SOSYAL MEDYADAN VERİ ÇEKME VE VERİ ANALİZİ

Cenk CAMKIRAN 150202015, Ferhat ÇAKIR 150202087, Yusuf Can AVCI 150202020,

Barış MENEKŞE 140202122

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

mailto:[cenkcamkiran@gmail.com](mailto:cenkcamkiran@gmail.com), mailto:[ferhatcakir2008@hotmail.com](mailto:ferhatcakir2008@hotmail.com) [yusufcanavci53@gmail.com](mailto:yusufcanavci53@gmail.com) <mailto:barismenekse@msn.com>



Özet

Dünya’da, insanlar zamanlarının büyük çoğunluğunu sosyal medyada gezinerek harcarlar. Haliyle de insanlar sosyal medyalarda gezinirken arkalarında iz bırakırlar. Bu izler, insanların farklı konular hakkındaki düşüncelerinden tutun da, insanların sosyal medya üzerinde takip ettiği, satın aldığı, tercih ettiği ürün ve markalar olabiliyor. Şirketler ve kurumlar, ticari amaçlı API’leri kullanarak bu izleri analiz eder ve sonuçlarını değerlendirir. Bu sonuçlara göre şirketler, mali yapılarını ve genel planlarını bu izlere göre düzenleyebilir ve ayrıca bu izlerden elde ettikleri verileri değerlendirerek, müşterilerini daha da memnun etme amacı güdebilirler. Bu tip durumlarda, Veri Madenciliği, Veri Analizi gibi bilgi alanlarına ihtiyaç duyuluyor ve her geçen gün bu bilgi alanlarının önemi kat ve kat artıyor.

## Giriş

## 1.1.1. Sosyal Medya Nedir?

Internet kullanıcılarının birbirleriyle tanışması, irtibata geçmesi, içerik paylaşımında bulunması, tartışma ortamı oluşturması ve ortak ilgi alanlarındaki kişilerin bir araya gelebileceği gruplar oluşturması amacıyla oluşturulan internet siteleri sosyal medya olarak tanımlanıyor.

## 1.1.2. Sosyal Medya Veri Analizi

Etrafta milyonlarca veri var. Bir bakıma milyonlarca çöp veri vardır. İşe yarar veriyi, bu çöp veriden çıkarmak oldukça önemlidir.

İnsanların sosyal ağlarda ürettiği devasa veriyi, devamlı takip etmek, izlemek, dinlemek ve bundan işe yarar veriler üretmek modern bir veri madenciliği örneğidir.

### **Çeşitli Sosyal Medyalar**

Dünyada popüler olan Sosyal Medyalara örnek verecek olursak, Twitter, Instagram, Linkedin ve Facebook. Ve bu raporumuzda, 3 adet Sosyal Medyayı bahsediyor olacağız.

## Sosyal Medyadan Veri Nasıl Çekilir?

### **Twitter’dan Veri Nasıl Çekilir?**

Twitter hesabımız yoksa <https://twitter.com/> adresine tıklayarak bir Twitter hesabı oluşturuyoruz. Twitter Developer Uygulamalarını (API) kullanabilmek için gerçek isminizle Twitter’a kaydolmanızı öneririz. Çünkü Twitter genellikle sahte hesaplar için Twitter Developer Uygulaması kullanılamamaktadır.

<https://developer.twitter.com/> adresine tıklayarak Twitter hesabımızla giriş yapıyoruz.

İlk başta Twitter size bazı sorular soracak. Bu soruları tam ve doğru bir şekilde yanıtlamalısınız. Aksi takdirde yaptığınız başvuru onaylanmaz.

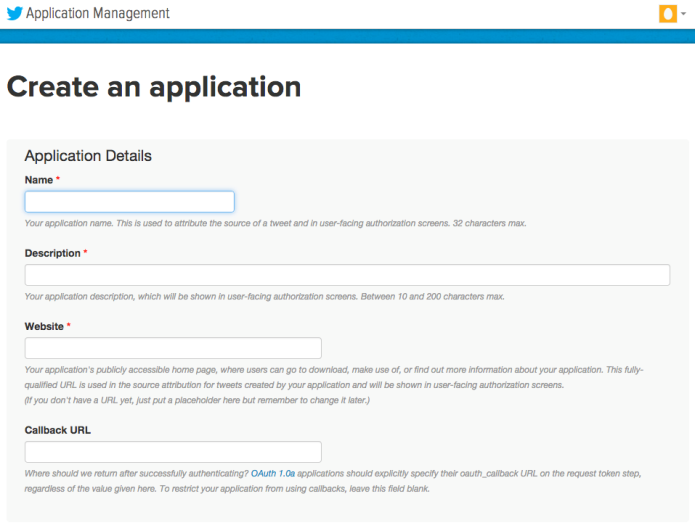
Twitter başvurunuzu onayladığına dair kayıt olduğunuz e-posta adresinize mail yollayacaktır.

### **Twitter API Key ve Token**

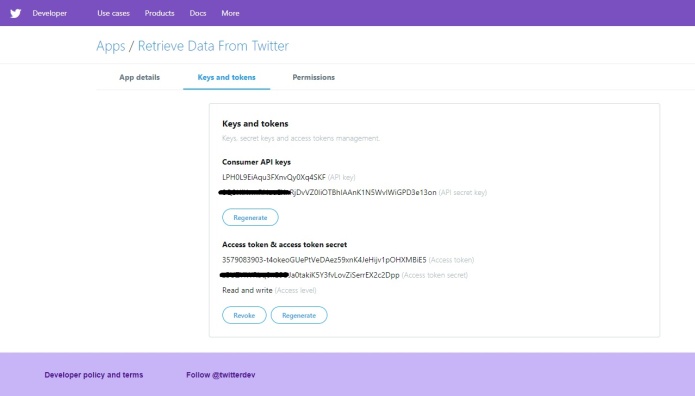
<https://developer.twitter.com/> sitesine tekrar giriş yaptıktan sonra karşımıza Create Application ekranı çıkmaktadır.

Create Application butonuna tıkladıktan sonra karşımıza Uygulamamızın adını, web sitesini ve uygulamamızla ilgili açıklamaları yazdıktan sonra Uygulamamızı başarıyla oluşturabiliriz.

Uygulamamızı oluşturduktan sonra, Twitter her kullanıcıya özel token (key, anahtar) vermektedir. Bu tokenleri veri çekme esnasında kod içerisinde kullanacağız.



Şekil 1: Twitter Developer Application Oluşturma Ekranı



Şekil 2: Twitter API Key ve Token

### **Tweepy Kütüphanesi**

Tweepy Modülü Python'da Twitter için geliştirmeler yapmaya yarayan bir kütüphanedir. Twitter, API'ı kullanılarak yazılmıştır. Gereken birçok işlemi, (Tweet atmak, timeline'ı okumak, takipçiler, takip edilenler ) bu kütüphane üzerinden yapabilirsiniz. Bu modülü [Github](https://github.com/tweepy/tweepy)'tan indirip manuel olarak kurabilirsiniz.

