçen 200 kelimeyi seçme
14 kelime_frekans_duzen_uygu <- yorumlar %>% # Kelime frekans tablosu oluşturma ve stop kelimeleri çıkarma
15 unnest_tokens(word, comment) %>% # Yorumları kelimelere ayır
16 count(word, sort = TRUE) %>% # Kelime frekansı hesaplama
17 filter(!word %in% turkce_dur_frekans) %>% # Stop kelimelerini çıkardık
18 mutate(
19   duygu = case_when(
20     word %in% pozitif_kelime_listesi ~ "Pozitif",
21     word %in% negatif_kelime_listesi ~ "Negatif",
22     TRUE ~ "Nötr"
23   )
24 )
25 )
```

(Şekil 11: Polarite için kelime çıkarma)

Burada da gerekli düzenlemeleri yaparak görselleştirmeye ağırlık verilmiştir.

```
26 # Pozitif ve Negatif kelimelerden en çok geçen 10 kelime
27 pozitif_kelime_frekans <- kelime_frekans_duzen_uygu %>%
28 filter(duygu == "Pozitif") %>%
29 slice_max(order_by = n, n = 10) # 10 pozitif kelime
30
31 negatif_kelime_frekans <- kelime_frekans_duzen_uygu %>%
32 filter(duygu == "Negatif") %>%
33 slice_max(order_by = n, n = 10) # 10 negatif kelime
34
35 # Pozitif ve Negatif kelimeleri birleştir
36 kelime_frekans_toplam <- bind_rows(pozitif_kelime_frekans, negatif_kelime_frekans)
37
38 # Dikey cubuk grafik oluşturma
39 ggplot(kelime_frekans_toplam, aes(x = reorder(word, n), y = n, fill = duygu)) +
40   geom_bar(stat = "identity", show.legend = FALSE, width = 0.7) + # Cubuk genişliği ve gösterim
41   coord_cartesian(ylim = c(0, max(kelime_frekans_toplam$n) * 1.1)) + # Y eksenini artırma
42   scale_fill_manual(values = c("Pozitif" = "#4CAF50", "Negatif" = "#F44336")) + # Renkleri belirleme
43   labs(title = "Pozitif ve Negatif Kelimeler (EN SIK GEÇEN 10)",
44        x = "Kelime",
45        y = "Frekans") +
46   theme_minimal() + # Minimal tema
47   theme(
48     plot.title = element_text(hjust = 0.5, size = 16, face = "bold", color = "#333333"), # Başlık
49     axis.text.x = element_text(angle = 45, hjust = 1, size = 12, color = "#555555"), # x eksen etiketleri
50     axis.text.y = element_text(size = 12, color = "#555555"), # Y eksen etiketleri
51     axis.title = element_text(size = 14, color = "#333333"), # Eksende başlıklar
52     panel.grid.major = element_line(color = "#e0e0e0"), # Ana grid çizgileri
53     panel.grid.minor = element_blank() # Küçük grid çizgilerini kaldırma
54   ) +
55   theme(axis.title.x = element_text(margin = margin(t = 10)), # X eksen başlık
56         axis.title.y = element_text(margin = margin(r = 10))) # Y eksen başlık
```

(Şekil 14: Duygu Analizi Sonucunda Pozitif ve Negatif Kelime Sonuçları)

3.BULGULAR

3.1 Kelime Bulutu

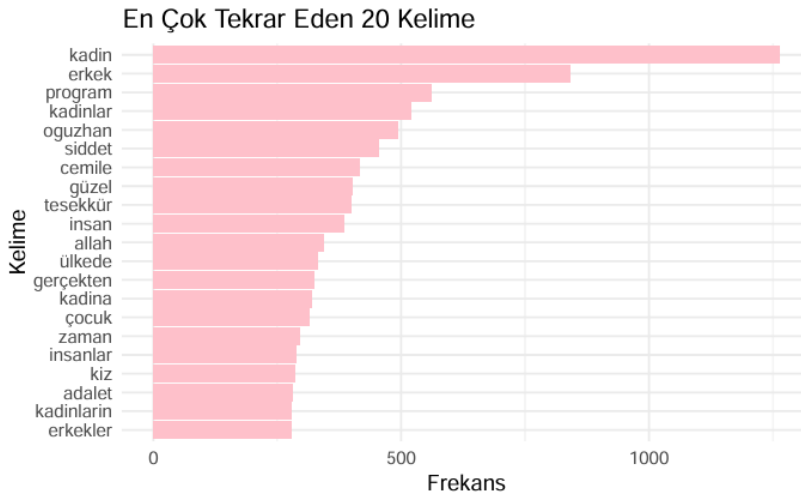
Elde ettiğim kelime bulutuyla birlikte analizini yaptığım video da eş değer kelimeler çıktığını söyleyebiliriz. Çıkan kelimelerin 1244 en çok yazılarak Kadın kelimesi olmuştur. Ardından 833 kere Erkek kelimesi, Program , Kadınlar, Oğuzhan ve Şiddet ardından gelerek videonun amacına uygun olduğunu ve insanların da videonun içeriğine uygun şekilde düşüncelerini belirttiğini anlamış oluyoruz.



