



**T.C.**

**İSTANBUL ÜNİVERSİTESİ İŞLETME FAKÜLTESİ İŞLETME İKTİSADI  
ENSTİTÜSÜ**

**YÖNETİM BİLİŞİM SİSTEMLERİ  
TEZSİZ YÜKSEK LİSANS PROGRAMI**

**DENİZBANK MOBİL YORUMLARI ÜZERİNDE DUYGU ANALİZİ  
MODELLEMESİ**

**Proje Danışmanı : Prof. Dr. Tunçhan CURA**

**Hazırlayan : Ali Kerem Şimşek**

**DÖNEM PROJESİ**

**HAZİRAN 2023**

## **PROJE KABUL VE ONAY**

**İşletme Fakóltesi İşletme İktisadı Enstitüsü 9501228951 numaralı öğrencilerinden Ali Kerem Şimşek ‘in bitirme projesi olarak yaptığı “DENİZBANK MOBİL YORUMLARI ÜZERİNDE DUYGU ANALİZİ MODELLEMESİ” başlıklı çalışması, .../.../.... tarihinde, değerlendirilerek başarılı/başarısız bulunmuştur.**

**Bitirme Projesi Danışmanı**

**Prof. Dr. Tunçhan CURA**

## **BEYAN**

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederiz.

**Ali Kerem Şimşek**

**2023**

## ÖZET

### DENİZBANK MOBİL YORUMLARI ÜZERİNDE DUYGU ANALİZİ MODELLEMESİ

Ne yazık ki bazı insanların yaptıkları yorumlar ve puanlamaları birbiriyle uyuşmamaktadır. Gerek yapılan yorumlar üzerinden şirketlerin daha doğru adımlar atmasını, gerek sınıflandırılmış yorumlar üzerinde analizler yaparak artı ve eksi yönlerini anlamada destek olması açısından bir duygu durum analizi yapıp, bunun üzerinde model geliştirilmiş ve yeni gelen yorumları da analiz ederek veri setine ekleyecek bir program geliştirilmiştir. Böylece yapılan her bir yorumu doğru bir şekilde eğitim veri setine entegre ederek modelin gelişimi amaçlanmıştır.

**Anahtar kelimeler:** Veri Ön İşleme, Duygu Durum Analizi, Makine Öğrenmesi Modelleri

## ÖNSÖZ

Öncelikle verdiği dersler ve kattığı değer ile üzerimde emekleri olan danışman hocam Prof. Dr. Tunçhan Cura ‘ya ve yine destekleriyle üzerimde emeği olan müdürüm sayın Tuncay Çavdar ‘a teşekkürlerimi sunarım. Ayrıca tüm eğitim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan aileme ve sevdiklerime teşekkürlerimi borç bilirim.

Ali Kerem Şimşek

Haziran 2023

## İÇİNDEKİLER

BEYAN .....	iii
ÖZET.....	iv
ÖNSÖZ .....	v
ŞEKİLLER LİSTESİ .....	viii
GİRİŞ .....	1
1. Duygu Analizi .....	2
1.1. Duygu Analizi Nedir? .....	2
1.2. Duygu Analizi Neden Önemlidir?.....	2
1.2.1. Nesnel öngörüler sağlama.....	2
1.2.2. Daha iyi ürün ve hizmetler oluşturma .....	2
1.2.3. Uygun ölçekte analiz.....	3
1.2.4. Gerçek zamanlı sonuçlar.....	3
1.3. Duygu Analizi Kullanım Örnekleri .....	3
1.3.1. Duygu analizi kullanım örnekleri nelerdir? .....	3
1.3.2. Müşteri hizmetlerini iyileştirme .....	3
1.3.3. Marka izleme .....	3
1.3.4. Pazar araştırması .....	3
1.3.5. Kampanya performansını takip etme.....	4
1.4. Duygu Analizi Nasıl Çalışır?.....	4
1.4.1. Ön işleme.....	4
1.4.2. Anahtar kelime analizi .....	4
1.5. Duygu Analizine Yaklaşımlar Nelerdir?.....	4
1.5.1. Kural tabanlı.....	4
1.5.2. Kural temelli analiz örneği.....	5
1.5.3. Artıları ve eksileri.....	5
1.5.4. ML.....	5
1.5.5. Eğitim.....	5
1.5.6. Artıları ve eksileri .....	5
1.5.7. Hibrit.....	6
1.6. Farklı Duygu Analizi Türleri Nelerdir? .....	6
1.6.1. Çok tonlu puanlama .....	6
1.6.2. Hedef tabanlı .....	6
1.6.3. Niyet tabanlı .....	6
1.6.4. Duygusal algılama.....	6
1.7. Duygu Analizindeki Zorluklar Nelerdir? .....	7

1.7.1.	İğneleyici söz.....	7
1.7.2.	Olumsuzluk.....	7
1.7.3.	Çok kutupluluk.....	7
1.8.	Semantik Analiz Nedir?.....	7
1.8.1.	Duygu analizi ve semantik analiz.....	7
2.	Uygulama.....	8
2.1.	Android Marketten Yorumları Çekmek.....	8
2.2.	Veri Ön İşleme.....	9
2.3.	Duygu Analizi.....	10
3.	Makine Öğrenmesine Hazırlık.....	13
3.1.	TF-IDF.....	13
4.	Makine Öğrenmesi.....	14
5.	Model Sürekliliği İçin Uygulama Geliştirme.....	15
6.	Sonuç.....	17
KAYNAKÇA.....		18

## ŞEKİLLER LİSTESİ

Şekil 1. Android Marketten Veri Çekme .....	8
Şekil 2. Verileri Tanıma .....	9
Şekil 3. Veri Ön İşleme Adımları .....	9
Şekil 4. Cümleleri Sınıflandırma .....	10
Şekil 5. Sınıfları Saydırma .....	11
Şekil 6. Stopwords ve Lemmatize .....	12
Şekil 7. En Çok Geçen Kelimeler .....	13
Şekil 8. TF-IDF & ngram model .....	14
Şekil 9. Modellerin Kurulması ve Test Edilmesi .....	14
Şekil 10. Hiper Parametre Optimizasyonu .....	15
Şekil 11. Çapraz Doğrulama İle Son Durumun Değerlendirilmesi .....	15
Şekil 12. Uygulamanın Geliştirilmesi .....	16
Şekil 13. Uygulamanın Çıktısı .....	17



## GİRİŞ

Günümüzde gittikçe artan sosyal medya kullanımı ve veri madenciliğinin temel ilkelerinden birisi olan “yenilenebilir ve kullanabilir veriden anlamlı ifadeler çıkarma” düşüncesi son günlerde önemli araştırma konularından birisi olan ‘Duygu Analizi’(Sentiment Analysis)’a olan popülerliği arttırmış ve beraberinde kullanım alanlarını genişletmiştir.

