PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

DOGAL DİL İŞLEME İLE DUYGU ANALİZİ

LİSANS TEZİ

Ramazan ÖLMEZ
(14253503)

Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Ü. Meriç ÇETİN

HAZİRAN 2020

Pamukkale Üniversitesi Bilgisayar Mühendisliği Bölümü, 14253503 numaralı Lisans Öğrencisi Ramazan ÖLMEZ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "DOGAL DİL İŞLEME İLE DUYGU ANALİZİ" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Dr. Öğr. Ü. Meriç ÇETİN** .....

Jüri Üyeleri : .....
Pamukkale Üniversitesi

.....
Pamukkale Üniversitesi

Teslim Tarihi : **Haziran 2020**
Savunma Tarihi : **Haziran 2020**

Önsöz

Çalışmalarım sırasında bilgi ve tecrübelerini benden esirgemeyen değerli hocam ve tez danışmanım Sayın Dr. Öğr. Ü. Meriç ÇETİN'e teşekkürlerimi sunarım.

Haziran 2020 Ramazan ÖLMEZ

İçindekiler

Sayfa

KISALTMALAR

DA : Duygu Analizi
MÖ : Makine Öğrenmesi
SGD : Stochastic Gradient Descent
DDİ : Doğal Dil İşleme
NB : Naive Bayes
LR : Lojistik Regresyon
DÖ : Derin Öğrenme
OLS : Ordinary Least Squares
MLE : Maximum Likelihood Estimation
SVM : Support Vector Machines
HTML : HyperText Markup Language
XML : Extensible Markup Language

SEMBOLLER

p	**:** Karakteristik özelliğin var olma olasılığı
b	**:** Lojistik regresyon katsayılarının vektörü
X	**:** Logistik regresyon sınıflandırıcı algoritmanın bazı değerleri
S|x	**:** x olayı gerçekleştiğinde, S olayının gerçekleşme olasılığı
x|S	**:** S olayı gerçekleştiğinde, x olayının gerçekleşme olasılığı
P(x)	**:** x olayı gerçekleşme olasılığı
P(S)	**:** S olayı gerçekleşme olasılığı

Tablo Listesi

Sayfa

Şekil Listesi

Sayfa

DOGAL DİL İŞLEME İLE DUYGU ANALİZİ

Özet

Dijital iletişimdeki üssel büyüme nedeniyle, bilgi teknolojisi günlük hayatımızın ayrılmaz bir parçası oldu. İnternetin ve sosyal medyanın hızlı bir şekilde yaygınlaşıp kabul görmesi, önceden bir konu hakkında pek mümkün ve kolay olmayan görüş bildirimini son kullanıcılar için de mümkün kılmıştır. Facebook, Instagram ve Twitter gibi sosyal medya platformları son yıllarda giderek daha popüler hale gelmiştir. Bu popülerleşme insanların aldıkları ürün hakkındaki düşünce ve görüşlerini sosyal medya ortamında sık sık yorumlamasına yol açmıştır. Bu noktada duygu analizi çalışmaları gerçekleştirilerek insanların olumlu, olumsuz veya tarafsız yorumları kategorilendirilmeye çalışılmıştır.

Bu tez çalışmasında sıklıkla kullanılan bir platform olan Twitter üzerinde paylaşılan Turkcell hakkındaki gönderiler ile duygu analizi çalışması yapılmıştır. Twitter üzerinden elde edilen Türkçe gönderilerden oluşan veriler içeriğine göre el ile pozitif veya negatif olarak etiketlenmiştir. Ham halde bulunan veri seti, başarılı sonuç vermeyeceği için bir takım ön işlemler geçirilmiştir. Kelimeler vektör hale getirdikten sonra eğitim ve test seti olarak iki parçaya bölünmüştür. Oluşturulan bu eğitim setleri kullanılarak farklı sınıflama algoritmaları ile modeller çıkarılmış ve bu modellerin çapraz doğrulama ile sınıflama başarımları bulunmuştur. Sınıflandırma için makine öğrenmesi algoritmalarından Lojistic Regresyon, Olasılıksal Dereceli Azalma ve Naive Bayes seçilmiştir. Eğitim setlerindeki özniteliksel farklılıklara bağlı olarak bu modeller için yapılan testlerde en iyi sınıflama başarımının; Lojistik Regresyon için %63.2, SGD Classifier için %64.5, Naive Bayes algoritması için ise %63.6 olduğu görülmüştür. Çalışma sonucunda en yüksek başarının Olasılıksal Dereceli Azaltma algoritmasında olduğu görülmüştür.

Anahtar Kelimeler: Duygu Analizi, Makine Öğrenmesi, Sınıflandırma

SENTIMENT ANALYSIS WITH NATURAL LANGUAGE PROCESSING

SUMMARY

Information technology has become an inseparable part of our daily lives because of the exponential growth in digital communication. The rapid spread and acceptance of the Internet and social media has also made it possible for end users to express their opinions about a subject that was not possible and easy before. Social media platforms such as Facebook, Instagram and Twitter have become increasingly popular in recent years. Due to this popularization, people frequently tend to comment on the products they buy on social media. At this point; positive, negative and neutral comments of people have been tried to be categorized by doing sentiment analysis study. This thesis aims to Do sentiment analysis study of the posts about turkcell shared on Twitter which is frequently used platform The data obtained from Turkish posts on Twitter are manually labeled as positive or negative depend on the content. The raw data has been subjected to a set of pretreatments as it might not have been successful in its original form. After the words are vectorized, they are divided into two parts as training and test sets. Using these training sets, models with different classification algorithms has been created and classification performance of these models has been found with cross verification. Logistic Regression, stochastic gradient descent and Naive Bayes were chosen from the machine learning algorithms to classify. Depend on the feature differences in the training sets, the best classification performance in the tests for these models has been observed as %63.2 for Logistic Regression, %64.5 for SGD Classifier and %63.6 for Naive Bayes algorithm. As a result of the study, it is observed that the highest success was in the stochastic gradient descent algorithm.

Keywords: Sentiment Analysis, Machine Learning, Classification

1. GİRİŞ

Sosyal medya, internet üzerinden fikir paylaşmak için çok popüler bir araç haline gelmiştir. Sosyal medya platformları her geçen gün daha fazla kullanıcı sayısına erişmektedir. İnsanlar ürünler, hizmetler, şirketler, gündemde bulunan konular hakkında fikirlerini sosyal medya aracılığıyla paylaşmaktan çekinmezler. Kullanıcılar çeşitli şekillerde bu ortamlarda veri paylaşmaktadırlar. Yine insanlar başkalarının duygularına, düşüncelerine, almak istedikleri ürünlerin kullananlar tarafından yapılan yorumlarına ve fikirlerine büyük önem duydukları için sosyal medyada duygu analizi (DA) çok önemli bir hale gelmiştir. Kullanıcılar hesapları üzerinden Twitter ortamını takip ederek ilgili olduğu konular hakkında paylaşılanları görebilir. Kullanıcılar, bu konular hakkında kendi paylaşımları ile duygusal durumunu ifade edebilir. Sosyal medya, şirket ve hükümetlerin karar vermesinde önemli bir araç haline gelmiştir. Şirketler Twitter paylaşımları üzerinden DA yaparak tüketicilerinin ürünleri hakkında ne söylediklerini kolayca belirleyebilmektedir. Twitter, kullanıcıların durumlarını paylaşabildikleri en popüler sosyal ağlardan birisidir. Twitter verileri, en fazla 280 karakterden meydana gelen cümlelerden oluşan ve veri yapısı gereğince duygu analizi için en çok tercih edilen verilerdir. Duyguları çıkarabileceğimiz kısa ve anlamlı bir metin içermektedir.

DA giderek artan popülaritesi ile daha iyi başarılar elde edebilmek adına birçok algoritma denenmiştir. Duygu sınıflandırması için çeşitli doğal dil işleme (DDİ) ve makine öğrenmesi (MÖ) teknikleri uygulanmaktadır. Bu teknikler içerisine son zamanlarda derin öğrenme (DÖ) algoritmaları da hızlıca girmektedir. Çalışmalar sonucunda alınan başarı puanları yüksektir fakat gerçek hayata uygulanma başarıları henüz yeterli değildir. Bu sebepten dolayı araştırmacılar için önemli bir ilgi alanıdır.