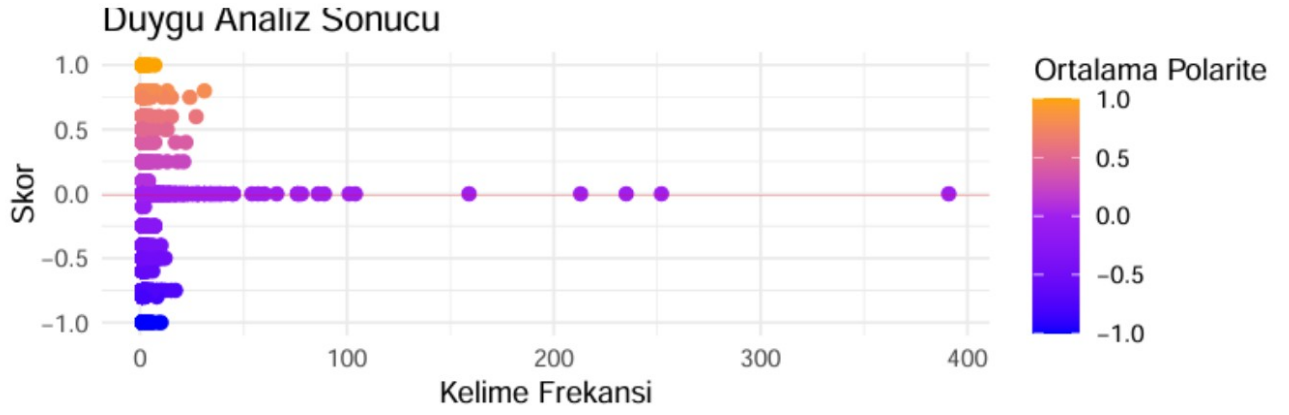
(Şekil 5 Sonuç: Kelime Bulutu)

3.2 Frekans Tablosu

(Şekil 6: En çok tekrar eden 20 kelime) burada yapmış olduğum işlemi sonucunda en çok tekrar eden kelimeler “kadın, erkek, program ve Oğuzhan” diyebiliriz. En çok kullanılan ve arada büyük bir uçurum oluşturan “kadın” kelimesi kelime bulutunda da öne çıktığını gördük ve analiz sonucunda da en çok tekrar eden kelimeler ile kelime bulutumuzun doğruluğunu desteklenmiş oldu.



(Şekil 6 Sonuç: En çok Tekrar eden kelime)

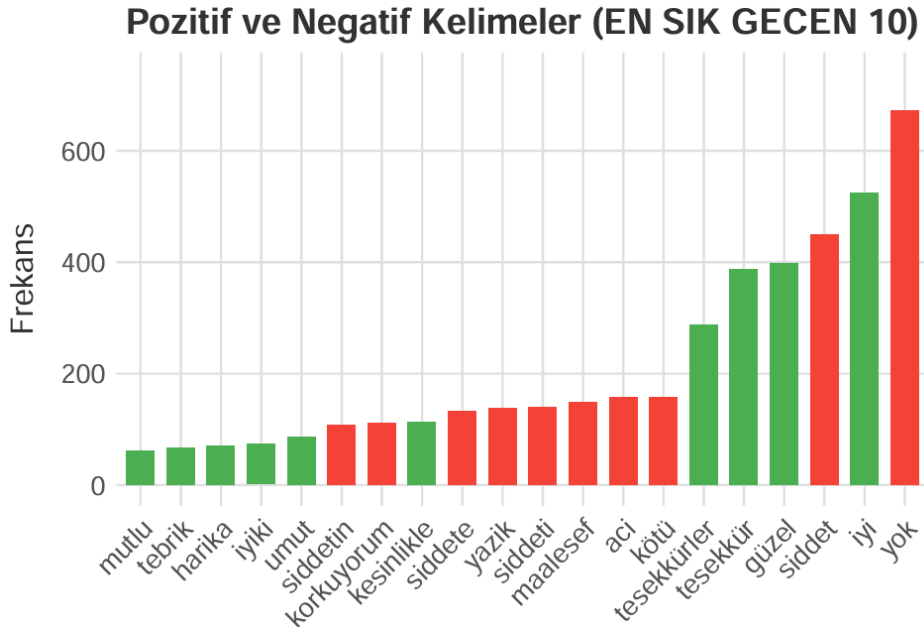


(Şekil 8 Sonuç: Kelime Frekansı)

mean	se.mean	CI.mean.0.95	var	std.dev	coef.var
0.0007980846	0.003899082	0.007643291	0.114295	0.3380754	42360.84

(Şekil 9 Sonuç: Kelime Matematiksel)

Aşağıda yaptığım “Pozitif Negatif Kelimeler” grafiğinde Türkçe kelimeler üzerinden en çok geçen duygu içerikli kelimelerin kullanım sıklığını göstermek istedim. Bunun sonucunda “yok, iyi, şiddet, güzel, teşekkürler...” gibi kelimelerin önde çıkması insanların video hakkındaki yorumlarını ve video içeriğini özetlediğini söyleriz. Bunun sonucunda insanlar ne kadar bu video içeriğinde geçenlere negatif duygularını belli etse de bir o kadar da videonun yapıldığı için teşekkürlerini belirttiğini söyleyebiliriz.