Duygu Analizi veya diğer bir adı ile ‘Fikir Madenciliği’(Opinion Mining) tam anlamıyla birlikte; bir yazarın veya bir yazının belirli bir konuya yönelik tutumunun olumlu(positive),olumsuz(negative),tarafsız(neutral) olarak değerlendirilmesi için bir metin parçasıyla belirtilen görüşleri/ifadeleri çeşitli algoritmalar aracılığı ile hesaplayıp tanımlama ve sınıflandırma süreçlerine verilen genel bir tanımdır.

Herhangi bir metne bakarak, olumlu veya olumsuz olarak tek başına değerlendirmek yeterince bir anlam ifade edemediği dijital bir çağ yaşıyoruz. Biraz eskiye dönmek gerekirse İnternet kavramının henüz trend olmadığı bir dönemde medya takibi yapabilmek oldukça kolaydı. Sayısal olarak basılı veya görsel medyanın tümünü takip edebilmek, bir kişi ya da kurumla alakalı çıkan tüm haberleri toparlayabilmek ve incelenebilmek çok kolaydı. Bu durum şimdilerde gelişen teknoloji ve sosyal medyanın artan gücü ile aynı kalmamakta; İnternet medyası dışında, oluşturulan bloglar, sözlükler, Twitter üzerinden paylaşılan tweetler, Facebook ve Instagram gibi oldukça popüler olan kanallarda her saniye paylaşılan post’lar milyonlarca belki de milyarlarca yeni kaynaklar üretmektedir.

Bu noktada Duygu Analizi, kişilerin ve olayların üretilen ürünlerin ve sunulan hizmetlerin hakkında duygu ve düşünce tespitini yapmayı mümkün kılabilenmektedir. Bu çalışmada DanizBank mobil uygulamasına yapılan yorumlar üzerinden uygulama hakkında genel düşüncelerin neler olduğunu inceleyeceğiz. Ayrıca geliştirdiğimiz model sayesinde her yeni gelen yorumu veri setimize ekleyip hem modeli geliştirecek, hem de analizin sürekliliğini sağlayacağız.

## 1. Duygu Analizi

### 1.1. Duygu Analizi Nedir?

Duygu analizi, mesajın duygusal tonunun olumlu, olumsuz veya tarafsız olup olmadığını belirlemek için dijital metnin analiz edilme sürecidir. Günümüzde şirketler, e-postalar, müşteri desteği sohbet dökümleri, sosyal medya yorumları ve incelemeler gibi büyük hacimli metin verilerine sahiptir. Duygu analizi araçları, yazarın bir konuya yönelik tutumunu otomatik olarak belirlemek için bu metni tarayabilir. Şirketler, müşteri hizmetlerini iyileştirmek ve marka itibarını artırmak için duygu analizinden elde edilen bilgileri kullanır.

### 1.2. Duygu Analizi Neden Önemlidir?

Fikir madenciliği olarak da bilinen duygu analizi, şirketlerin ürün ve hizmetlerini geliştirmelerine yardımcı olan önemli bir iş zekâsı aracıdır. Aşağıda duygu analizinin bazı avantajlarına değineceğiz.

#### 1.2.1. Nesnel öngörüler sağlama

İşletmeler, yapay zekâ (AI) tabanlı duygu analizi araçlarını kullanarak insan inceleyicilerle ilişkili kişisel ön yargılardan kaçınabilir. Sonuç olarak, şirketler müşterilerin görüşlerini analiz ederken tutarlı ve objektif sonuçlar elde eder.

Örneğin, aşağıdaki cümleyi inceleyelim:

İşlemcinin hızına hayran kaldım ama çabuk ısındığı için hayal kırıklığına uğradım.

Pazarlamacılar incelemenin cesaret kırıcı kısmını reddedebilir ve işlemcinin performansına karşı olumlu yönde ön yargılı olabilirler. Ancak, doğru duygu analizi araçları, duyguları nesnel olarak almak için metni sıralar ve sınıflandırır.

#### 1.2.2. Daha iyi ürün ve hizmetler oluşturma

Duygu analizi sistemi, şirketlerin ürün ve hizmetlerini gerçek ve açık müşteri geri bildirimlerine göre geliştirmelerine yardımcı olur. Yapay zekâ teknolojileri, müşterilerin olumsuz duygularla ilişkilendirdiği gerçek dünyadaki nesneleri veya durumları (varlıklar olarak adlandırılır) tanımlar. Yukarıdaki örnekte, ürün mühendisleri, hayal kırıklığına uğratan (olumsuz) ve

ısınan (varlık) işlemci (varlık) ile ilişkilendirilen metin analiz yazılımı nedeniyle işlemcinin ısı yönetimi yeteneğini geliştirmeye odaklanır.

### **1.2.3. Uygun ölçekte analiz**

İşletmeler, e-postalar, chatbot dökümleri, anketler, müşteri ilişkileri yönetimi kayıtları ve ürün geri bildirimi gibi çok sayıda yapılandırılmamış veriden sürekli bilgi çıkarır. Bulut temelli duygu analizi araçları, işletmelerin metin verilerindeki müşteri duygularını ortaya çıkarma sürecini uygun bir maliyetle ölçeklendirmesine olanak tanır.

### **1.2.4. Gerçek zamanlı sonuçlar**

İşletmeler, günümüzün hızla değişen ortamında potansiyel krizlere veya pazar trendlerine hızlı yanıt vermelidir. Pazarlamacılar, müşterilerin şirketin markası, ürünleri ve hizmetleri hakkında neler hissettiğini gerçek zamanlı olarak öğrenmek ve bulgularına göre anında harekete geçmek için duygu analizi yazılımına güvenir. Belirli anahtar kelimeler için olumsuz duygular algılandığında yazılımı uyarılar gönderecek şekilde yapılandırabilirler.