Python’da öncelikle **tweepy** kütüphanesini import etmemiz gerekiyor.

**import tweepy**

Giriş paragrafında kütüphanemizi dâhil etmiştik, şimdi uygulamamıza giriş yapacağız. Bunun key’lerimizi kullanacağız [2].

**consumer\_key=""  
consumer\_secret=""  
access\_token=""  
access\_token\_secret = ""**

[2]

OAuthHandler() fonksiyonu ile giriş yapıyoruz ve bunu auth değişkenine bağlıyoruz. Giriş yaptıktan sonrada uygulamamızın keylerini tanımlıyoruz. Girişi başarılı bir şekilde tamamladıktan sonra API'ımızı bir değişkene bağlıyoruz ve buradan sonrasına onunla devam ediyoruz [2].

**auth=tweepy.OAuthHandler(consumer\_key, consumer\_secret)**

**auth.set\_access\_token(access\_token, access\_token\_secret)**

**api=tweepy.API(auth)**

**api = tweepy.API(auth)**

[2]

Buradan sonrası tamamen api değişkenimizin fonksiyonlarına kalmış durumda. Örneğin bir tweet atmak için update\_status() fonksiyonunu kullanıyoruz [2].

**api.update\_status('HelloPythonCentral! #pythoncentral')**

[2]

Ana sayfada paylaşılan tweetleri almak için home\_timeline() fonksiyonunu kullanıyoruz. Bu fonksiyon ana sayfadaki son 20 tweeti bir liste içine atarak bize veriyor. For döngüsüyle de bu listeyi ekrana yazdırabiliyoruz [2].

**public\_tweets = api.home\_timeline()**

**for tweet in public\_tweets:**

**print(tweet.text)**

[2]

get\_user fonksiyonuna Screen Name’i (@example gibi) girilen kullanıcının id’sini user adlı değişkene atıyoruz. Bu user değişkeni, kullanıcıyla alakalı bilgileri tutmaktadır. Kullanıcı adı, id, Twitter’a hangi tarihte kayıt olduğu, takipçi sayısı, takip edilenlerin sayısı, description bölümü gibi birçok bilgi yer almaktadır. Bu değişkenin id’sini ekrana bastırmak için aşağıdaki kodun kullanılması gereklidir [2].

**user = api.get\_user('Twitter')**

**print(user.id)**

[2]

ID’si bulunan kullanıcının, user\_timeline fonksiyonuna argüman olarak girildiğinde kendi sayfasında daha önceden attığı son 20 tweeti for döngüsüyle ekrana bastırmak için aşağıdaki kodun kullanılması gereklidir [2].

**sonuc = api.user\_timeline(783214)**

**for tweet in sonuc:**

**print(tweet)**

[2]

Kullanıcının, diğer kullanıcılar tarafından retweet edilen en son 20 tweeti for döngüsüyle ekrana bastırmak için aşağıdaki kodun kullanılması gereklidir [2].

**retweets = api.retweets\_of\_me();**

**for tweet in retweets:**

**print(tweet.text)**

[2]

Kullanıcının screen name’ı follower fonksiyonuna argüman olarak girilerek kullanıcının takipçilerinin bilgilerini ekrana bastırmak için aşağıdaki kodu kullanıyoruz [2].

**followers = api.followers('Twitter')**

**print(followers)**

[2]

ID’si bulunan kullanıcının attığı tweet sayısını ekrana yazdırmak için aşağıdaki kod parçacığını kullanıyoruz [2].

**kullanıcı = api.get\_user(783214)**

**print(kullanıcı.statuses\_count)**

[2]

Twitter kullanıcıları içerisinde adında ‘xxxx’ kelimesi geçen 1000 kullanıcıyı bulmak için aşağıdaki kod kullanıyoruz. Page argümanıyla belirlenen kullanıcıları sayfalara atıyoruz. Sayfalar arası geçişi bu şekilde sağlıyoruz. Örnek olarak aşağıdaki kod parçacığında 2. Sayfada adında ‘giraffe’ bulunan sonuçları ekrana bastırıyoruz [2].

**result = api.search\_users('giraffe', page = 2)**

**print(result)**

[2]

Screen name’i girilen kullanıcının beğendiği tüm tweetleri bulup for döngüsüyle ekrana bastırmak için aşağıdaki kod parçacığını kullanıyoruz. Örnek olarak aşağıdaki kod parçacığında ‘ShalamanDay23’ adlı kullanıcının beğendiği tweetlerin 1.sayfasını gösteriyoruz [2].

**favs = api.favorites('ShalamanDay23', page=1)**

**for tweet in favs:**

**print(tweet.text)**

[2]

ID’si bulunan tweeti argüman içerisinde yazarak yazdığımız tweeti beğenmemizi sağlayan kod parçacığı aşağıdaki şekilde kullanıyoruz [2].

**api.create\_favorite(1058040614959439874)**

[2]

ID’si bulunan tweeti argüman içerisinde yazarak yazdığımız tweeti beğenmemizi geri almayı sağlayan kod parçacığı aşağıdaki şekilde kullanıyoruz [2].

**api.destroy\_favorite(1058040614959439874)** [2]

Her ülkede, trend olan konuların bulunduğu ülkeleri ve ülkelerin illerini for döngüsüyle ekrana yazdıran kod parçacığı aşağıdaki gibidir [2].

**trends = api.trends\_available()**

**'''for tweet in trends:**

**print(tweet)**

[2]

WOEID’si alınan ülkelerdeki trend olan ilk 10 konuyu eğer bilgileri kullanılabilir durumdaysa bilgileriyle beraber for döngüsüyle ekrana yazdıran kod parçacığı aşağıdaki gibidir [2].

**place = api.trends\_place(2972)**

**'''for trend in place:**

**print(trend)**

[2]

Screen name ‘i girilen kullanıcıyı takip ettiren kod parçacığı aşağıdaki gibidir [2].

**api.create\_friendship('XTheWitcherX')** [2]

Screen name ‘i girilen kullanıcıyı takipten çıkaran kod parçacığı aşağıdaki gibidir [2].

**api.destroy\_friendship('XTheWitcherX')** [2]

Screen name’i girilen kullanıcının takip ettiği kullanıcıların id’lerini ekrana yazdıran kod parçacığı aşağıdaki gibidir [2].

**friends = api.friends\_ids('ShalamanDay23')**

**print(friends)**

[2]

Screen name’i girilen kullanıcıyı takip eden kullanıcıların id’lerini ekrana yazdıran kod parçacığı aşağıdaki gibidir [2].

**friends = api.followers\_ids('ShalamanDay23')**

**print(friends)**

[2]

## Sosyal Medya Veri Analizi

Sosyal medya hesaplarını temel düzeyde bilgisayar ve internet kullanabilen herkes yönetebilir. Önemli olan hesaplarınızın verimli ve amaca uygun bir şekilde yönetilmesidir aslında. Peki bunu nasıl başarabiliriz [14]? Uzaktan bakınca bu çok zor ve karmaşık görünüyor olabilir. Ama gerekli işlemler yapıldığında çok karmaşık olan bu işi, basite indirgeyerek çözmüş oluruz.