DA nin temel amacı, kişinin bir konu hakkındaki paylaşımından olumlu, olumsuz veya tarafsız gibi bir sonuca varmaktır. DA çok farklı şekillerde yapılabilir. En yaygın yollardan birisi pozitiflik, negatiflik ve tarafsızlık puanlarını hesaplayıp metnin olumlu, olumsuz veya tarafsız sınıfa ait olduğuna karar verilmesidir. Bir başka seçenek ise

sıfırdan beşe kadar bir pozitiflik skoru bulmaktır. Bu araştırma sürecinde iki ölçekli bir sınıflandırma üzerinde yoğunlaştım.

DA aynı zamanda, tüketicilerin ürünler hakkındaki görüşleri, seçmenleri siyasi partilere karşı tutumları veya yatırımcıların hisse senetleri hakkında beklentileri ile de ilgilidir. DA çalışmaları Facebook yorumları, Twitter gönderileri, film dizi incelemeleri gibi birbirlerinden değişik veriler ile gerçekleştirilebilir. DA için genel olarak iki yaklaşım ele alınır. Bunlardan ilki sözlük tabanlı DA' dir. Metindeki duygu içeren kelimelerden hesaplanır ve metinde tanımlanan duyguları ifade eden kelimelerden DA yapılır. İkinci yaklaşıma ise makine öğrenmesi tabanlı yaklaşım ismi verilir. Makine öğrenmesi tabanlı DA yaklaşımında, ilk olarak etiketli verilerin oluşturduğu geniş bir kümeden bir duygu modeli oluşturulur. Daha sonra sınıfı bilinmeyen verinin model aracılığı ile sınıfı tahmin edilir. Model, veriden çıkarılan öznitelikleri kullanarak duygu etiketlerini olumlu, olumsuz veya tarafsız olarak tahmin eder. Bu duygu sınıfları, paylaşanın mesaj içeriğinde ele alınan konuya karşı duygusunu ifade eder.

Son yıllarda Türkçe için yapılan birkaç DA çalışması olmasının yanında bugüne kadar yapılan çalışmaların çoğu İngilizce dili için gerçekleştirilmiştir. Bu çalışmadaki amaç Türkçe veri seti ile makine öğrenmesi tabanlı bir yaklaşım oluşturup başarımların hesaplanması ve farklı makine öğrenmesi algoritmalarının birbirleriyle kıyaslanmasıdır. Ayrıca performansı arttırabilecek, daha fazla bilgi çıkarımı sağlayacak yeni özelliklerin ve metotların incelenmesi de bu tezin amaçları arasına girmektedir. Bunlar kısaca Türkçe dilinin karakteristik özelliklerine uygun ön işlemler, yeni özellik çıkarımları ve bilgi çıkarımı yöntemlerinin kullanılması olarak belirtilebilir.

1.1 Tezin Amacı

Türkçe'nin sondan eklemeli bir dil yapısına sahip olması duygu analizi çalışmalarını zorlaştırmaktadır. Bu tez çalışmasının amacı, Twitter platformu üzerinden paylaşılan Türkçe Tweet'lerin pozitif veya negatif sınıflara ait olduklarının analiz edilmesidir.

Twitter paylaşımlarının, duygu analizi için 6 adımlı bir süreç izlenmiştir. Bu süreçler şu şekilde tanımlanabilir:

- Veri setini oluşturma

- Veri setinin el ile pozitif ve negatifliğinin etiketlenmesi
- Veri seti ön işleme (ayrıştırma, temizleme, öznitelik çıkarımı ...)
- Model oluşturmak ve test setinin bu model ile test edilmesi
- Sonuçların görselleştirilmesi

1.2 Literatür Araştırması

Duygu analizi bir sınıflandırma problemidir. Duygu analizi ile ilgili Makine Öğrenmesi ve sözlük tabanlı yöntemlerle birçok akademik çalışma yapılmıştır. Son yıllarda DDİ ve Görüntü İşleme alanlarında yüksek başarılı sonuçlar veren derin sinir ağları tabanlı Derin Öğrenme (DÖ) yöntemi de DA için kullanılan önemli yöntemlerden birisidir. Bu yöntem İngilizce için çokça kullanılan ve literatürde en yüksek başarımların elde edildiği çalışma alanı olarak karşımıza çıkmaktadır.

Meral ve diğ. [1], 9 farklı alanda (Telekom, Sağlık, Finans, Spor,...) 8321 adet Twitter gönderisinden elde ettikleri veri ile duygu analizi çalışması gerçekleştirmişler. Bu gönderileri olumlu, olumsuz ve nötr olarak etiketlemişler ve MÖ algoritması konusunda Naive Bayes, Rastgele Orman Ağacı ve Destek Vektör Makineleri kullanmışlardır. En başarılı sonucu %89.5 ile Rastgele Orman Ağacı algoritmasında elde etmişler.

Çoban ve diğ. [2], Twitter üzerinden elde ettikleri Türkçe tweetler ile his simgelerini kullanarak (':)', ':(' ...) 2 farklı grup oluşturmuşlar. Metinleri his simgeleri yardımıyla etiketlemişler. Pozitif ve negatif olmak üzere 2 farklı etiket kullanmışlar. Çalışmalarınında, 10000 pozitif, 10000 negatif olmak üzere toplam 20000 adet Türkçe Twitter paylaşımı oluşturmuşlar. Birden çok MÖ algoritması kullanmışlar ve en başarılı sonucu %66.06 ile Multinomial Naive Bayes ile elde etmişler.

Kaynar ve diğ. [3], deneyler için film yorumlarını tercih etmişler. 2000 film yorumundan oluşan IMDB veri setini kullanarak bu verileri pozitif ve negatif sınıflara ayırmışlar. Çalışmalarında Naive Bayes, Destek Vektör Makineleri, Merkez Tabanlı Sınıflayıcı ve Çok Katmanlı Yapay Sinir Ağları ile çalışmışlar. En başarılı sonucu %89.73 ile Yapay Sinir Ağları vermiş.

Wang ve diğ. [4], ABD seçimleriyle ilgili, başkan adayları hakkında atılan twitler içerisinde 200 den fazla kural uygulayarak Twitter verilerine erişmişler. Çalışmalarında 17.000 twit kullanmışlar. Bu gönderileri %16 pozitif, %56 negatif, %18 nötr ve %10 belirli değil şeklinde 4 kategori olarak etiketlemişler. Çalışmalarında Naive Bayes Sınıflandırıcı kullanarak %59 doğruluğa ulaşmışlar.

Santos ve diğ. [5] çalışmalarında film yorumlarından ve Twitter gönderilerinden oluşan iki farklı veri seti kullanmışlar. Birinci veri setinde 11855, ikinci veri setinde 96498 veri varmış. Bu veriler üzerinde pozitif ve negatif etiket uygulamışlar. Çalışma kapsamında iki farklı Yapay Sinir Ağı oluşturarak, %69 ve %77 oranında başarı oranı yakalamışlar.