## **1.3. Duygu Analizi Kullanım Örnekleri**

### **1.3.1. Duygu analizi kullanım örnekleri nelerdir?**

İşletmeler, zekâ elde etmek ve farklı alanlarda eyleme geçirilebilir planlar oluşturmak için duygu analizini kullanır.

### **1.3.2. Müşteri hizmetlerini iyileştirme**

Müşteri destek ekipleri, yanıtları görüşmenin ruh hâline göre kişiselleştirmek için duygu analizi araçlarını kullanır. Acil olan konular, duygu analizi yeteneğine sahip yapay zekâ (AI) tabanlı sohbet robotları tarafından tespit edilir ve destek personeline iletilir.

### **1.3.3. Marka izleme**

Kuruluşlar, sosyal medyada, forumlarda, bloglarda, haber makalelerinde ve diğer dijital alanlarda markaları hakkındaki ifadeleri ve sohbetleri sürekli olarak izler. Duygu analizi teknolojileri, halkla ilişkiler ekibinin ilgili devam eden hikâyelerden haberdar olmasını sağlar. Ekip, şikayetleri ele almak veya olumlu trendlerden yararlanmak için altta yatan ruh hâlini değerlendirebilir.

### **1.3.4. Pazar araştırması**

Bir duygu analizi sistemi, neyin işe yarayıp neyin yaramadığını öğrenerek işletmelerin ürün tekliflerini geliştirmelerine yardımcı olur. Pazarlamacılar, belirli ürün özelliklerine ilişkin daha derin öngörüler elde etmek için çevrimiçi inceleme sitelerindeki yorumları, anket yanıtlarını ve sosyal medya

gönderilerini analiz edebilir. Bulguları buna göre yenilik yapan ürün mühendislerine aktarırlar.

### **1.3.5. Kampanya performansını takip etme**

Pazarlamacılar, reklam kampanyalarının beklenen yanıtı oluşturmalarını sağlamak için duygu analizi araçlarını kullanır. Sosyal medya platformlarındaki konuşmaları izlerler ve genel duyguların cesaret verici olmasını sağlarlar. Net duygu beklentinin altına düşerse pazarlamacılar kampanyayı gerçek zamanlı veri analizine göre değiştirirler.

## **1.4. Duygu Analizi Nasıl Çalışır?**

Duygu analizi, metinleri insanlara benzer şekilde anlaması için bilgisayar yazılımını eğiten doğal dil işleme (NLP) teknolojilerinin bir uygulamasıdır. Analiz, nihai sonucu vermeden önce tipik olarak birkaç aşamadan geçer.

### **1.4.1. Ön işleme**

Ön işleme aşamasında, duygu analizi, metnin temel mesajını vurgulamak için anahtar kelimeleri tanımlar.

- Belirteçlere ayırma, bir cümleyi birkaç ögeye veya belirteçlere ayırır.
- Kök çözümleme, kelimeleri kök formlarına dönüştürür. Örneğin, “olmak”ın kök formu “ol”dur.
- Gereksiz kelimeleri kaldırma, cümleye anlamlı değer katmayan kelimeleri filtreler. Örneğin, ile, için, de ve ki etkisiz kelimelerdir.

### **1.4.2. Anahtar kelime analizi**

NLP teknolojileri, çıkarılan anahtar kelimeleri daha fazla analiz eder ve onlara bir duygu puanı verir. Duygu puanı, duygu analizi sistemindeki duygusal unsuru gösteren bir ölçüm ölçeğidir. Analitik amaçlar için metinde ifade edilen duyguların göreceli bir algısını sağlar. Örneğin, araştırmacılar müşteri incelemelerini analiz ederken memnuniyet için 10'u, hayal kırıklığı içinse 0'ı kullanır.

## **1.5. Duygu Analizine Yaklaşımlar Nelerdir?**

Duygu analizi yazılımı tarafından kullanılan üç ana yaklaşım vardır.

### **1.5.1. Kural tabanlı**

Kural tabanlı yaklaşım, önceden belirlenmiş sözlüklere göre belirli anahtar kelimeleri tanımlar, sınıflandırır ve puanlar. Sözlükler, yazarın niyetini,

duygusunu ve ruh hâlini temsil eden kelimelerin derlemeleridir. Pazarlamacılar, farklı ifadelerin duygusal ağırlığını yansıtmak için olumlu ve olumsuz sözlüklere duygu puanları atar. Bir cümle için olumlu, olumsuz veya tarafsız olup olmadığını belirlemek için yazılım, sözlükte listelenen kelimeleri tarar ve duygu puanını toplar. Nihai puan, genel duygusal duruşu belirlemek için duygu sınırlarıyla karşılaştırılır.

#### **1.5.2. Kural temelli analiz örneği**

Mutlu, uygun fiyatlı ve hızlı gibi kelimelerin olumlu sözlükte ve yetersiz, pahalı ve zor gibi kelimelerin olumsuz sözlükte bulunduğu bir sistem düşünün. Pazarlamacılar olumlu kelime puanlarını 5 ile 10 arasında ve olumsuz kelime puanlarını -1 ile -10 arasında belirler. Kötü değil ifadesinin olumlu olarak algılanması gibi, ikili olumsuzları tanımlamak için özel kurallar belirlenir. Pazarlamacılar, 3'ün üzerindeki genel duygu puanının olumlu olduğuna karar verirken - 3 ile 3 arasındaki puanlar karma duygu olarak etiketlenir.

#### **1.5.3. Artıları ve eksileri**

Kural tabanlı bir duygu analiz sistemi kurmak kolaydır ancak ölçeklendirmek zordur. Örneğin, metin girişinde niyet iletmek için yeni anahtar kelimeler keşfettiğinizde sözlükleri genişletmeye devam etmeniz gerekir. Ayrıca, farklı kültürlerden etkilenen cümleleri işlerken bu yaklaşım doğru olmayabilir.