(Şekil 13 Sonuç: Duygu Analizi Sonucunda Pozitif ve Negatif Kelime Sonuçları)

4. SONUÇ VE TARTIŞMA

Bu analizde Youtube Apı sayesinde çektiğimiz verilerle kullanıcıların **Mevzular Açık Mikrofon | 6284 Özel** videosu yorumlarını analiz etme fırsatı elde ettik. Bu sayede kullanıcıların video için düşüncelerini duygularını veri madenciliği dalı olan metin madenciliğini kullanarak analiz edilmiştir. Bu videoda asıl amacım videonun izlenmeden bir insanda nasıl bir etki yaratacağını tespit etmek ve yorum yapan kullanıcıların duygularını anlayabilmektir. Bu sonuçları “kelime bulutu, frekans tablosu, kelime ağı grafiği ve duygu analizi” gibi yöntemleri kullanarak kullanıcı yorumlarını detaylı bir şekilde incelenmiştir.

Kelime bulutunu kullanarak yorumların içinde en çok kullanılan 50 kelime ve en çok kullanılan 20 kelime frekans grafiği görselleştirilmesinden yararlanılarak “kadın”, “erkek” gibi kelimelere vurgu yapıldığını böylelikle videonun temel konusuna değinildiğini anlamış bulunmaktayız. Ayrıca “program”, “şiddet” ve “Oğuzhan” gibi kelimeler de sıkça kullanıldı bu durum video içeriğiyle doğrudan ilişkili bir tartışma zemini oluşturduğunu anlayabiliriz.

Kelime ağı grafiğinde ise kelimelerin birbirleriyle olan ilişkileri incelendi ve kullanıcı yorumları ışığında video içeriği ile uyumlu olduğunu ayrıca kelime ağlarının belirli konu başlıkları altında toplanarak video içerisindeki önemli olayları vurguladığını söyleyebiliriz bu programın hedeflenen amacına uygun olduğunu gösterir. Bu kelimeler programa teşekkür, İstanbul sözleşmesi, erkekleri koruma ve konuk oyuncuların isimleri ve anlatılan hikayelerin bağlantıları oluşturularak videonun temel amacını ve içeriğini destekleyendir.

Duygu analizi sonuçlarına baktığımızda kullanıcıların yorumları genel olarak nötr bir duygu içerisinde olduğunu görülmekte bu sonuca “Mean” değerinin 0’a yakın çıkması, kelimeler tek başına ele alındığında anlam ifade etmede sınırlı kalıyor. Bu yüzden 0’a yakın çıktı diyebiliriz. Diğer bir analiz de kelimeleri pozitif negatif ayırarak yaptım burada da duygu durumların hem olumsuz hem de olumlu kelimelerin ne sıklıkla geçtiğini tespit etmekte. Bu iki analiz sonucunda video izleyicilere karışık duygular yarattığını ama videonun içeriğine değer verildiğini de göstermektedir.

Özetle, yapılan analizler sonucunda video izleyiciler üzerinde anlamlı bir etki yarattığını ve kullanıcı yorumlarının video içeriğiyle örtüştüğünü söyleyebiliriz. İnsanlar videoya dair duygularını düşüncelerini genel olarak olumlu ve olumsuz şeklinde ifade etmiş ayrıca video karşı teşekkürlerini de söylemişler. Böylelikle analizlerin tutarlılığını ve video içeriğinin amacına ulaştığını açıkça ortaya koymaktadır.

5. KAYNAKÇA

[Mevzular Açık Mikrofon | 6284 Özel](#)

[R ekosisteminde metin madenciliği nasıl yapılır? - newslabtürkiye](#)

[ChatGPT](#)

<https://dergipark.org.tr/tr/download/article-file/2317288>

[Veri Biliminde API kullanımı. YouTube API kullanarak Video Verilerini... | by Seherkumsar | Medium](#)

[R Programlama Dili ile Türkçe Tweetlerde Duygu Analizi \(Sentiment Analysis\) | by Sezen AKÇA | Medium](#)

[Google Çeviri \(Translate\) API'sinin Python ile Kullanımı ve Pandas Uygulaması | by Yiğit Şener | Medium](#)

[Yusuf Gül Veri Madenciliği.pdf](#)