Geniş kapsamlı olarak bir diğer çalışma ise, Rosenthal ve diğ. [6] çalışmalarında yerel ve küresel Twitter trendlerinden faydalanarak Arapça konuşulan bazı ülkelerde güncel olaylara dayanan İngilizce ve Arapça gönderileri seçmişler. Konular arasında: "Donald Trump", "Halep", "Suriyeli Mülteciler" ve "Vejetaryenlik" gibi birbirleri ile çok alakası olmayan alanlar belirlemişler. İngilizce 132321, Arapça ise 23323 gönderi üzerinde çalışmışlar. Çalışmalarını 5 alt kategoride şekillendirmişler. Bu kategoriler şu şekilde ifade edilebilir:

1. Bir tweetin Pozitif ve Negatif duygu içerip içermediğine karar verilmesi
2. Bir tweetin Pozitif, Negatif ve Nötr duygu içerip içermediğine karar verilmesi
3. Bir twetin Güçlü Pozitif, Zayıf Pozitif, Güçlü Negatif, Zayıf Negatif ve Nötr duygu içerip içermediğine karar verilmesi
4. Bir konu hakkında bir dizi tweet verildiğinde, tweet'lerin pozitif ve negatif sınıflar arasındaki dağılımının tahmin edilmesi
5. Bir konu hakkında bir dizi tweet verildiğinde, tweetlerin beş sınıf boyunca dağılımının tahmin edilmesi (Güçlü Pozitif, Zayıf Pozitif, Nötr, Zayıf Negatif ve Güçlü Negatif)

Ortigosa ve diğ., Facebook platformu üzerinde yaptığı çalışmada [7], İspanyolca dili ile atılmış olan kullanıcı yorumlarını kullanarak bir duygu analizi gerçekleştirmişler. Çalışmaları kapsamında elde ettikleri verileri pozitif ve negatif olarak etiketleyerek

2 çeşit etiket verisi kullanmışlar. 3 farklı yaklaşım gerçekleştirmişler (Sözlük Tabanlı, Makine Öğrenimi ve Hibrit). En iyi doğruluğu %83.27 Hibrit yaklaşım ile yakalamışlar.

Vural ve diğ. [8], Türkçe film yorumları için sözlük tabanlı bir DA çalışması yürütmüşlerdir. "beyazperde.com" adresinden topladıkları veri kümesini kullanmışlardır. Çalışmalarında olumlu-olumsuz sınıflandırma yapmışlar ve %76 başarı elde etmişlerdir.

Şimşek ve Özdemir, çalışmalarında [9], borsadaki değişim ile Twitter kullanıcıların ekonomi ile ilgili attıkları tweetler arasında bir ilişki olup olmadığını incelemişlerdir. Duygu sözlüğünden sekiz farklı duyguya ait 113 özellik seçilerek, bu özellikler ışığında tweetler mutlu-mutsuz olarak sınıflandırılmıştır. Çalışma sonucunda %45 ilişki olduğu saptanmıştır.

2. BİLİMSEL ARKAPLAN

2.1 Doğal Dil İşleme

Doğal Dil İşleme (DDİ), yapay zekanın gelişimi ve dil bilimle ortaklaşa geliştirilen çalışmalar sonucunda hayatımıza girmiş bir terimdir. En geniş kapsamıyla DDİ, Türkçe, İngilizce gibi doğal dillerdeki metinlerin, ses dalgalarının bilgisayar tarafından algılanarak yazılım programında çözümlenmesi ve bilgisayar ortamına aktarılmasıdır.

DDİ, makine diline çeviri, spam ile mücadele, bilginin çıkarılması, özetleme, soru cevaplama gibi birbirinden farklı birçok alanda kullanılmaktadır. Belirli bir metinden görüş almayı amaçlayan duygu analizi de DDİ'nin bir çalışma alanıdır.

2.2 Duygu Analizi

Duygu ve fikirlerin metinlerden elde edilerek çıkarılması, nesnel veya öznel olarak değerlendirilmesi duygu analizi sonucunda ortaya çıkmaktadır. Duygu analizi temel olarak bir metin işleme işlemi olup verilen metinin duygusal olarak ifade etmek istediği sınıfı belirlemeyi amaçlar. DA'nın ilk çalışmaları duygusal kutupsallık olarak geçmekte olup verilen metni olumlu, olumsuz ve nötr olarak sınıflandırmayı amaçlamaktadır. Daha sonra yapılan çalışmalar farklı duygu durumlarını belirten analizlere de izin vermiştir.

Duygu analizi uygulamaları birçok alanda kullanılmaktadır. Tüketici hizmetleri, sağlık, siyasal seçimler, sosyal organizasyonlar, finans hizmetleri alanları örnek verilebilir.

2.2.1 Duygu analizi araştırma seviyeleri

Duygu analizinde farklı araştırma seviyeleri kullanılmıştır. Birbirinden değişik 3 farklı seviyeden bahsedilebilir.

2.2.1.1 Döküman seviyesi

Bu yöntemle, bir dökümana göre tüm düşünceyi negatif ya da pozitif çıkaran yöntemdir. Bu analiz biçimi birden çok ürünü ya da durumu karşılaştıran dokümanlar için uygun değildir. Çünkü birden fazla durum ya da ürün karşılaştırması yapıldığında birden fazla sonuç çıkması beklenmektedir. Oysaki bu yöntem ile sadece bir sonuç çıkmaktadır.

2.2.1.2 Cümle seviyesi

Cümle seviyesinde DA, her cümlenin duygularını tek tek analiz eder. Pozitif, negatif ya da nötr sonucunu çıkarma işlemi yapılmaktadır. Nötr genelde duygu veya fikir belirtmeyen cümle olduğu anlamına gelmektedir.

2.2.1.3 Varlık ve görüş seviyesi

Doküman seviyesi ve cümle seviyesi analizleri insanların neyi sevip neyi sevmediğini tam olarak keşfedemez. Varlık ve Görüş Seviyesi ise küçük taneli analiz yapmayı sağladığından daha doğru analizler yapmayı sağlamaktadır. Bu analiz yöntemiyle dil yapılarına bakmak yerine direkt duygu ile ilgilenir. Bu durumda da Doküman ve cümle seviyesi analizlerden farklı olmasını sağlamaktadır.

2.2.2 Duygu analizi yaklaşımları

DA yaklaşımları kısaca sözlük temelli yaklaşım, makine öğrenimi temelli yaklaşım ve ikisinin karışımı olan hibrit temelli yaklaşım olarak adlandırılabilir. Makine öğrenimi temelli yaklaşımı, makine öğrenimi algoritmalarının kullanımını içerir. Sözlük temelli yaklaşım, bilinen ve önceden derlenmiş teknik terimlerin bir koleksiyonunu ifade eden duygu sözcüklerine dayanır. Hibrit yaklaşım ise her iki yaklaşımı (makine öğrenimi ve sözlük temelli) birleştirerek bir sonuç elde etmeye dayanır.

2.2.2.1 Sözlük temelli yaklaşım

Sözlük tabanlı yaklaşım, metni analiz etmek için kullanılan duygu sözlüğünü bulmayı içerir. Sözlük tabanlı yaklaşım, her terime karşılık gelen duyarlılık puanlarına sahip terimler içeren bir sözlük kullanır. Terim, tek bir cümle, kelime veya deyim ile ilgili olabilir. Terimin sözlükte varlığı veya yokluğu duygusunu tanımlayacaktır. Sözlük tabanlı yaklaşım ise kendi içerisinde ikiye ayrılır. Bu yaklaşımlar şu şekildedir:

1. Sözlük Tabanlı Yaklaşım

2. Kütüphane Tabanlı Yaklaşım

2.2.2.2 Makine öğrenimi temelli yaklaşım

MÖ yaklaşımını kullanan metin sınıflandırması, kabaca gözetimli ve gözetimsiz öğrenim olarak ikiye ayrılabilir. Gözetimli öğrenme, çok sayıda etiketli veri kümesinin kullanılmasına izin verirken gözetimsiz öğrenme, toplanmamış veri kümelerinin kullanımını içerir. Gözetimsiz öğrenim, etiketli veri setinin bulunması zor olan durumlarda kullanılır. MÖ temelli yaklaşımı ile yapılan çalışmaların çoğunluğunda gözetimli öğrenme sınıflandırması kullanıldığı görülmektedir. Genellikle film yorumları ve Twitter yazıları için çalışmalar literatürde yer almaktadır. Makine öğrenme temelli tekniklerde iki tip veri setine ihtiyaç vardır. Bunlardan ilki eğitim veri seti ikincisi ise test veri setidir. Sınıflandırıcılar bilgi çıkarımı için etiketlenmiş veri kümesine ihtiyaç duyarlar, böylece bu veri kümesi sınıflandırıcıyı eğitmek için kullanılır. Bu veri kümesi, eğitim veri kümesi olarak bilinir. Ancak, sınıflandırıcılar tahmin için kullanıldığından, sınıflandırıcının performansını etiketlenmemiş veri kümesi kullanarak değerlendirmek mümkün değildir; aynı eğitim veri kümesini kullanmak da performans değerlendirmesi için iyi bir yöntem değildir, çünkü eğitim sırasında sınıflandırıcı bu girdileri kullanmıştır ve bu değerlendirme tahmin performansını ölçmez. Dolayısıyla, daha doğru ölçümler elde etmek için, etiketlenmiş veri kümesi iki gruba ayrılır. İlk grup eğitim aşaması için kullanılırken diğeri sınıflandırıcıyı test etmek için kullanılır. Bu yaklaşımda, değerlendirme için kullanılan veriler eğitime dahil edilmez, ancak bu veri kümesindeki girdilerin gerçek sınıfları bilinir; böylece doğru değerlendirme ve ölçümler için test veri kümesi sınıflandırıcı tarafından işlenir ve sınıflandırıcının tahminleri ile gerçek etiketler karşılaştırılır.