#### **1.5.4. ML**

Bu yaklaşım bilgisayar yazılımına metin üzerinden duyguları tanımlamayı öğretmek için makine öğrenimi (ML) tekniklerini ve sinir ağları ile derin öğrenme gibi duygu sınıflandırma algoritmalarını kullanır. Bu süreç, bir duygu analizi modeli oluşturmayı ve bilinmeyen verilerdeki duyguyu yüksek doğrulukla tahmin edebilmesi için bilinen veriler üzerinde tekrar tekrar eğitmeyi içerir.

#### **1.5.5. Eğitim**

Eğitim sırasında, veri bilimcileri çok sayıda örnek içeren duygu analizi veri kümelerini kullanır. ML yazılımı veri kümelerini girdi olarak kullanır ve önceden belirlenmiş sonuca ulaşmak için kendini eğitir. Yazılım, çok sayıda farklı örnekle eğitim alarak farklı kelime düzenlemelerinin nihai duygu puanını nasıl etkilediğini ayırt eder ve belirler.

#### **1.5.6. Artıları ve eksileri**

ML duygu analizi, çok çeşitli metin bilgilerini doğru bir şekilde işlediği için avantajlıdır. Yazılım yeterli örneklerle eğitime girdiği sürece, ML duygu analizi, mesajların duygusal tonunu doğru bir şekilde tahmin edebilir. Bununla birlikte, eğitilmiş bir ML modeli bir iş alanına özeldir. Bu, pazarlama verileriyle eğitilmiş duygu analizi yazılımının yeniden eğitilmeden sosyal medya izleme için kullanılamayacağı anlamına gelir.

### **1.5.7. Hibrit**

Hibrit duygu analizi, hem ML hem de kural tabanlı sistemleri birleştirecek çalışır. Metinde bağlamsal niyet türetirken hızı ve doğruluğu optimize etmek için her iki yöntemin özelliklerini kullanır. Ancak, iki farklı sistemi bir araya getirmek zaman ve teknik çaba gerektirir.

## **1.6. Farklı Duygu Analizi Türleri Nelerdir?**

İşletmeler, müşterilerinin ürün veya hizmetlerle etkileşimde bulunurken nasıl hissettiğini anlamak için farklı türde duygu analizleri kullanır.

### **1.6.1. Çok tonlu puanlama**

Çok tonlu duygu analizi, metin amacını birden çok duygu seviyesine ayırmayı ifade eder. Tipik olarak yöntem, her bir eşit segmentin çok olumlu, olumlu, tarafsız, olumsuz ve çok olumsuz ifadelerini temsil ettiği 0 ile 100 arasında bir ölçekte kullanıcı duygusunu derecelendirmeyi içerir. E-ticaret mağazaları, satın alma deneyimini ölçmek için çok tonlu puanlama yöntemi olarak 5 yıldızlı bir derecelendirme sistemi kullanır.

### **1.6.2. Hedef tabanlı**

Hedef tabanlı analiz, bir ürün veya hizmetin belirli yönlerine odaklanır. Örneğin, dizüstü bilgisayar üreticileri müşterilere ses, grafik, klavye ve dokunmatik yüzey ile ilgili deneyimleri hakkında anket yaparlar. Müşterilerin niyetini donanım ile ilgili anahtar kelimelerle ilişkilendirmek için duygu analizi araçlarını kullanırlar.

### **1.6.3. Niyet tabanlı**

Niyet tabanlı analiz, pazar araştırması yaparken müşteri duygusunun anlaşılmasına yardımcı olur. Pazarlamacılar, belirli bir müşteri grubunun satın alma döngüsündeki konumunu anlamak için fikir madenciliği kullanırlar. İzlenen konuşmalarda indirimler, fırsatlar ve incelemeler gibi kelimeleri aldıktan sonra satın almak isteyen müşteriler için hedeflenmiş kampanyalar yürütürler.

### **1.6.4. Duygusal algılama**

Duygusal algılama, bir kişinin metni yazarken ki psikolojik durumunu analiz etmeyi içerir. Duygusal algılama, yalnızca kategorilere ayırmaktan daha derine indiği için duygu analizinin daha karmaşık bir disiplindir. Bu yaklaşımda, duygu analizi modelleri kişinin kelime seçimi yoluyla neşe, öfke, üzüntü ve pişmanlık gibi çeşitli duyguları yorumlamaya çalışır.

## 1.7. Duygu Analizindeki Zorluklar Nelerdir?

Doğal dil işleme (NLP) teknolojilerindeki gelişmelere rağmen, insan dilini anlamak makineler için zordur. Aşağıda verilenler gibi insan iletişiminin daha ince nüanslarını yanlış yorumlayabilirler.

### 1.7.1. İğneleyici söz

Bir bilgisayarın iğneleyici söz içeren cümlelerde duyguları analiz etmesi son derece zordur. Şu cümleyi düşünün: Aynen, harika. Siparişimin gelmesi üç hafta sürdü. Bilgisayar senaryoyu tam olarak anlayıp cümleyi analiz etmedikçe, deneyimi harika kelimesine dayanarak olumlu olarak etiketleyecektir.

### 1.7.2. Olumsuzluk

Olumsuzluk, cümlede anlamın tersine çevrilmesini iletmek için olumsuz kelimelerin kullanılmasıdır. Örneğin, Aboneliğin pahalı olduğunu söyleyemem. Aboneliğin ucuz olduğunu düşündüm. Öyle değildi. örneğinde olduğu gibi, özellikle iki cümle arasında olumsuzlama olursa duygu analizi algoritmaları, bu tür cümleleri doğru yorumlamakta zorluk çekebilir.

### 1.7.3. Çok kutupluluk

Çok kutupluluk, bir cümle birden fazla duygu içerdiğinde ortaya çıkar. Örneğin, bir ürün incelemesinde şöyle bir ifade vardır: Sağlam yapıdan memnunum ama renkten etkilenmedim. Yazılımın altta yatan duyguyu yorumlaması zorlaşır. Her bir varlığı ve ona karşılık gelen duyguyu çıkarmak için hedef tabanlı duygu analizini kullanmanız gerekecektir.

## 1.8. Semantik Analiz Nedir?

Semantik analiz, metin bilgilerindeki kelimelerin anlamını anlamak için kullanılan bir bilgisayar bilimi terimidir. Cümlelerde kelimeler ve dil bilgisi doğruluğu arasındaki ilişkiyi anlamak için makine öğrenimini (ML) ve doğal dil işlemeyi (NLP) kullanır.