Sosyal medya madenciliği (Mining the Social Web, Data Mining) ülkemizde ne kadar çok popüler olmasa da dünya üzerinde birçok kişi tarafından ilgiyle takip ediliyor. Twitter, hızlı bir şekilde mesajlaşma platformu sağlamasının yanında program geliştirenler için bir ara yüze de sahip. Bu ara yüz sayesinde, profili halka açık olan kişilerin paylaştığı durumları takip etmenin yani sıra kimin kaç takipçisi var, bir post ne kadar paylaşılmış gibi detaylı analiz yapmak da mümkündür [15].

## 1.2.2. Twitter Veri Analizi

## Python’da öncelikle indirdiğimiz kütüphaneyi import etmemiz gerekiyor. Import işlemi kütüphanedeki önceden tanımlı fonksiyonları kullanmamıza olanak verir. Haliyle bir kütüphaneyi import ettiğimizde, o kütüphanedeki hazır fonksiyonları kendimiz define etmeden kullanabiliyoruz. Import kısmını kendi kodumuzdan direkt olarak kopyalayacağız.

## 1.2.2.1.Veri Analizinde Kullanılan Kütüphaneler

**NumPy**, Python’da bilimsel hesaplamanın temel paketidir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler (maskelenmiş diziler ve matrisler gibi) ve dizilerdeki hızlı işlemler için matematiksel, mantıksal, şekil işleme, sıralama, seçme, g / Ç içeren bir dizi yordam sunan bir Python kitaplığıdır , Ayrık Fourier dönüşümü, temel doğrusal cebir, temel istatistik işlemler, rassal simülasyon ve çok daha fazlasını numpy kütüphanesi sunar [11].

**Pandas,** python programlama dili için **yüksek performanslı**, kullanımı kolay **veri yapıları** ve **veri analiz araçları** sağlayan açık kaynaklı bir BSD lisanslı kütüphanedir. **Csv** ve **text** dosyalarını açmaya ve  içerisinde bulunan verileri okuyarak istenen sonuca  kolayca ulaşmak için kullanılmaktadır. Yani bir excel dosyasını açarak içerisinde bulunan bir sütunu veya satırı seçerek işlemleri yapabiliriz. Numpy kütüphanesinde yapılan verilerin şekillendirilmesi daha detaylı bir biçimde kullanılabilmektedir [12].

**Matplotlib**, grafik çizim paketi Python’la bilimsel programlamanın en önemli araçlarından birisidir. Çok kuvvetli bir paket olan **Matplotlib** ile verileri etkileşimli olarak görselleştirebilir, yayınlamaya uygun yüksek kalitede çıktılar hazırlayabiliriz. Hem iki boyutlu hem de üç boyutlu grafikler üretilebilir [13].

**TextBlob**, metin verilerini işlemek için bir Python (2 ve 3) kütüphanesidir. Konuşma bölümü etiketleme, isim öbek çıkarma, duygu analizi, sınıflandırma, çeviri ve daha fazlası gibi ortak doğal dil işleme (NLP) görevlerini gerçekleştirebilmek için basit bir API sağlar [14].

Python’da öncelikle **numpy, pandas, matplotlib** ve **textblob** kütüphanesini import etmemiz gerekiyor.

**import tweepy**

**import numpy as np**

**import pandas as pd**

**from tkinter import \***

**from time import sleep**

**from datetime import datetime**

**from textblob import TextBlob**

**import matplotlib.pyplot as plt**

**import matplotlib**

## Sentiment Analizi

## Duygu analizi, bir yazı parçasının olumlu, olumsuz veya nötr olup olmadığını belirleme yöntemidir. Bir konuşmacının fikrini veya tutumunu inceleyen fikir madenciliği olarak da bilinir [4].

## Bu teknolojinin ortak kullanım noktası, insanların belirli bir konu hakkında nasıl hissettiklerini keşfetmektir [4].

Fikir madenciliği, doğal dil işleme, metin analizi, bilişimsel dilbilim ve biyometrelerin, duygusal durumları ve öznel bilgileri sistematik olarak tanımlamak, ölçmek ve incelemek için kullanılmasını ifade eder [4].

Genel olarak ifade edilen duygu analizi, bir konuşmacının, yazarın veya başka bir konunun bir konu, genel bağlamsal kutupluluk veya bir belgeye, etkileşime veya olaya olan duygusal tepkisine ilişkin tutumunu belirlemeyi amaçlamaktadır. Bu tutum, bir yargılama ya da değerlendirme ya da amaçlanan duygusal iletişim olabilir [4].

**import tweepy**

**from tkinter import \***

**from time import sleep**

**from datetime import datetime**

**from textblob import TextBlob**

**import matplotlib.pyplot as plt**

**import matplotlib**

Python’da öncelikle veriler üzerindeki analiz sonuçlarını grafik üzerinde ekranda göstermek için **matplotlib** kütüphanesini, sentiment (duygu) analizi için ise **TextBlob** kütüphanesini import etmemiz gerekiyor.

Arayüz için iki bileşen (etiket) kullanacağız: Biri arama için diğeri analiz edilecek örneklem büyüklüğü (sample size) veya tweet sayısı için. Ayrıca ‘submit’ butonuna da ihtiyacımız olacak, böylece butona tıklandığında ‘getData’ adlı fonksiyonumuzu çağırabilmiş olacağız [3].

**root = Tk()**

**label1 = Label(root, text="Search")  
E1 = Entry(root, bd =5)**

**label2 = Label(root, text="Sample Size")  
E2 = Entry(root, bd =5)**

**submit = Button(root, text ="Submit", command = getData)**

[3]

Bilgisayarın GUI'yi ekranda gösterebilmesi için bileşenlerimizi paketlememiz ve ardından root’u (kök ekran) döndürmemiz gerekir [3].

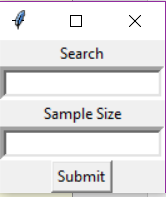
**label1.pack()  
E1.pack()**

**label2.pack()  
E2.pack()**

**submit.pack(side =BOTTOM)**

**root.mainloop()**

[3]



Şekil 3: GUI'den bir görüntü

Öncelikle, Textbox (kutucuklar) bileşenlerine girilen verileri almak için aşağıdaki 2 fonksiyonu yazmamız gereklidir [3].

**def getE1():  
 return E1.get()**

**def getE2():  
 return E2.get()**

[3]

Şimdi ‘getData’ fonksiyonunu kodlamaya hazırız. Şu andan itibaren, buraya kadar yazdığımız tüm kodlar bu fonksiyon içerisinde olmalı [3]:

**def getData():  
 #Code**

[3]

‘E1()’ ve ‘E2()’ fonksiyonlarını kod içerisinde çağırmamız gerekiyor. Bu 2 fonksiyon, Textbox (kutucuklar) bileşenlerine girilen tweet sayısı (sample size) ve anahtar kelimeyi alıp 2 ayrı değişkene atmaya yarıyor [3].