2.3 Makine Öğrenmesi

Makine Öğrenmesi (MÖ), matematiksel ve istatistiksel işlemler ile veriler üzerinden çıkarımlar yaparak tahminlerde bulunan sistemlerin bilgisayarlar ile modellenmesidir. Günümüzde MÖ için bir çok yaklaşım ve algoritma mevcuttur. MÖ temelde öğrenme yöntemine göre üç gruba ayrılır.

- Gözetimli Öğrenme
- Gözetimsiz Öğrenme
- Takviyeli Öğrenme

MÖ'nin başlıca uygulama alanları, makine algılaması, bilgisayarlı görü, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, biyoinformatik, kredi kartı dolandırıcılığı denetimi, borsa çözümlemeleri, DNA dizilerinin sınıflandırılması, konuşma ve el yazısı tanıma, bilgisayarlı görmede nesne tanıma, oyun oynama gibi sıralanabilir. MÖ'nin birçok alanda başarılı sonuçlar vermesi, DDİ için kullanılmasını da popülerleştirmiştir.

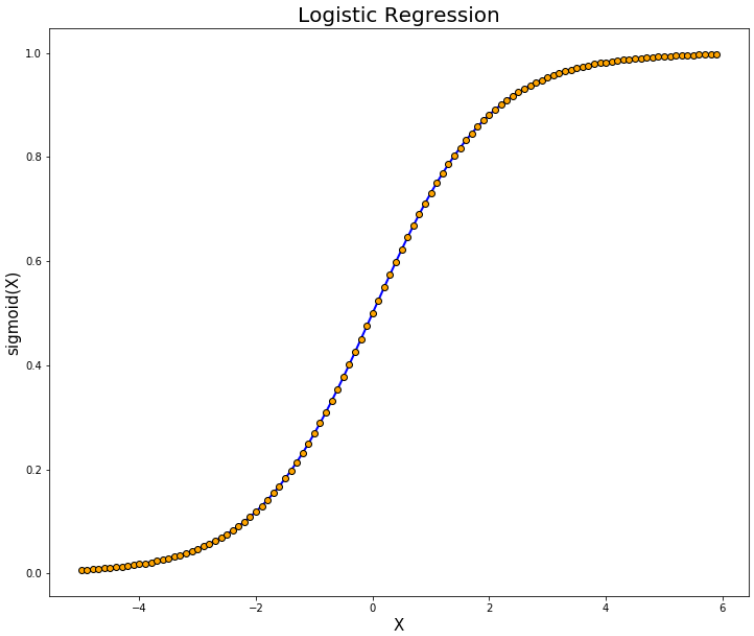
Bu çalışmada MÖ tekniklerinden Lojistik Regresyon, SGD ve Multinomial Naive Bayes kullanılmıştır. SGD daha önceki birçok çalışmada en iyi başarımı sağlayan teknik olmakla beraber Multinomial Naive Bayes sınıflandırıcı bazı alanlarda gayet başarılı sonuçlar verebilmektedir.

2.3.1 Lojistik regresyon - Logistic regression

Lojistik Regresyon sınıflandırma işlemi yapmaya yarayan bir regresyon yöntemidir. Kategorik veya sayısal verilerin sınıflandırılmasında kullanılır. Lojistik Regresyon'un amacı, iki yönlü karakteristiği ile ilgili bir dizi bağımsız değişken arasındaki ilişkiyi tanımlamak için en uygun modeli bulmaktır.

Lojistik Regresyon'da olaylar bağımsızdır. Bağımlı değişken ve bağımsız değişkenler arasında doğrusal bir ilişki varsaymaz, ancak açıklayıcı değişkenlerin logitleri ile yanıt arasındaki doğrusal ilişkiyi varsayar. Bağımsız değişkenler, orijinal bağımsız değişkenlerin güç terimleri veya bazı diğer doğrusal olmayan dönüşümleri bile olabilir. Bağımlı değişken normal dağılım göstermek zorunda değildir, ancak tipik olarak üstel bir aileden gelen bir dağılım varsayar (Binom, çoklu terim, normal...). Hataların bağımsız olması gerekir ancak normal dağılmaz. Parametreleri tahmin etmek için sıradan en küçük kareler yerine (OLS) yerine maksimum olasılık tahmini (MLE) kullanır ve bu nedenle büyük örneklem yaklaşımlarına dayanır.

Lojistik Regresyon, bağımlı değişken binary olduğunda yürütülecek uygun regresyon analizidir. Tüm regresyon analizlerinde olduğu gibi, lojistik regresyon da bir tahmini



Şekil 2.1 : Lojistik Regresyon

analizdir. Lojistic Regresyon, veriyi tanımlamak ve bir bağımlı ikili değişken ile bir veya daha fazla nominal, sıra arası, aralık veya oran seviyesinde bağımsız değişkenler arasındaki ilişkiyi açıklamak için kullanılır.

Lojistik Regresyon analizinin merkezinde, bir olayın log oranını tahmin eden görev bulunur. Matematiksel olarak, lojistik regresyon aşağıdaki gibi tanımlanmış çoklu doğrusal regresyon fonksiyonunu tahmin eder:

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + ... + b_kX_k \tag{2.1}$$

Denklem 2.1' de görüleceği üzere p; karakteristik özelliğinin var olma olasılığıdır.

$$olasilik = (p)/(1-p) \tag{2.2}$$

Denklem 2.2' de görüleceği üzere; olasılık, karakteristik özelliğin var olma olasılığının, karakteristik özelliğin var olmama olasılığına bölünmesi ile elde edilir.

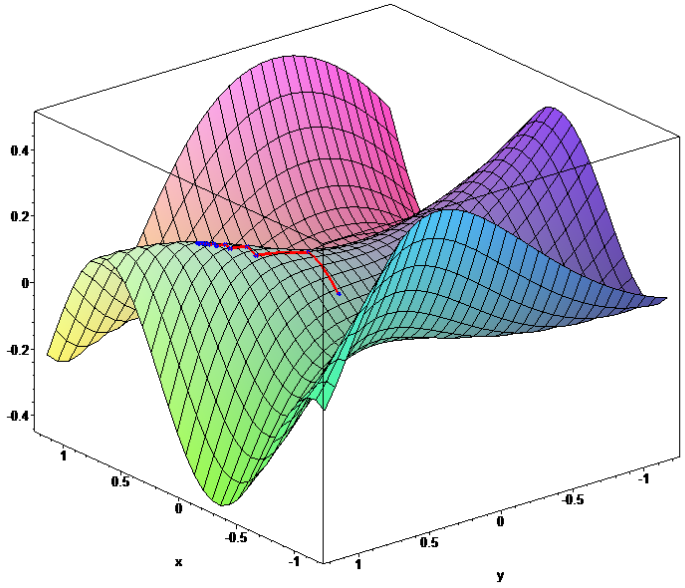
$$logit(p) = \ln \frac{p}{1-p} \tag{2.3}$$

Karekök hataların toplamını en aza indirgeyen parametreleri seçmek yerine (sıradan regresyon gibi), lojistik regresyonda tahmin, örnek değerlerin gözlem olasılığını en yükseğe çıkaran parametreleri seçer.