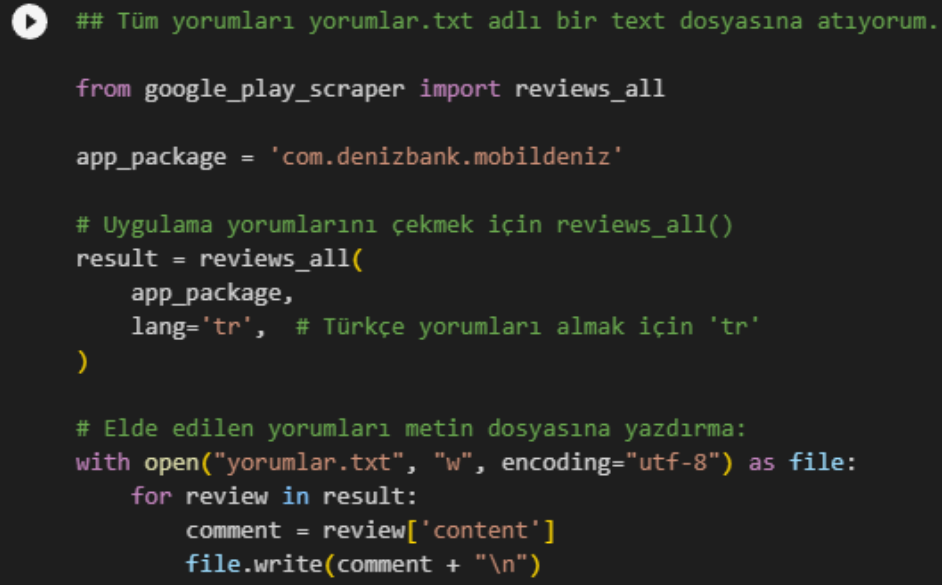
### 1.8.1. Duygu analizi ve semantik analiz

Duygu analizi çözümü, altta yatan duyguyu anlayarak metni sınıflandırır. ML algoritmasını belirli veri kümeleriyle eğiterek veya kural tabanlı sözlükler ayarlayarak çalışır. Bu arada, semantik bir analiz daha kapsamlı ve çeşitli bilgileri anlar ve onlarla çalışır. İşletmelerin müşterilerini daha iyi anlamalarına yardımcı olmak için her iki dil teknolojisi de entegre edilebilir.

## 2. Uygulama

Projeyi geliştirirken Google Colab kullandım. Şimdi burada yazdığım kodları yakından inceleyelim.

### 2.1. Android Marketten Yorumları Çekmek



```
## Tüm yorumları yorumlar.txt adlı bir text dosyasına atıyorum.  
  
from google_play_scraper import reviews_all  
  
app_package = 'com.denizbank.mobildeniz'  
  
# Uygulama yorumlarını çekmek için reviews_all()  
result = reviews_all(  
    app_package,  
    lang='tr', # Türkçe yorumları almak için 'tr'  
)  
  
# Elde edilen yorumları metin dosyasına yazdırma:  
with open("yorumlar.txt", "w", encoding="utf-8") as file:  
    for review in result:  
        comment = review['content']  
        file.write(comment + "\n")
```

Şekil 1. Android Marketten Veri Çekme

İlk olarak uygulamaya yapılan yorumları çekip bir text dosyasına satır satır yazdırıyorum.



## 2.2. Veri Ön İşleme

```
[ ] from warnings import filterwarnings
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from PIL import Image
from nltk.corpus import stopwords
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score, GridSearchCV, cross_validate
from sklearn.preprocessing import LabelEncoder
from textblob import Word, TextBlob
from wordcloud import WordCloud

[ ] filterwarnings('ignore') ## Bazı hataların önüne geçmek için, ignoreladım.
pd.set_option('display.max_columns', None) ## Tüm kolonları gösterdim.
pd.set_option('display.width', 200) ## Max genişliği 200 aldım. Rahat gözüksün diye.
pd.set_option('display.float_format', lambda x: '%.2f' % x) ## virgülden sonra 2 basamak göstermesi için.

[ ] AppFile = "/content/drive/MyDrive/DenizBank Mobil Yorumlar/yorumlar.txt"

# Metin dosyasını satır satır okuyarak verileri elde edin
with open(AppFile, "r", encoding="utf-8") as file:
    comments = [line.strip() for line in file]

# Verileri bir DataFrame'e dönüştürün
df = pd.DataFrame({"Yorumlar": comments})

print(df)
```

	Yorumlar
0	Normalde güzel çalışıyordu ama iki gündür açıl...
1	Mobil veri olayı yüzünden anlaşılmamı kestim bu ...
2	Mobil giriş yapamıyorum
3	Mobil deniz 🏖️ puan
4	Gayet iyi kullanışlı.
...	...

Şekil 2. Verileri Tanıma

Veri ön işleme için ve daha sonra kullanacağımız bazı işlemler için ilk olarak kütüphaneleri tanımlıyor, ardından bazı hataların önüne geçmek için birkaç ayar yapıp text dosyamızdan verilerimizi işlenebilir bir hal alması için bir dataframe'e dönüştürüyorum.

```
[ ] # Bütün yorumları küçük harf yapıyorum.
# İleride kullanacağım teknikler büyük küçük harf duyarlılığına sahip. Bu yüzden tüm kelimeleri küçük harfli yapıyorum.
df['Yorumlar'] = df['Yorumlar'].str.lower()

[ ] df['Yorumlar'] = df['Yorumlar'].str.replace('[^\w\s]', '') ## Noktalama işaretlerini boşluk ile değiştiriyorum.
## [^\w\s] bu ifade noktalama işaretlerini seçer.
```

Şekil 3. Veri Ön İşleme Adımları

Daha sonra, verilerimiz yorumlar olduğu için onları standart hale sokmamız gerek. Yani her bir kelimeyi işlenebilir, makinanın anlayacağı bir şekle sokmamız gerekiyor. Bunun için önce tüm harfleri küçük harfe çevirdim. Sonra da cümlelerde ki noktalama işaretlerini kaldırdım.