**getE1()  
 keyword = getE1()**

**getE2()  
 numberOfTweets = getE2()  
 numberOfTweets=int(numberOfTweets)**

[3]

Verilerimizi saklamak için listeleri (dizileri) kullanabiliriz. Bir liste, tweet'lerin kutupsallığı (veya duyarlılığı) için, diğeri ise tweet'lerin sayısı için kullanılmaktadır. (Böylece verileri sakladıktan sonra grafiklendirebiliriz) [3]

**polarity\_list = []  
numbers\_list = []  
number = 1**[3]

Varsayılan değer 0 olduğu için tweetlerin sayısının 1 olarak atanması gerekir. Şimdi tweet'leri baştan sona döngüyle gezip analiz etmeye başlayabiliriz. TextBlob kütüphanesi kullanarak, her tweetin duyarlılığını (polarite, sentiment) bulabilir ve bu duyarlılığı ‘polarity’ isimli değişkenimizin içinde tutabiliriz. Daha sonra bu değişkeni ‘polarity\_list’ isimli listemize ekleriz ve döngüyle gezilen tweet sayısını da ‘number\_list’ adlı listemize ekleyebiliriz [3].

**analysis = TextBlob(tweet.text)  
analysis = analysis.sentiment  
polarity = analysis.polarity polarity\_list.append(polarity) numbers\_list.append(number)  
number = number + 1**

[3]

Bir for döngüsü ve try deyimini kullanarak, aranan anahtar kelimeye ve tweet sayısına göre aşağıdaki kodu yazdık [3].

**for tweet in tweepy.Cursor(api.search, keyword, lang="en").items(numberOfTweets):  
 try:  
 analysis=TextBlob(tweet.text)  
 analysis=analysis.sentiment  
 polarity=analysis.polarity  
 polarity\_list.append(polarity)  
 numbers\_list.append(number)  
 number = number + 1**

**except tweepy.TweepError as e:  
 print(e.reason)**

**except StopIteration:  
 break**

[3]

Matplotlib kütüphanesi ile bir dağılım grafiği çizebilmek için önce y eksenini tanımlamamız gerekir [3].

**axes = plt.gca()  
axes.set\_ylim([-1, 2])**

[3]

Ve sonra verileri içeren listelerimizi scatter fonksiyonuna argüman olarak atarız [3].

**plt.scatter(numbers\_list, polarity\_list)**

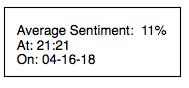
[3]

Topladığımız tweet'lerin genel duygularını göstermek için toplanan tüm Tweet'lerin ortalamasını hesaplıyoruz. Ayrıca, belirli bir zamanda duyarlılığı gösterdiğimizden, tarih ve saati görüntülemek istiyoruz [3].

**averagePolarity=(sum(polarity\_list))/(len(polarity\_list))  
averagePolarity="{0:.0f}%".format(averagePolarity \* 100)  
time=datetime.now().strftime("At: %H:%M\nOn: %m-%d-%y")**

**plt.text(0, 1.25, "Average Sentiment: " + str(averagePolarity) + "\n" + time, fontsize=12, bbox=dict(facecolor='none', edgecolor='black', boxstyle='square, pad = 1'))**

[3]



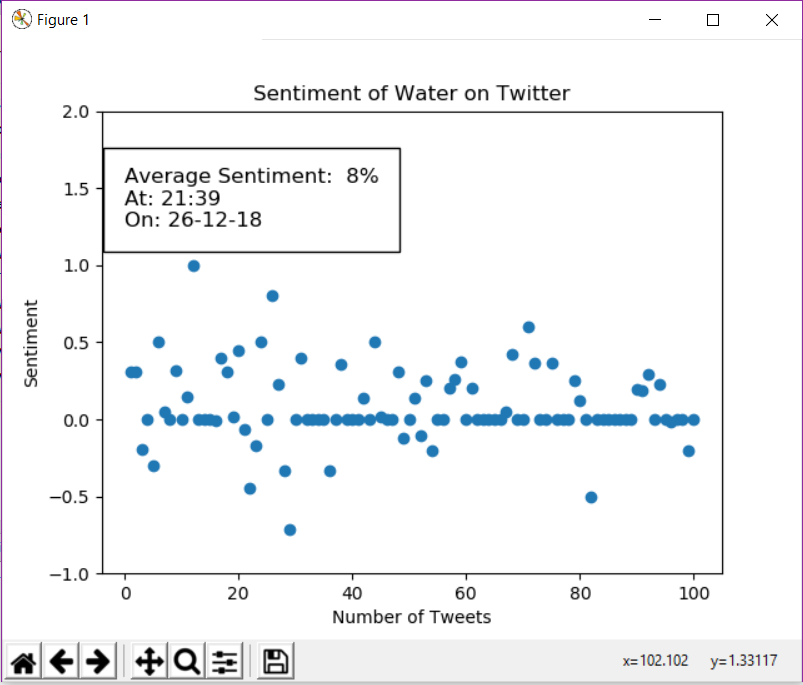
Şekil 4: Anahtar bilgiler bir kutucuk içerisinde gösterilmektedir.

Başlık için bunu kullanabiliriz [3].

**plt.title("Sentiment of " + keyword + " on Twitter") plt.xlabel("Number of Tweets")  
plt.ylabel("Sentiment")**

[3]

Ve son olarak grafiği görüntülemek için plot.show () fonksiyonunu kullanırız [3].



Şekil 5: Duygu analizinin grafiksel gösterimi

## Tweet’lerin Grafiksel Analizi

Belirli bir kullanıcıdan çekilen tweetleri kategorilere ayırmak için aşağıdaki kod parçası kullanılır. Kodu kısaca anlatmak gerekirse; belirli kullanıcıdan çekilen tweetler for döngüsüyle gezilir ve her bir tweetin içeriği, pandas kütüphanesinde bulunan DataFrame bileşeninin ‘tweets’ adlı sütünunun altına atanır. Ve yine belirli kullanıcıdan çekilen tweetler for döngüsüyle gezilir ve her bir tweetin id’si, tweetin uzunluğu, tweetin ne zaman, hangi tarihte atıldığı, tweetin hangi platformdan (Android, IOS, Bilgisayar) atıldığı, tweet kaç kişi tarafından beğenilmiş ve tweetin almış olduğu retweet sayısı gibi özellikler çıkarılır, bu özellikler numpy kütüphanesinde bulunan ‘array’ yani diziye (listeye) atadıktan sonra, bu özellikler DataFrame’de ‘tweets’, ‘id’ gibi sütunların altında toplanır. Ve böylelikle çekilen her bir tweetin özelliklerini kategorilere ayırmış oluruz [10].

**class TweetAnalyzer():  
def tweets\_to\_data\_frame(self, tweets):  
 df = pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['tweets'])  
  
 df['id'] = np.array([tweet.id for tweet in tweets])  
 df['len'] = np.array([len(tweet.text) for tweet in tweets])  
 df['date'] = np.array([tweet.created\_at for tweet in tweets])  
 df['source'] = np.array([tweet.source for tweet in tweets])  
 df['likes'] = np.array([tweet.favorite\_count for tweet in tweets])  
 df['retweets'] = np.array([tweet.retweet\_count for tweet in tweets])**

**return df**

[5]

Twitter kullanıcısı olarak ‘**realDonaldTrump**’ adlı kişiyi seçtik. Bu kişinin kendi sayfasından en son attığı 40 tweeti çekeceğiz [10].