2.3.2 Olasılıksal dereceli azalma - stochastic gradient descent

Olasılıksal Dereceli Azalma (SGD), Destek Vektör Makineleri (SVM) ve Lojistik Regresyon gibi konveks kayıp fonksiyonları altında lineer sınıflandırıcıların ayırt edici öğreniminde basit ama çok etkili bir yaklaşımdır. SGD, makine öğrenmesi alanında uzun zamandır var olsa da, son zamanlarda büyük ölçekli öğrenme bağlamında dikkat çekmektedir.

SGD, metin sınıflandırmasında ve doğal dil işlemede sıklıkla karşılaşılan büyük ölçekli ve seyreltilmiş makine öğrenmesi problemlerine başarıyla uygulanmıştır.



Şekil 2.2 : Olasılıksal Dereceli Azalma - SGD

SGD, uygulama kolaylığı ve verimlilik bakımından diğer algoritmalara göre kolaylılık sağlar. Fakat, düzenlileştirme parametresi ve yineleme sayısı gibi bir dizi hiper parametre gerektirdiği ve özellik ölçeklemeye duyarlı olması da algoritmanın dezavantajı denilebilir.

SGD'de amaç her bir adımda veri setindeki her bir örneği gezdikten sonra güncellemektir.

Standart dereceli azalma algoritması objektif $\Theta ve J(\Theta) parametreleriniuekildegncellenir$:

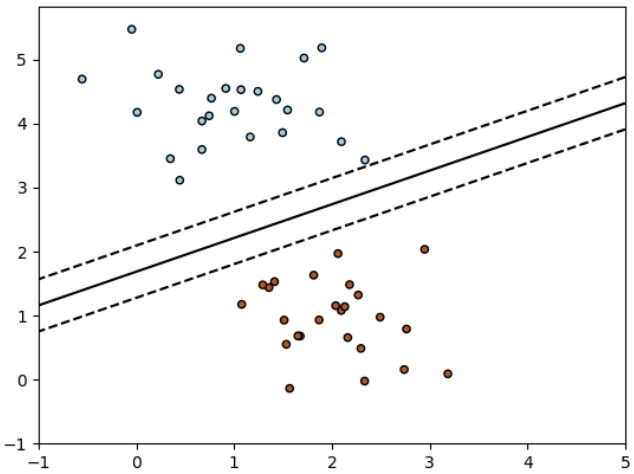
$$\Theta = \Theta - \alpha \nabla_\theta E[J(\Theta)] \tag{2.4}$$

Denklem 2.4'te oluşan beklenti, tam eğitim seti üzerinden maliyet ve dereceyi değerlendirerek yaklaşıklaştırılır. SGD, güncellemede ki beklentiyi ortadan kaldırır ve yalnızca bir veya birkaç eğitim örneği kullanılarak parametrelerin değişim derecesini hesaplar. Yeni güncelleme;

$$\Theta = \Theta - \alpha \nabla_\theta J(\Theta; x^i, y^i) \tag{2.5}$$

Genellikle SGD'deki her parametre güncellemesi, tek bir örneğe kıyasla birkaç eğitim örneği veya minibatch olarak hesaplanır. Bunun nedeni iki yönlüdür; önce parametre güncellemesindeki değişimi azaltır ve daha istikrarlı yakınsama getirilir.İkincisi ise, hesaplamanın maliyetinin ve derecenin iyi vektörize edilmiş bir hesaplamada kullanılması gereken yüksek düzeyde optimize edilmiş matris işlemlerinden yararlanmasını sağlar. [10]

SGD ile ilgili son nokta, veriyi algoritmaya sunmamızın sırasındadır. Veriler, anlamlı bir sıra ile verilirse dereceyi bozabilir ve zayıf yakınsamaya yol açabilir. Bunu önlemek için genellikle verilerin herbir eğitim aşamasından önce rastgele karıştırılması sağlanmaktadır.



Şekil 2.3 : Olasılıksal Dereceli Azalma Sınıflandırıcısı - SGDClassifier

2.3.3 Naive bayes

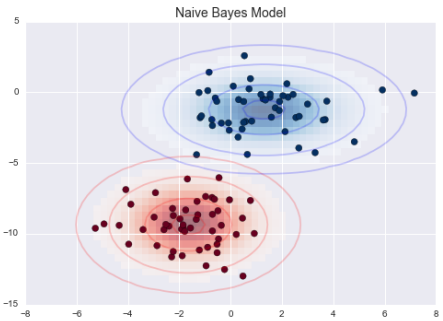
Naive Bayes bağımsızlık önermesini kullanan basit bir istatistiksel sınıflandırıcıdır. Bu önerme "sınıf koşullu bağımsızlık" olarak adlandırılır ve sınıflandırmada kullanılacak her bir öznitelik ya da parametrenin istatistiksel açıdan bağımsız olması gerekliliğini ifade eder. Naive Bayes sınıflandırıcının avantajlarından birisi diğer sınıflandırıcılara göre çok az miktarda eğitim kümesi ile gerekli parametreleri tahmin edebilmesidir. Bunun nedeni, özniteliklerin bağımsızlığı sayesinde, tüm kovaryans matrisinin yerine sadece ilgili sınıfa ait değişkenlerin kovaryansı hesaplanıyor olmasıdır. Naive Bayes algoritması her özniteliğin sonuca olan etkilerinin olasılık olarak hesaplanması temeline dayanmaktadır.

$$P(S_i|x) \times p(x) = p(x|S_i) \times P(S_i) \tag{2.6}$$

$$P(x|S_i)P(S_i) > p(x|S_j)P(S_j), \forall j \neq, i \tag{2.7}$$

$$P(S_i) \prod_{k=1}^{L} P(x_k|S_i) > p(S_j) \prod_{k=1}^{L} P(x_k|S_j) \tag{2.8}$$

Denklem 2.6, 2.7 ve 2.8' de P(S_i) ve P(S_j) sırasıyla sınıflandırma yapılacak i ve j sınıflarının öncel olasılıkları, P(S_i|x) ve P(S_j|x), sırasıyla i ve j sınıflarının ardıl olasılıkları, P(x) x'in olasılık yoğunluk fonksiyonu ve P(x|S_i) x'in i sınıfına bağlı koşullu olasılık yoğunluk fonksiyonu olsun. Denklem 2.7' ye göre x örneği sınıf i'ye aittir. Eğer Bayes karar teoremine özniteliklerin istatistiksel olarak bağımsızlığı varsayımı eklenirse bir Naive Bayes sınıflandırıcısı (Denklem 2.8) elde edilir.



Şekil 2.4 : Naive Bayes

3. METODOLOJİ

Bu bölümde yapılan tez çalışması ve DA için kullanılan metotlar hakkında detaylı bilgi verilmektedir.

3.1 Veri Kümesi

Çalışmamda, günümüzde en çok kullanılan mikro blog sitesi olan Twitter'dan elde ettiğim tweetlerden oluşan bir veri kümesi kullandım. Twitter veri kümesi imla ve dil kuralları bakımından oldukça zayıf bir veri kümesidir. Sınırlı karakterlerle yazılan metinler, kullanılan kısaltmalar, harf eksiklikleri ve kuralsız kurulan cümle yapılarından dolayı DDİ için zor bir veri setidir. Kullanıcılar fikirlerini, görüşlerini 280 kararktere sığdırmak zorundadırlar. Bu da yanlış yazılmış cümlelere ve kısaltmalara sebebiyet vermektedir. DDİ'nin birçok seviyesinde bu metinler işlenirken, kurallı ve kontrollü metinlere nazaran daha düşük başarılar çıkarmaktadır. Twitter üzerinden elde edilen veri seti iki farklı yöntem ile oluşturulmuştur. İki yöntem sonucunda 10700 tweet bulunan bir veri seti meydana gelmiştir. Bu veri kümesi tamamen Türkçe dili ile atılan gönderileri içermektedir. Tweetleri belirli bir anahtar kelimeye göre arattım. Anahtar kelimeyi, Türkiye merkezli teknolojik iletişim operatör şirketi olan "Turkcell" seçtim.