### 2.3. Duygu Analizi

Burada transformers kütüphanesini kullanacağımız için ilk olarak onu yüklememiz gerek. Bunu için : !pip install transformers komutunu kullanmalıyız.

```
## 13dk 56sn
import pandas as pd
from transformers import pipeline

sentiment_analysis_tr = pipeline("sentiment-analysis", model="savasy/bert-base-turkish-sentiment-cased")

data = df[:]

def analyze_sentiment(text):
    result = sentiment_analysis_tr(text)[0]
    label = result["label"]
    return label

data['Duygu'] = data["Yorumlar"].apply(analyze_sentiment)

print(data)
```

Downloading (...)ve/main/config.json: 100% 596/596 [00:00<00:00, 12.3kB/s]  
Downloading pytorch\_model.bin: 100% 443M/443M [00:16<00:00, 35.8MB/s]  
Downloading (...)okenizer\_config.json: 100% 39.0/39.0 [00:00<00:00, 1.33kB/s]  
Downloading (...)solve/main/vocab.txt: 100% 263k/263k [00:00<00:00, 4.87MB/s]  
Downloading (...)cial\_tokens\_map.json: 100% 112/112 [00:00<00:00, 5.78kB/s]  
Xformers is not installed correctly. If you want to use memory\_efficient\_attention to accelerate training use the following command to install Xformers  
pip install xformers.

	Yorumlar	Duygu
1	mobil veri olayı yüzünden anlaşılmamı kestim bu ...	negative
2	mobil giriş yapamıyorum	negative
3	mobil deniz puan	positive
4	gayet iyi kullanışlı	negative
5	25yıldır vazgeçemediğim bankam	positive
...	...	...
4993	güzel bir uygulama	positive
4994	banka neden var	negative
4995	beş para etmez uygulama	negative
4996	berbat hatta berbat ötesi daha ileri ve teknik...	negative
4997	bu şekilde yeni güncelleniz hiç hoşuma gitme...	negative

[4997 rows x 2 columns]

Şekil 4. Cümleleri Sınıflandırma

Bu kısım da önemli noktalardan birisi, hazır bir veri seti üzerinde çalışmış olmam. Github 'da bir kişinin hazırladığı "savasy/bert-base-turkish-sentiment-cased" veri setini model olarak kullandım. Günün sonunda Yorumlar sütununun yanına Duygu adlı bir sütun daha ekleyerek, bu modele göre bir sınıflandırma yaptırmış oldum.

Ardından Şekil 5 'de ki adımları izleyerek her bir sınıfta kaç adet yorum olduğunu ve en çok geçen cümleleri (satırları) getirip veriyi biraz inceledim.

```

# DataFrame'deki "etiket" sütununu kullanarak pozitif ve negatif yorumları sayma
yorumlar = data['Yorumlar']
etiketler = data['Duygu']

# Pozitif yorumların sayısı
pozitif_yorumlar = yorumlar[etiketler == 'positive']
pozitif_yorum_sayisi = len(pozitif_yorumlar)

# Negatif yorumların sayısı
negatif_yorumlar = yorumlar[etiketler == 'negative']
negatif_yorum_sayisi = len(negatif_yorumlar)

# Sonuçları yazdırma
print("Pozitif yorum sayısı:", pozitif_yorum_sayisi)
print("Negatif yorum sayısı:", negatif_yorum_sayisi)

```

```

Pozitif yorum sayısı: 1728
Negatif yorum sayısı: 3269

```

```

[ ] from collections import Counter

# Negatif yorumlardaki en çok tekrar eden satırı bulma
en_cok_tekrar_eden_satir_neg = Counter(negatif_yorumlar).most_common(1)
en_cok_tekrar_eden_satir_neg = en_cok_tekrar_eden_satir_neg[0][0]

# Pozitif yorumlardaki en çok tekrar eden satırı bulma
en_cok_tekrar_eden_satir_pos = Counter(pozitif_yorumlar).most_common(1)
en_cok_tekrar_eden_satir_pos = en_cok_tekrar_eden_satir_pos[0][0]

# Sonuçları yazdırma
print("Negatif yorumlardaki en çok tekrar eden satır:")
print(en_cok_tekrar_eden_satir_neg)
print("Pozitif yorumlardaki en çok tekrar eden satır:")
print(en_cok_tekrar_eden_satir_pos)

```

```

Negatif yorumlardaki en çok tekrar eden satır:
berbat
Pozitif yorumlardaki en çok tekrar eden satır:
güzel

```

Şekil 5. Sınıfları Saydırma

```
[ ] import nltk
nltk.download('stopwords') # Stopwords fonksiyonu için.
nltk.download('wordnet') # Lemmatize fonksiyonu için.

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True

sw = stopwords.words('turkish') ## Stopwordsleri sw değişkenine atadım. Nedir bu stopwordsler; "acaba, ama, birkaç, diye, eğer, gibi, hep, ki, kez" gibi kelimelerdir.
negatif_yorumlar = negatif_yorumlar.apply(lambda x: " ".join(x for x in str(x).split() if x not in sw))
pozitif_yorumlar = pozitif_yorumlar.apply(lambda x: " ".join(x for x in str(x).split() if x not in sw))
## split() metodunun ön tanımlı dgeeri boşluktur. Bu yüzden ekstra bir şey yapmamıza gerek kalmadı.
## x for x in str(x) burada ise metni string olarak alıyorum.
## if x not in sw ile stopwords olanları eliyorum.
## lambda x: " ".join() ile de boşluklara göre birleştiriyorum. Kelimeler arasında boşluk bırakarak birleştirir.

[ ] negatif_yorumlar = negatif_yorumlar.apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
pozitif_yorumlar = pozitif_yorumlar.apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
## Burada kelimeleri köklerine ayırıyorum.

temp_df_neg = pd.Series(' '.join(negatif_yorumlar).split()).value_counts()
temp_df_pos = pd.Series(' '.join(pozitif_yorumlar).split()).value_counts()
## value_counts() ile kelimeleri saydım.

print(temp_df_neg)
print(temp_df_pos)
```

```
bir      1244
uygulama    771
mobil      578
müşteri    475
banka      403
...
istedigimi      1
aradığınız      1
oluruz          1
getirirseniz    1
kullanışlıydı    1
Length: 12523, dtype: int64
güzel          335
iyi            230
bir            217
uygulama       198
harika         111
...
2www2www2      1
aaawawaa2www2wawawaw22www2222w      1
belediyesi      1
beyoğlu         1
uygulama         1
Length: 2348, dtype: int64
```

Şekil 6. Stopwords ve Lemmatize

Şekil 6 ‘da ise, Stopwords ve Lemmatize işlemlerini uyguladım.