**twitter\_client = TwitterClient()  
tweet\_analyzer = TweetAnalyzer()  
  
api=twitter\_client.get\_twitter\_client\_api()  
  
tweets=api.user\_timeline(screen\_name="realDonaldTrump", count=40)df=tweet\_analyzer.tweets\_to\_data\_frame(tweets)** [5]

Numpy kütüphanesi kullanarak, seçilmiş olan kullanıcının attığı son 40 tweetin ortalama uzunluğunu bulmak için aşağıdaki kod parçasını yazarız [10].**print(np.mean(df['len']))** [5]

Kullanıcının attığı son 40 tweet arasında, en çok beğenilmiş tweetinin beğenilme (like) sayısını ekrana yazdırmak için aşağıdaki kod paçasını kullanırız [10].

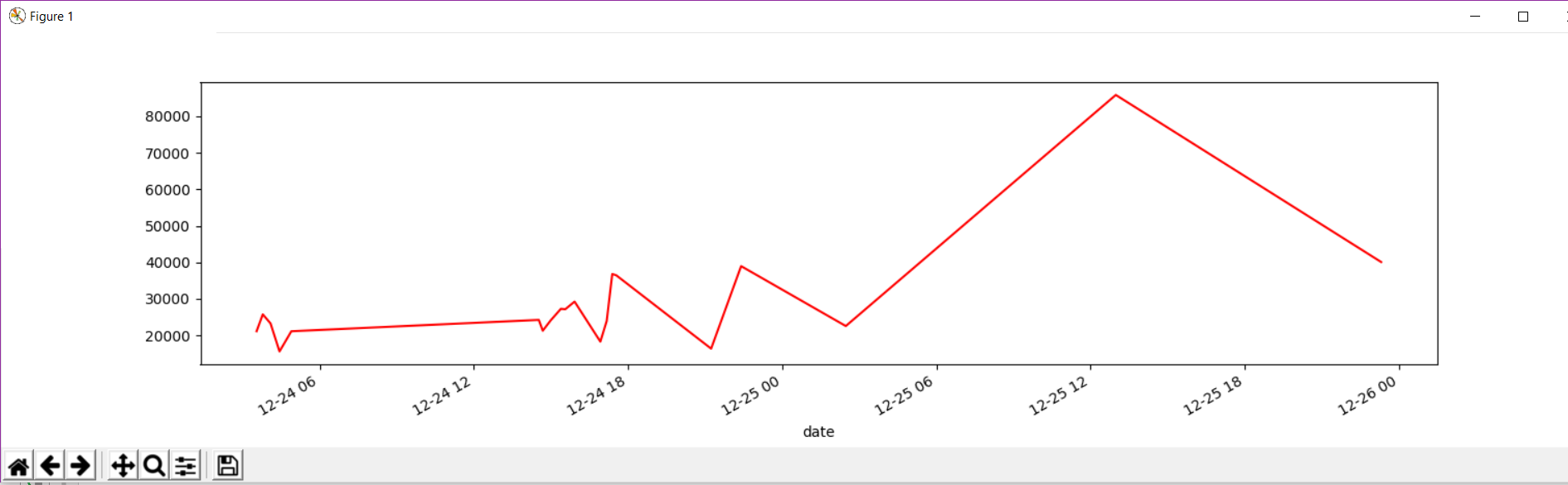
**print(np.max(df['likes']))** [5]

Kullanıcının attığı son 40 tweet arasında en çok retweet alan tweetinin retweet sayısını ekranda göstermek için aşağıdaki kod parçasını kullanırız [10].

**print(np.max(df['retweets']))** [5]

Kullanıcının belirli bir tarih aralığında attığı tweetlerin almış olduğu retweet sayısını matplotlib kütüphanesi ile grafik üzerinde, ekranda göstermek için aşağıdaki kod parçasını kullanırız [10].

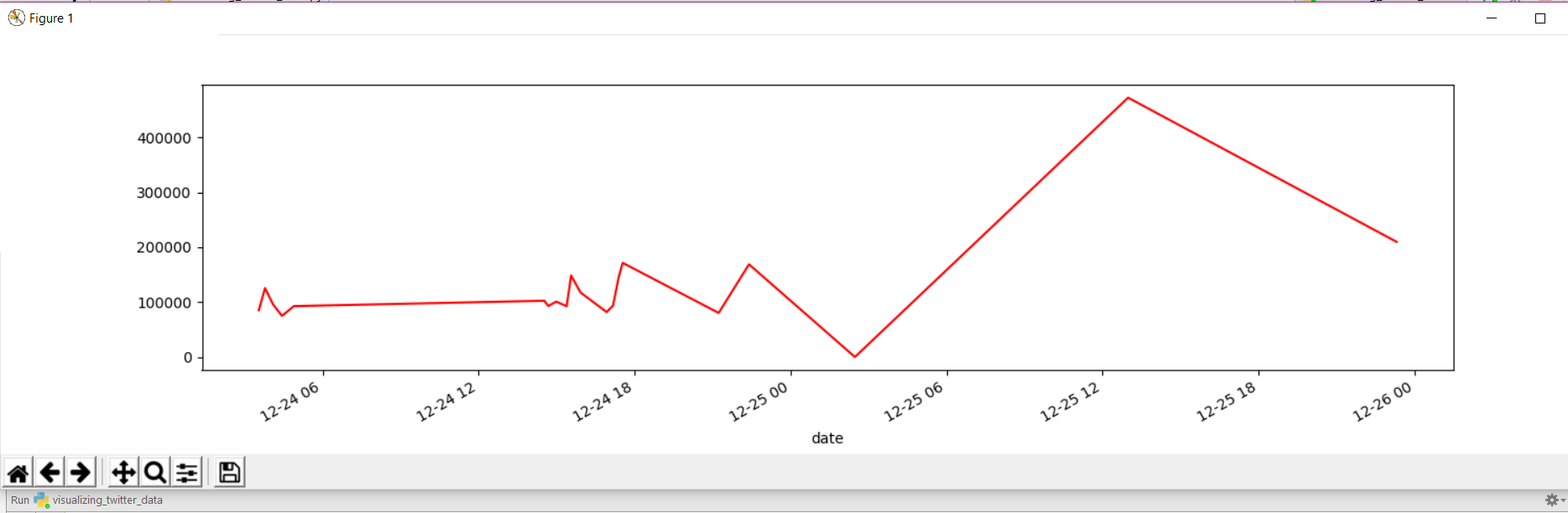
**time\_retweets=pd.Series(data=df['retweets'].values, index=df['date'])  
time\_retweets.plot(figsize=(16, 4), color='r')  
plt.show()**[5]



Şekil 6: Kullanıcının belirli bir tarih aralığında atmış olduğu tweetlerinin aldığı retweet sayısının grafiksel gösterimi

Kullanıcının belirli bir tarih aralığında attığı tweetlerin almış olduğu beğenilme (like) sayısını üzerinde matplotlib kütüphanesi ile grafik ekranda göstermek için aşağıdaki kod parçasını kullanırız [10].