Tablo 3.1 : Veri kümesi

Metot	Veri Sayısı
Tweepy	5350
Twint	5350

3.1.1 Tweepy

Tweepy, Python programlama dili ile Twitter API'sine erişmek için geliştirilmiş bir kütüphanedir. [11] Tweepy kullanarak Twitter API'a erişebilmek için yapılması gereken bir takım gereksinimler vardır. Bunlardan bir tanesi, https://developer.twitter.com/en/apply-for-access adresinden Standart API için başvuru talebi yapılması gerekmektedir. Benim yapmış olduğum başvuru olumlu karşılandıktan sonra veri setimi oluşturmak için Tweepy kütüphanesi kullanmaya

başladım. Tweepy kütüphanesinin bir dezavantajı sınırlı sürede sınırlı sayıda tweet kazıma işlemine izin veriyor olmasıydı. Veri setimin yarısını bu yöntem ile elde ettikten sonra bir diğer yönteme geçiş gerçekleştirdim.

3.1.2 Twint

Twint projesi, Francesco Poldi ve Cody Zacharias tarafından geliştirilmiş, Python programlama dilinde yazılmış ve Twitter API kullanmadan veri kazıma işlemi sunan gelişmiş bir açık kaynaklı OSINT aracıdır. [12] Twint kullanmamım asıl nedeni Twitter API' a bağlı olmadığı için sunmuş olduğu hizmetti. Bu hizmet sayesinde Tweepy kütüphanesi ile elde edebildiğim verinin fazlasını, daha kısa sürede elde edebiliyordum. Veri setimin kalan yarısını Twint projesi ile elde ettim.

3.2 Verilerin Etiketlenmesi

Yapılan DA çalışmalarında veri etiketlemek için değişik yollar kullanılmaktadır. Bu yollardan en ilkel olanı veri setini oluşturan kişi veya kişilerce etiketlemenin el ile yapılmasıdır. El ile etiketleme yapmanın bazı avantajları ve dezavantajları vardır.

Metinlerin içlerinde bulunan iğneleme, şaka yollu yapılan yorumlar gibi bilgisayar tarafından anlamı çıkarılması zor olan cümleler bulunur. Bu cümlelerin insan tarafından algılanması daha kolay olur. Bu cümlelerin el ile etiketlenmesi bir avantajdır.

DA çalışmalarında veri seti ne kadar büyürse başarı oranı o kadar artar. Büyüyen veri setlerinde el ile etiketleme işlemi oldukça güç hale gelir.

Çalışma boyunca 10700 metinden oluşan veri setini tek tek kontrol ederek etiketleme işlemini gerçekleştirdim. Etiketleme işleminde iki sınıf kullandım (pozitif, negatif). Etiketleme işlemi sonucunda 10700 verinin 5500 tanesinin pozitif, 5200 tanesinin negatif olduğu sonucu ortaya çıktı.

Tablo 3.2 : Etiketlenen veri kümesi

Etiket	Veri Sayısı
Pozitif	5500
Negatif	5200

3.3 Ön İşleme

Twitter ile yapılan DA çalışmalarında verilerin ham hali ile işlenebilmeleri çok mümkün değildir. Bazı sorunlardan dolayı bu metinlerin işlenmeden önce ön işlemden geçmeleri gerekmektedir. Tüm ön işlem adımları uygulandıktan sonra geriye kalan temiz veriler analiz edilecek olanlardır.

3.3.1 Özel karakterlerin kaldırılması

Veri setinin içerisinde Twitter'da çok fazla kullanılan , @ ve RT işaretleri mevcuttu. DA çalışmasına etki etmeyen bu işaretler kaldırılarak metinlerin daha temiz gözükmesi sağlandı. Noktalama işaretleri kaldırılarak, girdinin çok boyutluluğunun azaltılması sağlandı. Bu işaretler metine kayda değer bir bilgi katmadıkları gibi metinlerin işlenmesi aşamasında fazlalık oluştururlar.

3.3.2 Sayıların kaldırılması

Veri kümesinde ki sayılar olumlu ya da olumsuz duyguları temsil etmemektedir; ayrıca veri kümesinin uzunluğunu herhangi bir fayda olmadan uzatırlar. Metinden gereksiz sayıların çıkarılması vektörel temsil anlamında ciddi boyutta bir kolaylık sağlayacaktır.

3.3.3 Etkisiz kelimelerin çıkarılması

Etkisiz kelimeler, bir dilde çok sık kullanılan (Türkçe'de "bir", "bu", "şu" gibi) kelimelerdir. Bu kelimelerin gözardı edilmelerinin sebebi, hemen her yazıda geçtiklerinden ötürü DA sonuçlarına pozitif bir katkı sağlamamaları, hatta bu sonuçları negatif yönde etkileyip daha isabetsiz sonuçların dönmesine sebep olmalarıdır. Dolayısıyla, modelin karmaşıklığını azaltmak ve başarı oranını arttırmak için bu kelimeler kaldırılmalıdır.

Literatür çalışmalarında NLTK kütüphanesinin Türkçe Stop Words'lerinin kullanıldığı sıkça görülmektedir. Ben çalışmamda, Ahmet Aksoy tarafından yayınlanan [13] Türkçe Dolgu Sözcükleri çalışmasını kullandım.

3.3.4 İfadelerin kaldırılması

Twitter, çok fazla kesim tarafından kullanılan bir mikro blog sitesidir. Bu çeşitlilik, paylaşılan gönderilerde birçok farklılığa sebebiyet vermektedir. İnsanlar, gönderilerini zenginleştirmek ve duygularını daha iyi ifade etmek adına ifadeler (emoticons)

kullanırlar. İfadelere örnek olarak ":)", ":/", ":-)" verilebilir.Bu ifadelerin DA için anlamlı şeyler sunabilmesi için yazı haline getirilmeleri gerekmektedir. Bu çalışma kapsamında, ifadeler yazı haline getirilmeden metinden direkt kaldırılmıştır.

3.3.5 Linklerin kaldırılması

Gönderiler içerisinde DA için anlamlı olmayan, hiçbir duygu belirtmeyen linkler mevcuttur. Bu linklerin kaldırılması için, HTML ve XML belgelerini ayrıştıran Beautiful Soup Python paketini kullandım.

Tablo 3.3 : Ön İşlemler

İçerik	Ön İşlem
Özel Karakterler (#, @, RT)	Kaldırıldı
Sayılar	Kaldırıldı
Etkisiz Kelimeler (’ve’, ’ama’)	Kaldırıldı
İfadeler ("(:")	Kaldırıldı
Linkler ve URL’ler ("https://")	Kaldırıldı
Tekrar Eden Kelimeler ("asslaaaa")	Tekilleştirildi
Büyük Harfler	Küçük Harfe Çevrildi

Tweetlere uygulanan ön işlemler ve yapılan eylemler Tablo 3.3’te gösterilmiştir. Bu ön işlemler sonucunda verilerdeki gürültü azaltılmış ve işlenmiş tweetler oluşturulmuştur.

3.4 Kelime Köklerinin Bulunması

Sondan eklemeli bir dil olan Türkçede, teorik olarak bir kelime sonsuz sayıda ek alabilir ve aynı kökten türeyen çok sayıda kelime oluşturabilir. Bu eklerin her biri kelimenin anlamını, zamanını, durumunu ve türünü değiştirebilir. Bu yüzden kelime köklerinin kullanılması, özellikle sondan eklemeli diller için önemli bir ön işleme aşamasıdır.

Türkçe için kelime kökleri bulunurken, anlamsal değişiklik yaratan eklerin atılmaması ya da atılsa bile bu anlamın saklanabilmesi adına kelimelerin bir şekilde işaretlenmesine dikkat etmek gerekir.