Nedir bu Stopwords? Dilimizdeki “acaba, ama, birkaç, diye, eğer” gibi anlam değeri olmayan, ölçülemeyen kelimelerdir. Bunları çıkartıyoruz ki cümlelerde ki anlam bütünlüğünü koruyalım.

Ardından Lemmatize işlemi yapıyorum. Lemmatize ise kelimeleri köklerine ayırma işlemidir. Burada dikkat etmemiz gereken durum ise Türkçenin çekimli bir dil olmasıdır. Yani göz, gözlük, gözlükçü gibi kelimenin köküne yapım ve çekim ekleri ekleyerek kelimenin anlamının değiştiğidir. Böyle durumlarda kelimenin kökünü değil tümünü dikkate almalıyız. Biz burada anlam bütünlüğü üzerinden değil de kelimenin kökleri üzerinden ne duydu ifade ettiğini ağırlık hesabı ile yaptığımız için köklerine ayırıyorum.

```
# En çok geçen ilk 10 kelimeyi yeni birer listeye atadım.
temp_df_neg_fnl = temp_df_neg[:10]
temp_df_pos_fnl = temp_df_pos[:10]

print(temp_df_neg_fnl)
print(temp_df_pos_fnl)
```

bir	1244
uygulama	771
mobil	578
müşteri	475
banka	403
yok	388
giriş	356
kredi	320
var	285
kadar	274
dtype: int64	
güzel	335
iyi	230
bir	217
uygulama	198
harika	111
süper	109
deniz	102
banka	96
ederim	92
denizbank	86
dtype: int64	

Şekil 7. En Çok Geçen Kelimeler

Lemmatize ve stopwords işlemlerini yaptıktan sonra geriye kalan değerli varsayabileceğimiz kelimeler içinde en çok tekrar eden 10 kelimeyi seçiyorum ve inceliyorum.

### 3. Makine Öğrenmesine Hazırlık

Bu aşamada verileri öyle bir hale getirmeliyiz ki makinenin anlayabileceği bir hale gelsin. Bunun için birkaç yöntem var.

1. Count Vectors : Kısaca her bir kelimeyi vektörize ederek saydırma işlemidir.
2. TF-IDF Vectors : Normalize edilmiş frekansları temsil eder.
3. Word Embeddings (Word2Vec, Glove, BERT gibi)

#### 3.1. TF-IDF

Count Vectors 'ün ortaya çıkarabileceği bazı yanlışlıkları gidermek adına, normalize edilmiş yani standartlaştırılmış bir kelime vektörü oluşturma yöntemidir.

Bu aşamada kelimeleri vektörlere dönüştürüp makinenin anlayabileceği bir hale sokacağız.

Yine bu aşamada, kelimelerin dökümanlarda geçme frekansını ve kelimelerin tüm corpusta geçme frekansları odağında bir standartlaştırma işlemi uygulamış olacağız.

Yine burada, ngram 'lar üzerinde çalışmamı yürüteceğim. Nedir bu ngram model?

Mesela ngram\_range = (2,4) şu anlama gelmektedir;

2-gram'lar: "Bu ürün", "ürün gerçekten", "gerçekten çok", "çok kullanışlı", "kullanışlı ve", "ve kaliteli"

3-gram'lar: "Bu ürün gerçekten", "ürün gerçekten çok", "gerçekten çok kullanışlı", "çok kullanışlı ve", "kullanışlı ve kaliteli"

4-gram'lar: "Bu ürün gerçekten çok", "ürün gerçekten çok kullanışlı", "gerçekten çok kullanışlı ve", "çok kullanışlı ve kaliteli"

yani 2li 3lü ve 4lü ngram modelleri oluşturur.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
tf_idf_word_vectorizer = TfidfVectorizer()

X_tf_idf_word = tf_idf_word_vectorizer.fit_transform(X) ## Word model

tf_idf_ngram_vectorizer = TfidfVectorizer(ngram_range=(2,5))
X_tf_idf_ngram = tf_idf_ngram_vectorizer.fit_transform(X) ## ngram model
```

Şekil 8. TF-IDF & ngram model

#### 4. Makine Öğrenmesi

```
[ ] ##count model
vectorizer = CountVectorizer()
X_count = vectorizer.fit_transform(X)
rf_model_1 = RandomForestClassifier().fit(X_count, y)
cross_val_score(rf_model_1, X_count, y, cv=10, n_jobs=-1).mean() ##n_jobs=-1 bütün işlemcileri kullan demektir.

0.897141482965932

[ ] ##word model
rf_model_2 = RandomForestClassifier().fit(X_tf_idf_word, y)
cross_val_score(rf_model_2, X_count, y, cv=10, n_jobs=-1).mean()

0.8931327327327327

[ ] ## ngram model
rf_model = RandomForestClassifier().fit(X_tf_idf_ngram, y)
cross_val_score(rf_model, X_count, y, cv=10, n_jobs=-1).mean()

0.8975378757515029
```

Şekil 9. Modellerin Kurulması ve Test Edilmesi

Burada bir sınıflandırma modeli oluşturduğumuz için Random Forest modelini kullanmanın daha faydalı olacağını düşündüm ve tek tek count vectorizer, tf-idf ve ngram modellerini deneyerek en başarılı olan modeli seçtim. Bu model ngram ile hazırladığım model oldu. %89.7 doğruluk elde etti. Başarımı hesaplarken 10 katlı çapraz doğrulama yaptım. Hemen ardından şekil 10 'da ki işlemleri gerçekleştirdim. Hiper parametre optimizasyonu uygulayıp başarıyı arttırmaya çalıştım.

```
[ ] rf_model = RandomForestClassifier(random_state=17) ## random_state=17 ile her defasında aynı sonucu alırız.
## Sayıyı rastgele yazıyoruz.