**time\_favs=pd.Series(data=df['likes'].values, index=df['date'])  
time\_favs.plot(figsize=(16, 4), color='r')  
plt.show()**[5]

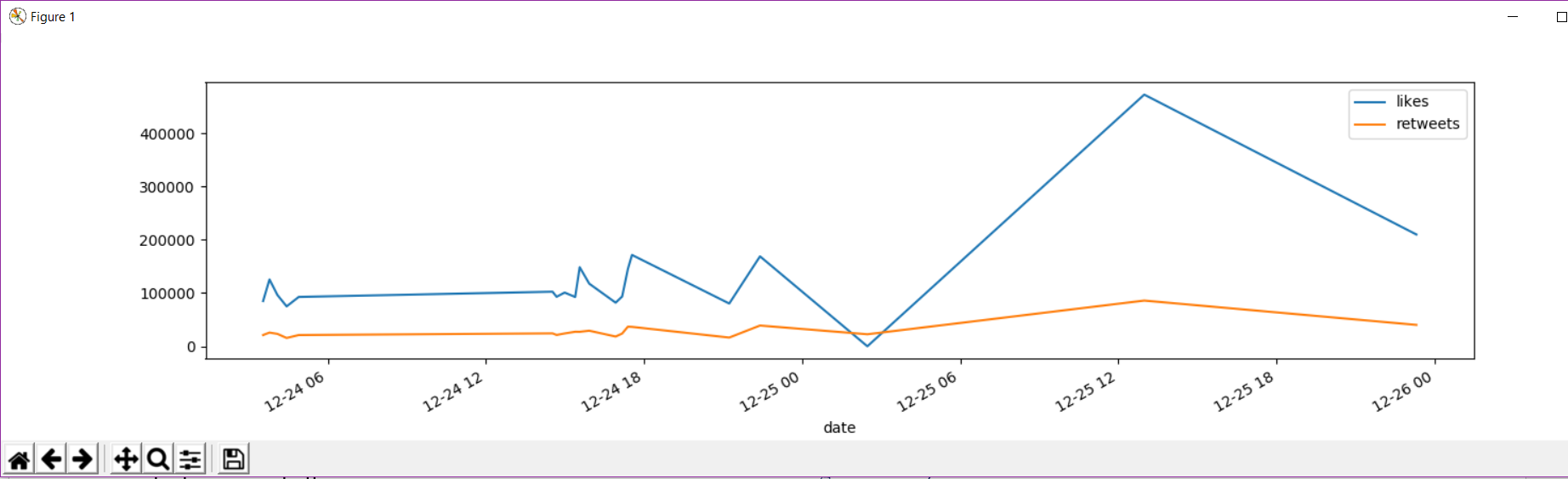


Şekil 7: Kullanıcının belirli bir tarih aralığında atmış olduğu tweetlerinin aldığı beğenilme sayısınının grafiksel gösterimi

Kullanıcının belirli bir tarih aralığında attığı tweetlerinin almış olduğu hem retweet sayısını hem de beğenilme (like) sayısını matplotlib kütüphanesi ile grafik üzerinde ekranda göstermek için aşağıdaki kod parçasını kullanırız [10].

**time\_likes=pd.Series(data=df['likes'].values, index=df['date'])  
 time\_likes.plot(figsize=(16, 4), label="likes", legend=True)**   
  
 **time\_retweets=pd.Series(data=df['retweets'].values, index=df['date'])  
 time\_retweets.plot(figsize=(16, 4), label="retweets", legend=True)  
plt.show()**

[5]



Şekil 8: Kullanıcının belirli bir tarih aralığında attığı tweetlerinin almış olduğu hem retweet sayısını hem de beğenilme sayısının grafik üzerinde gösterimi

## Gerçek-Zamanlı Tweet Çekme

Twitter üzerinde, belirlenen hashtag’te atılan tweetleri canlı (anlık, live) olarak çekmek için gerekli olan fonksiyonu aşağıdaki gibi yazarız [8].

**def stream\_tweets(self, fetched\_tweets\_filename, hash\_tag\_list):  
listener=StdOutListener(fetched\_tweets\_filename)  
 auth=OAuthHandler(twitter\_credentials.CONSUMER\_KEY, twitter\_credentials.CONSUMER\_SECRET)  
 auth.set\_access\_token(twitter\_credentials.ACCESS\_TOKEN, twitter\_credentials.ACCESS\_TOKEN\_SECRET)  
 stream = Stream(auth, listener)  
stream.filter(track=hash\_tag\_list)** [6]

hash\_tag\_list değişkeni ile birlikte hashtag listesi belirlenmiş ve bu hashtag’lerde atılan tweetleri txt dosyasına ve ekrana yazdırmak için aşağıdaki kodu kullanırız [8].

**if \_\_name\_\_ == '\_\_main\_\_':  
 hash\_tag\_list = ["donald trump", "hillary clinton", "barack obama", "bernie sanders"]  
 fetched\_tweets\_filename="tweets.txt"  
  
 twitter\_streamer=TwitterStreamer()  
 twitter\_streamer.stream\_tweets(fetched\_tweets\_filename, hash\_tag\_list)**

[6]



Şekil 9: Twitter üzerinde atılan tweetlerin canlı olarak çekildikten sonraki görüntüsü

## Tweet’lerin Bilgi Alanları

Belirli bir kullanıcıdan çekilen tweetleri kategorilere ayırmak için aşağıdaki kod parçası kullanılır. Kodu kısaca anlatmak gerekirse; belirli kullanıcıdan çekilen tweetler for döngüsüyle gezilir ve her bir tweetin içeriği, pandas kütüphanesinde bulunan DataFrame bileşeninin ‘tweets’ adlı sütünunun altına atanır. Ve yine belirli kullanıcıdan çekilen tweetler for döngüsüyle gezilir ve her bir tweetin id’si, tweetin uzunluğu, tweetin ne zaman, hangi tarihte atıldığı, tweetin hangi platformdan (Android, IOS, Bilgisayar) atıldığı, tweet kaç kişi tarafından beğenilmiş ve tweetin almış olduğu retweet sayısı gibi özellikler çıkarılır, bu özellikler numpy kütüphanesinde bulunan ‘array’ yani diziye (listeye) atadıktan sonra, bu özellikler DataFrame’de ‘tweets’, ‘id’ gibi sütunların altında toplanır. Ve böylelikle çekilen her bir tweetin özelliklerini kategorilere ayırmış oluruz [9].