Çalışmamda farklı kök bulma algoritmaları kullandım. Fakat hiçbirisi kök bulma konusunda hedeflediğim başarıyı yakalayamadı. Birbirlerinden çok farklı sonuçlar döndürmeselerde çalışmamın devamında Porter2 Stemming Algoritması olarakta

bilinen SnowballStemmer'ı kullandım. Porter Stemmer'dan daha agresif olmakla birlikte aynı zamanda daha başarılı sonuçlar vermektedir.

3.5 Öznitelik Temsili

DA'inde önemli bir nokta öznitelik çıkarımıdır. Özniteliklerin seçilmesi sınıflandırıcıların performansı açısından çok önemlidir. Öznitelik metotlarından TF-IDF ve Bag of Words metotları literatürde sıkça kullanılmaktadır.

3.5.1 Bag of words

Kelime çantası olarak bilinen bu model DDİ'de kullanılan basitleştirici bir temsil biçimidir. Bu modelde bir metin kelimelerin çantası halinde temsil edilir. Çoksallık tutulurken, dil bilgisi ve kelime sırası hataları göz ardı edilir.

Çalışmamda terim-sayma amacıyla; bir metnin dökümanı koleksiyonunu terim sayısı matrisine dönüştüren CountVectorizer sınıfını kullandım.

3.5.2 Terim frekansı x Ters belge frekansı / TFxIDF

Terim sıklıklarının sayılması ile ilgili en önemli sorun, sık kullanılan terimlerin dökümanda baskın olmaları ve artık dökümanı temsil eder hale gelmeleridir. Bu terimler çok değerli bilgi içermeseler bile, özellik kümesindeki diğer terimlerin etkisiz hale gelmesine neden olurlar. Bu problemi çözmek için "Terim Frekansı x Ters Belge Frekansı" anlamına gelen "TF x IDF" modelini ve skorlama yöntemini kullanabiliriz.

Hesaplamada iki ölçüt kullanılır:

1. Terim Sıklığı (TF)

2. Ters Belge Sıklığı (IDF)

TFxIDF'in j. dökümandaki i. terim için denklemi şu şekildedir:

$$TFxIDF = TF(i,j) * IDF(i) \tag{3.1}$$

Denklem 3.1'de görüldüğü gibi TF(i,j), dökümandaki i. terimin sıklığının dökümandaki toplam terim sayısına bölünmesiyle elde edilir. IDF(i) ise, toplam döküman sayısının, i. terimi içeren döküman sayısına bölümünün logaritmik sonucudur.

Çalışma sürecinde dökümanları bir TFxIDF özellik matrisine dönüştürmek için TfidfVectorizer kullandım.

3.6 Makine Öğrenmesi

DA'nde önemli bir nokta makine öğrenmesi sınıflandırıcısının seçilmesidir. Veri kümesinin karakteristiğine göre en uygun, hızlı ve yüksek başarımlı sınıflandırıcıları seçmek önemli bir konudur. Yapılan çalışmalarda pek çok algoritma denenmiş, herkesin paylaşımına açık olan bir takım veri setlerinde uygulanmıştır. Son zamanlarda gelişen teknoloji sayesinde derin öğrenme algoritmaları da kendilerine bu algoritmalar arasından yer edinmiştir.

Çalışmam dahilinde derin öğrenme algoritmaları kullanmayı düşündüm, fakat bu düşüncemden vazgeçtim. Bu çalışmada sınıflandırıcı olarak Logistic Regression, Naive Bayes ve SGD algoritmaları kullandım.

4. BAŞARI

MÖ algoritmalarının başarımını ölçmek için farklı yollar mevcuttur. Doğruluk, duyarlılık, geriçağırım, F-ölçütü veya kappa değerleri bunlardan bazılarıdır. Bu ölçütlerin belirlenmesinde bazı parametreler kullanılır. Bu parametrelerden bazıları şunlardır:

- Doğru Pozitif / True Positive (TP): Gerçekte pozitif olan ve sınıflandırıcı tarafından pozitif tahmin edilen durumlar
- Yanlış Pozitif / False Positive (FP): Gerçekte negatif olan fakat sınıflandırıcı tarafından pozitif tahmin edilen durumlar
- Doğru Negatif / True Negative (TN): Gerçekte negatif olan ve sınıflandırıcı tarafından negatif tahmin edilen durumlar
- Yanlış Negatif / False Negative (FN): Gerçekte pozitif olan fakat sınıflandırıcı tarafından negatif tahmin edilen durumlar

4.1 Başarı Ölçütleri

4.1.1 Doğruluk

Doğruluk, tahmin sonucunun gerçek değerine yakınlık derecesine denir. Yapılan doğru tahminlerin, yapılan toplam tahmin sayısına oranıdır. Doğruluk, sınıflandırma başarımı için ortak bir ölçüttür.

$$Dogruluk = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

Denklem 4.1'de görüldüğü üzere doğruluk; doğru sınıflandırılmış tweetlerin, tüm tweet sınıflarının toplamına bölümüyle bulunur.

4.1.2 Duyarlılık

Duyarlılık, doğru sınıflandırılmış pozitif tweetlerin, pozitif olarak sınıflandırılmış tüm tweetlere oranıdır. Sınıflandırıcının, veri setinden bağımsız olarak belirli bir sınıfta ne kadar başarım gösterdiğini gösterir. Ancak sınıflandırıcı tarafından tahmin edilen yanlış negatifleri dikkate almaz. Bu olumsuzluğu gidermek için genellikle geriçağırım(recall) ile birlikte kullanılır.

$$Duyarlilik = \frac{TP}{TP + FP} \tag{4.2}$$

Denklem 4.2'de görüldüğü üzere duyarlılık; doğru sınıflandırılmış pozitif tweetlerin, pozitif olarak sınıflandırılmış tüm tweetlere oranıdır.

4.1.3 Geri çağırma

Geri çağırma, doğru sınıflandırılmış pozitif tahminlerin ne kadar başarılı sınıflandırıldığını gösterir.

$$Gericagirim = \frac{TP}{TP + FN} \tag{4.3}$$

Denklem 4.3'te görüldüğü üzere geri çağırma, doğru sınıflandırılmış pozitif tweetlerin, doğru olarak sınıflandırılmış tüm tweetlere oranı ile bulunur.

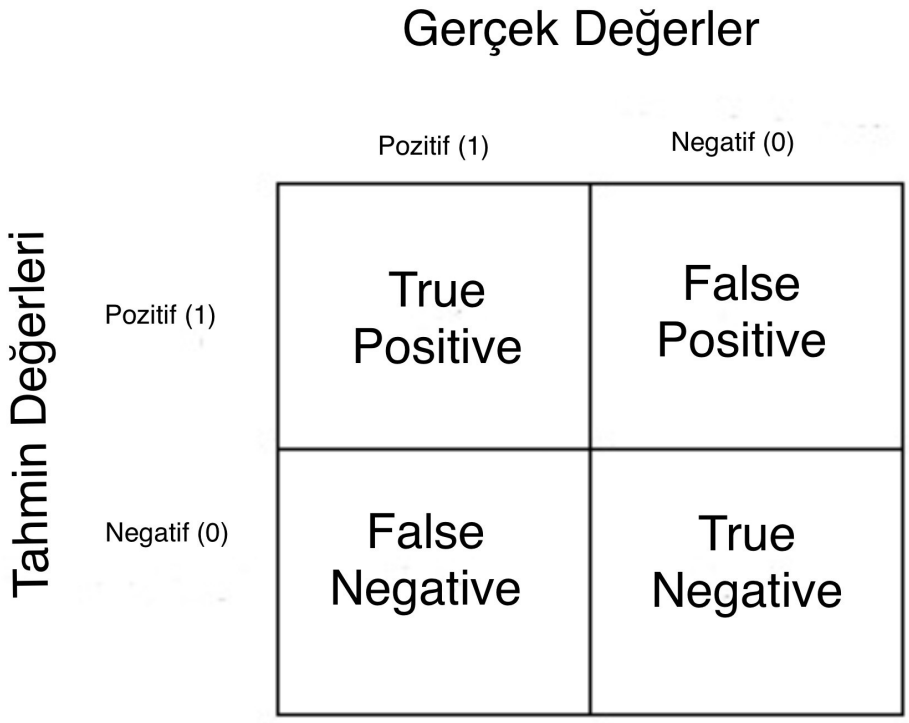
4.1.4 F1 skoru

F1 skoru, sınıflandırıcının doğruluğunu ölçmek için kullanılır. F1 skoru, hem duyarlılık hem de geri çağırma kullanarak bir testin başarısının ölçümünü sağlayabilir. F1 skoru ekstrem durumları cezalandırmak için aritmetik ortalama yerine harmonik ortalamayı kullanır.

$$F1 - Skor = \frac{2 * GeriCagirma * Duyarlilik}{GeriCagirma + Duyarlilik} \tag{4.4}$$

4.1.5 Karmaşıklık matrisi - Confusion matrix

Karmaşıklık matrisi (confusion matrix), MÖ ve özellikle istatistiksel sınıflandırma problemlerinde, hata matrisi olarak da bilinen, bir algoritmanın performansının görselleştirilmesine izin veren özel bir tablo düzenidir.