[ ] rf_params = {"max_depth": [5, 8, None], ## max derinlik
                 "max_features": [5, 7, "auto"], ## bölünmelerde göz önünde bulundurulacak özellik, değişken sayısı
                 "min_samples_split": [2, 5, 8, 20], ## bölünmelerde bir yaprakta ne kadar örnek olması gerektiğinin parametresi
                 "n_estimators": [100, 200, 500]} ## kaç tane ağaç fit edileceğini ifade eder

[ ] ## Başarının değerlendirilmesi için; 1 saat 46dk 30sn
    rf_best_grid = GridSearchCV(rf_model,
                               rf_params,
                               cv=10,
                               n_jobs=-1,
                               verbose=True).fit(X_count, y) ## verbose=True raporlama almamızı sağlar.

Fitting 10 folds for each of 108 candidates, totalling 1080 fits

[ ] rf_best_grid.best_params_ ## Yukarıda en iyi sonuç veren parametreleri girdik.

{'max_depth': None,
 'max_features': 5,
 'min_samples_split': 2,
 'n_estimators': 500}

▶ rf_final = rf_model.set_params(**rf_best_grid.best_params_, random_state=17).fit(X_count, y)
## **rf_best_grid.best_params_ iki yıldız ile yazarsak, içindeki key ve valueları al demek olur.
```

Şekil 10. Hiper Parametre Optimizasyonu

Hiper parametre optimizasyonu ile bulduğumuz en iyi değerleri parametreler olarak atadığımızda modelimiz artık şekil 11 'deki gibi %90 başarımları yakaladı.

```
▶ ## Son kez skorumuza bakalım.
cross_val_score(rf_final, X_count, y, cv=5, n_jobs=-1).mean()

0.9023385385385385
```

Şekil 11. Çapraz Doğrulama İle Son Durumun Değerlendirilmesi

## 5. Model Sürekliliği İçin Uygulama Geliştirme

```

print("\nCıkmak için 'çıkış' yazınız.")
print("\nEğer tahmin yanlışsa '0' yazın. \nDoğruysa '1' yazın: \n")

yeni_veriler = []

while True:
    print("-----")
    print("Yorumunuzu girin : ")
    yorum = input()
    if yorum == 'çıkış':
        break

    yorum = yorum.lower()
    yorum = re.sub(r'^\w\s', '', yorum)

    yeni_yorum = pd.Series([yorum])
    yeni_yorum_count = vectorizer.transform(yeni_yorum)
    tahmin = rf_final.predict(X_count[-1, :])

    if tahmin == 0:
        tahmin_etiketi = "negatif"
    else:
        tahmin_etiketi = "pozitif"

    print("-->Tahmin :", tahmin_etiketi)
    # Tahmin yanlışsa düzeltme yapılacak.
    print("Tahmin Doğru mu? (1/0)")
    duzeltme = 1
    duzeltme = int(input())
    if duzeltme == 0:
        tahmin = 1 - tahmin ## tahminin tersini alarak düzeltilmiş olucaz.
        print("\nDüzeltilme yapıldı.\n")
        if tahmin == 0:
            tahmin_etiketi = "negatif"
        else:
            tahmin_etiketi = "pozitif"
    print("Yorum: ", yorum, "--> Tahmin: ", tahmin_etiketi)

```

```

yeni_veri = {"Yorumlar": yorum, "Duygu": tahmin}
yeni_veriler.append(yeni_veri)

# Yeni yorumu veri setine yükleme;
#yeni_veri = pd.DataFrame.from_dict({"Yorumlar": [yorum], "Duygu": tahmin})
#data = pd.concat([data, yeni_veri], ignore_index=True)

# Veri kümesinin de güncellenmesi gerek;
y = data["Duygu"] ## Bağımlı değişkenimiz.
X = data["Yorumlar"] ## Bağımsız değişkenimiz.
X_count = vectorizer.fit_transform(X)

print("-----")

print("Çıkış yapıldı.")

yeni_veriler_df = pd.DataFrame(yeni_veriler)
yeni_veriler_df["Duygu"] = yeni_veriler_df["Duygu"].astype(int) # Duygu sütununu integer tipine dönüştürme
data = pd.concat([data, yeni_veriler_df], ignore_index=True)

# Güncellenmiş veri kümesini dosyamıza yazdırıyoruz ki datalarımız büyüsün, modelimiz doğru tahminlerle gelişsin.
data.to_csv('/content/drive/MyDrive/DenizBank Mobil Yorumlar/Yorumlar2.txt', index=False, sep='\t')

print("Veri kümesi güncellendi.")

```

Şekil 12. Uygulamanın Geliştirilmesi



Burada ki amacım, kurduğum model sayesinde yapılan her bir yeni yorumu tahminleyip, yanlış tahminlemesi halinde doğru sınıflandırıp veri setimize yüklemek ve sürekli olarak gelişen bir veri seti ve haliyle de sürekli olarak gelişen bir yapay zeka uygulaması geliştirmektir. Şekil 13 'de ise uygulamanın konsol üzerinde ki çıktısını gözlemleyebilirsiniz.

```
Çıkmak için 'çıkış' yazınız.

Eğer tahmin yanlışsa '0' yazın.
Doğruysa '1' yazın:

-----
Yorumunuzu girin :
harikamsı
-->Tahmin : negatif
Tahmin Doğru mu? (1/0)
0

Düzeltilme yapıldı.

Yorum: harikamsı --> Tahmin: pozitif
-----
-----
Yorumunuzu girin :
çıkış
Çıkış yapıldı.
Veri kümesi güncellendi.
```

Şekil 13. Uygulamanın Çıktısı

## 6. Sonuç

Tüm bu projede yapılmak istenen, gelişen teknolojik çağda internet ortamında yapılan herhangi bir yorumun işlenerek ne ifade ettiğini analiz etmek ve bunun üzerinde bir model kurarak bilgisayarın bu model sayesinde yeni gelen bir yorumun tahmin etmesini sağlamaktır. Bu sayede elimizde bir konu hakkında yapılan yorumların, düşüncelerin ağırlıklı olarak ne üzerine olduğu, daha çok pozitif mi yoksa negatif mi olduğunu anladık ve sürekli hale getirebildik.

## KAYNAKÇA

<https://ardabatuhandemir.medium.com/duygu-analizi-sentiment-anaylsis-nedir-68a59a8b0142>

<https://aws.amazon.com/tr/what-is/sentiment-analysis/>