**class TweetAnalyzer():  
def tweets\_to\_data\_frame(self, tweets):  
 df=pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['tweets'])  
  
 df['id'] = np.array([tweet.id for tweet in tweets])  
 df['len']=np.array([len(tweet.text) for tweet in tweets])  
 df['date']=np.array([tweet.created\_at for tweet in tweets])  
 df['source']=np.array([tweet.source for tweet in tweets])  
 df['likes']=np.array([tweet.favorite\_count for tweet in tweets])  
 df['retweets']=np.array([tweet.retweet\_count for tweet in tweets])**

**return df**

[7]

Twitter kullanıcısı olarak ‘**realDonaldTrump**’ adlı kişiyi seçtik. Bu kişinin kendi sayfasından en son attığı 40 tweeti çekeceğiz. Tweetlerin hangi bilgi alanlarına erişebildiğimizi öğrenmek için aşağıdaki kodu çalıştırırız [9].

**print(dir(tweets[0]))** [7]

Tweetlerin hangi tarihte atıldığı, tweetin almış olduğu like ve retweet sayısı, tweetin içeriği, ID’si, tweetin kim tarafından atıldığı gibi bilgi alanlarına erişmek mümkündür. Aşağıda, bu bilgiler liste şeklinde yazılmıştır [9].

**['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_',**

**'\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_',**

**'\_\_init\_subclass\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_',**

**'\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_',**

**'\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_api', '\_json', 'author', 'contributors',**

**'coordinates', 'created\_at', 'destroy', 'entities', 'favorite', 'favorite\_count',**

**'favorited', 'geo', 'id', 'id\_str', 'in\_reply\_to\_screen\_name', 'in\_reply\_to\_status\_id',**

**'in\_reply\_to\_status\_id\_str', 'in\_reply\_to\_user\_id', 'in\_reply\_to\_user\_id\_str', 'is\_quote\_status',**

**'lang', 'parse', 'parse\_list', 'place', 'possibly\_sensitive', 'retweet', 'retweet\_count', 'retweeted',**

**'retweets', 'source', 'source\_url', 'text', 'truncated', 'user']**

[7]

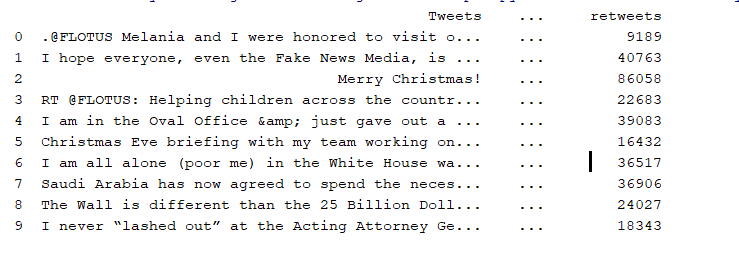
Aşağıdaki kod parçasıyla, ilgili tweetin almış olduğu retweet sayısını ekrana yazdırabiliriz. Diğer bilgi alanları için de aynı kurallar geçerlidir [9].

**print(tweets[0].retweet\_count)** [7]

Tweetlerin bir ya da birden fazla bilgi alanını ekrana tablo şeklinde bastırmak için aşağıdaki kod parçacığını kullanırız [9].

**df=tweet\_analyzer.tweets\_to\_data\_frame(tweets)**

**print(df.head(10))** [7]



Şekil 10: Örnek Çıktı

## Facebook Veri Analizi

**1.2.2.1. Facebook’tan Veri Nasıl Çekilir?**

Facebook’tan veri çekmek için aşağıdaki adımları uygularız.

Öncelikle API tokeni alınır.

1. [developers.facebook.com](http://developers.facebook.com) linkine gidilir ve bir hesap oluşturulur.

2. [developers.facebook.com/tools/explorer](https://developers.facebook.com/tools/explorer) linkine gidilir.

3. Uygulamalarım bölümüne gidilir. En üst sağ tarafın aşağı kısmına inilir ve ‘Yeni bir uygulama ekle’ seçilir. Gösterilecek isim ve kategori seçilir ve sonra ‘APP ID Oluştur’ a basılır.

4. Yeniden [developers.facebook.com/tools/explorer](https://developers.facebook.com/tools/explorer) linkine gidilir. Uygulamalarım kısmının altında “Graph API Tarayıcısı” görülür. Buraya tıkladıktan sonra aşağı inilir ve uygulama seçilir.

5. Sonra ‘Token Al’ seçilir. Buradan aşağı inilir, “Kullanıcı Giriş Tokeni Al’ seçilir. Menüden görülecek izinler seçilir ve sonra ‘Giriş Tokeni Al’ a tıklanır.

6. [developers.facebook.com/tools/accesstoken](https://developers.facebook.com/tools/accesstoken/) linkine gidilir. ‘Kullanıcı Tokeni’ ne göre ‘Debug’ lar seçilir.”Token Erişimini Uzat’ yapılır. Bu sayede token her 2 saatte bir süre bitimi yaşamaz [1].

## Facebook’tan Graph API ile Veri Çekme

İçinde poetry kelimesi geçen etkinlikleri bulan ve 10000 tanesini gösteren kod parçası aşağıdaki gibidir [1].

**graph=facebook.GraphAPI(access\_token=token, version = 2.7)**

**events=graph.request(‘/search?q=Poetry&type=event&limit=10000’)**

[1]

Bulunan bu etkinliklerin isimlerini listeleyerek bastıran kod parçası aşağıdaki gibidir [1].

**eventList = events[‘data’]** [1]

Bu etkinlikler arasından ilkinin ID’sini gösteren kod parçası aşağıdaki gibidir [1].

**eventid = eventList[1][‘id’]**

Bu etkinliğin ID’sinden bütün bilgileri alıp kullanmak için birkaç değişken belirlememize yarayan kod parçası aşağıdaki gibidir [1].

**event1=graph.get\_object(id=eventid,  
 fields=’attending\_count,can\_guests\_invite,category,cover,declined\_count,description,end\_time,guest\_list\_enabled,interested\_count,is\_canceled,is\_page\_owned,is\_viewer\_admin,maybe\_count,noreply\_count,owner,parent\_group,place,ticket\_uri,timezone,type,updated\_time’)**

**attenderscount=event1[‘attending\_count’]**

**declinerscount=event1[‘declined\_count’]**

**interestedcount=event1[‘interested\_count’]  
maybecount=event1[‘maybe\_count’]**

**noreplycount=event1[‘noreply\_count’]**

[1]

Etkinliğe ‘Katılanlar’ın sayısını alıp bunu json formatında gösteren kod parçası aşağıdaki gibidir [1].

**attenders=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/attending?access\_token="+token+”&limit=”+str(attenderscount))**

**attenders\_json=attenders.json()**[1]

Etkinliğin yöneticilerini gösteren kod parçası aşağıdaki gibidir [1].

**admins=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/admins?access\_token="+token)**

**admins\_json=admins.json()**

[1]

Bu etkinliğe ‘Katılmayı Reddeden’lerin sayısını gösteren kod parçası aşağıdaki gibidir [1].

**decliners=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/declining?access\_token="+token+”&limit=”+str(declinerscount))**

[1]

Bu etkinlikle ‘İlgilenen’lerin sayısını gösteren kod parçası aşağıdaki gibidir [1].

**interested=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/interested?access\_token="+token+”&limit=”+str(interestedcount))**

[1]

Bu etkinliğe katılmayı ‘Belki’ olarak cevaplayanların sayısını gösteren kod parçası aşağıdaki gibidir [1].

**maybe=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/maybe?access\_token="+token+”&limit=”+str(maybecount))**

[1]

Bu etkinliğe ‘Cevap Vermeyenler’in sayısını gösteren kod parçası aşağıdaki gibidir [1].

**noreply=requests.get(“**[**https://graph.facebook.com/v2.7/**](https://graph.facebook.com/v2.7/)**"+eventid+"/noreply?access\_token="+token+”&limit=”+str(noreplycount))**

[1]

* + 1. **Linkedin’den Veri Nasıl Çekilir?**

LinkedIn API Access Token almak için "developer.linkedin.com" adresine girip "Create Application" butonuna basılmalıdır.

Basıldıktan sonra gelecek ekrandaki form doldurulup gönderilmelidir. Başvuru için formu doldurduktan sonra authentication key'iniz ile bir sayfaya yönlendirileceksiniz. Geçerli bir web adresini authorize redireck URL olarak girin. Bu işlemden sonra access tokenınıza ulaşabilirsiniz.

LinkedIn authorization code'unuzun alınabilmesi için aşağıdaki parametreler ile <https://www.linkedin.com/oauth/v2/authorization> adresine istek göndermeniz gerekmektedir [16].

**response\_type: bu her zaman kod olmalı.**

**client\_id: bu Authentication Keys altında görünen ilk koddur.**

**redirect\_uri: bu, daha önce belirttiğiniz Authorized Redirect URL’lerinden biri olmalıdır.**

**state: Bu, CSRF saldırılarına karşı korumak için tasarlanmış, seçtiğiniz benzersiz bir dizedir.**

[16]

Bu parametre verilerinize göre özel bir link elde edilir. GET isteğiniz için bu linki tarayıcınızın adres çubuğuna yapıştırıp Enter'a basmak yeterlidir. Bu yapıldıktan sonra LinkedIn hesabınıza giriş yapmanızı isteyen bir ekran görmelisiniz. İzin ver tuşuna bastığınızda, geri arama olarak belirttiğiniz web sitesine yönlendirilirsiniz. Authentication Code'un oluşturulduğunu ve daha önce belirtilen durum dizgisinin de sorgunun sonunda olduğunu görüyorsunuz. Aşağıda belirtilen değişkenlerle https://www.linkedin.com/oauth/v2/accessToken sitesinde istek gönderildiğinde Access Token'e erişilebilmektedir [16].

**grant\_type: authorization code olmalıdır.**

**code: bu, önceki adımdaki yönlendirme URL'sine sahip olduğunuz kod olmalıdır.**

**redirect\_uri: bu, daha önce belirttiğiniz Authorized Redirect URL’lerinden biri olmalıdır.**

**client\_id and client\_secret: Authorization Keys altındaki iki alan.**

[16]

Bu erişim için yapılması gerekenler [17]:

**from linkedin\_v2 import linkedin**

**API\_KEY = 'wFNJekVpDCJtRPFX812pQsJee-gt0zO4X5XmG6wcfSOSlLocxodAXNMbl0\_hw3Vl'**

**API\_SECRET = 'daJDa6\_8UcnGMw1yuq9TjoO\_PMKukXMo8vEMo7Qv5J-G3SPgrAV0FqFCd0TNjQyG'**

**RETURN\_URL = 'http://localhost:8000'**

**authentication = linkedin.LinkedInAuthentication(API\_KEY, API\_SECRET, RETURN\_URL, linkedin.PERMISSIONS.enums.values())**

**print authentication.authorization\_url #bu URL'i tarayıcıda açar**

**application = linkedin.LinkedInApplication(authentication)**

**authentication.authorization\_code = 'AQTXrv3Pe1iWS0EQvLg0NJA8ju\_XuiadXACqHennhWih7iRyDSzAm5jaf3R7I8'**

**authentication.get\_access\_token() #Access Tokenı aldıktan sonra API'yi çağırabiliriz.**

**#Connections API, hesaplarına erişim izni veren bir kullanıcı için 1. derece bağlantıların bir listesini döndürür.**

**application.get\_connections()**

**{**

**"elements": [**

**{**

**"to": "urn:li:person:9HfhE6QlBz"**

**}**

**],**

**"paging": {**

**"total": 1,**

**"count": 50,**

**"start": 0,**

**"links": []**

**}**

**}**

**application.get\_connections(totals\_only=True)**

**{**

**"elements":[**

**],**

**"paging":{**

**"total":303,**

**"count":0,**

**"start":0**

**}**

**}**

[17]

# Sonuçlar

Çalışmamızın genel özeti:

Dünyada çok sık kullanılan sosyal medyalardan bahsedip, bu sosyal medya üzerindeki verilerin önemini anlatmak, sosyal medyalardan nasıl veri çekebileceğimizi göstermek ve bu veriler üzerinde ne tür analizler yapabileceğimizi göstermek için bu çalışmayı yapmış bulunmaktayız.

# Referanslar

[1]<https://towardsdatascience.com/how-to-use-facebook-graph-api-and-extract-data-using-python-1839e19d6999>

[2]<http://docs.tweepy.org/en/v3.5.0/>

[3]<https://medium.freecodecamp.org/basic-data-analysis-on-twitter-with-python-251c2a85062e>

[4]<https://www.sestek.com/tr/duygu-analizi/>

[5]<https://github.com/vprusso/youtube_tutorials/blob/master/twitter_python/part_4_visualizing_tweet_data/visualizing_twitter_data.py>

[6]<https://github.com/vprusso/youtube_tutorials/blob/master/twitter_python/part_1_streaming_tweets/tweepy_streamer.py>

[7]<https://github.com/vprusso/youtube_tutorials/blob/master/twitter_python/part_3_analyzing_tweet_data/analyzing_twitter_data.py>

[8]<https://www.youtube.com/watch?v=wlnx-7cm4Gg>

[9]<https://www.youtube.com/watch?v=WX0MDddgpA4&t=875s>

[10]<https://www.youtube.com/watch?v=w9tAoscq3C4&t=1s>

[11]<https://www.python.tc/numpy-numpy-egitim-serisi-giris/>

[12]<http://ertugruldeniz.com/pandas-nedir-ve-nasil-kullanilir-141>

[13]<https://pybilim.wordpress.com/2014/01/01/matplotlib-1-temel-grafikler/>

[14]<https://pybilim.wordpress.com/2014/01/01/matplotlib-1-temel-grafikler/>

[14]<https://www.sosyalmedyakampusu.com/yeni-baslayanlar-icin-sosyal-medya-analizi/>

[15]<https://medium.com/@abdullahkurkcu/sosyal-medya-madenciligi-python-twitter-475166a9f2e1>

[16]<https://medium.com/@ellesmuse/how-to-get-a-linkedin-access-token-a53f9b62f0ce>

[17]<https://pypi.org/project/python-linkedin-v2/>