Şekil 4.1 : Karmaşıklık Matrisi - Confusion Matrix

5. SONUÇ VE DEĞERLENDİRME

5.1 Uygulama

Bu çalışmada, Twitter üzerinden Tweepy ve Twint modülü ile "Turkcell" anahtar kelimesine sahip 10700 adet metinden oluşan veri seti oluşturulmuştur. Oluşturulan veri seti el ile pozitif ve negatif olmak üzere 2 sınıfa etiketlenmiştir. Veri setindeki ham halde bulunan metinler, tokenlere ayrıştırılıp normalize edilerek belirli ön işleme adımlarından geçirilmiş ve doğal dil araç takımı(NLTK) altında bulunan Snowball Stemmer ile kök haline getirilmiştir. Bu aşamadan sonra öznitelikleri çıkartılmış ve vektör haline dönüştürülmüşlerdir. Vektörleştirme işlemi kelime çantası modeli (bag of words) ve terim frekansı x ters belge frekansı (tfxidf) ile gerçekleştirilmiştir. Sınıflandırma modeli için 3 algoritma seçilmiştir. Bu algoritmalar; Lojistik Regresyon (Logistic Regression), Olasılıksal Dereceli Azaltma (Stochastic Gradient Descent - SGD) ve Naive Bayes'tir. Kurulan modeller üzerinde elde edilen başarı oranları aşağıda verilmiştir.

Tablo 5.1 : Başarı

MÖ Modeli	Öznitelik	Duyarlılık	Geriçağırma	F1 Skor	Doğruluk
LR	BoW	0.66	0.63	0.62	0.6327
LR	TFxIDF	0.64	0.61	0.59	0.6135
SGD	BoW	0.65	0.64	0.64	0.6434
SGD	TFxIDF	0.65	0.63	0.63	0.6450
NB	BoW	0.65	0.63	0.63	0.6364
NB	TFxIDF	0.64	0.63	0.62	0.6322

Farklı öznitelikler ile kurulan Mö modellerinin başarıları Tablo 5.1'de görülmektedir. Bu modellerin karmaşıklık matrisleri ise aşağıdaki gibidir:

Tablo 5.2 : Lojistik Regresyon BoW Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	908	179
Negatif	607	446

Tablo 5.2'de BoW ile Lojistik Regresyon modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %63.27'lik bir doğruluk oluştuğu görülüyor.

Tablo 5.3 : Lojistik Regresyon TFxIDF Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	929	158
Negatif	669	384

Tablo 5.3'te TFxIDF ile Lojistik Regresyon modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %61.35'lik bir doğruluk oluştuğu görülüyor.

Tablo 5.4 : SGD BoW Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	830	257
Negatif	506	547

Tablo 5.4'te BoW ile Olasılıksal Dereceli Azalma (SGD) modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %64.34'lük bir doğruluk oluştuğu görülüyor.

Tablo 5.5 : SGD TFxIDF Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	849	238
Negatif	543	510

Tablo 5.5'te TFxIDF ile Olasılıksal Dereceli Azalma (SGD) modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %64.50'lik bir doğruluk oluştuğu görülüyor.

Tablo 5.6 : Naive Bayes BoW Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	836	251
Negatif	527	526

Tablo 5.6'da BoW ile Naive Bayes modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %63.64'lük bir doğruluk oluştuğu görülüyor.

Tablo 5.7 : Naive Bayes TFxIDF Karmaşıklık Matrisi

.	Pozitif	Negatif
Pozitif	851	236
Negatif	551	502

Tablo 5.7'de TFxIDF ile Naive Bayes modeli oluşturarak karşımıza çıkan karmaşıklık matrisini görüyoruz. Bu matris sonucunda %63.22'lik bir doğruluk oluştuğu görülüyor.

5.2 Sonuç

Bu çalışmada, iki farklı duygu halinin, iki farklı yöntem (BoW ve TFxIDF) kullanılarak üç farklı makine öğrenmesi algoritması ile analizleri yapılmıştır. Sınıflandırma aşamasında LR, SGD ve NB algoritmaları kullanılmıştır. Veri seti, 5500 pozitif, 5200 negatif tweetten oluşmaktadır. Elde edilen sonuçlara göre en yüksek başarı %64.50 ile TFxIDF kullanılarak SGD sınıflandırıcısında görülmüştür. SGD sınıflandırıcının diğer sınıflandırıcılardan daha iyi olduğu anlaşılmıştır. TFxIDF ve BoW yöntemleri arasında çok bir başarı farkı olmadığı söylenebilir.

5.3 Tartışma

Çalışma sonucunda, elde edilen başarı yüzdesinin diğer çalışmalara göre daha düşük düzeyde kaldığı görülmüştür. Bunun sebebinin, veri setinin iyi olmamasından ve gerçekleştirilen ön işleme adımlarından dolayı olduğunu düşünmekteyim. Veri setinin iyileştirilmesi ve ön işleme adımlarının daha tutarlı yapılması sonucunda elde edilen başarı yüzdesi arttırılabilir.

Kaynakça

[1] **Meral, M. ve Diri, B.** (2014). IEEE 22nd Signal Processing and Communicatios Applications Conference, IEEE, Turkey.

[2] **Çoban, O.**, **Özyer, B. ve Özyer, G.T.** (2015). Sentiment Analysis for Turkish Twitter Feeds, Signal Processing and Communications Applications (SIU), IEEE, IEEE, Turkey.

[3] **Kaynar, O., Y.M.G.Y. ve Albayrak, A.** (2016). Makine Öğrenmesi Yöntemleri ile Duygu Analizi, Sentiment Analysis with Machine Learning Techniques, IDAP, Turkey.

[4] **Wanh, H., C.D.K.A.B.F.N.S.** (2012). A System for Real-time Twitter Sentiment Analysis of 2012 U.S.Presidential Election Cycle, cilt 68, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistic, s.115–120.

[5] **Santos, N. ve Gatti, M.** (2014). Deep Convolutional Neural Networks forSentiment Analysis of Short Texts, 25, Proceedings of COLING 2014, Ireland, s.69–78.

[6] **Rosenthal, S., F.N. ve Nakov, P.** (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter, Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), Canada, s.502–518.

[7] **Ortigosa, A.**, **Martin, J. ve Carro, R.**, (2014). Sentiment analysis in Facebook and its application to e-learning, Computers in Human Behavior, s.527–541.

[8] **Vural, G.A.**, **Cambazoğlu, B.B.**, **Şenkul, P. ve Özge Z. Tokgöz** (2013). A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish, 437–445.

[9] **Şimşek, M.U. ve Özdemir, S.** (2012). A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish, cilt 6, Application of Information and Communication Technologies (AICT).

[10] SGD, scikit-learn, `https://scikit-learn.org/stable/modules/sgd.html`.

[11] Tweepy, `[http://docs.tweepy.org/en/latest]`.

[12] Twint, `[https://github.com/twintproject/twint]`.

[13] Türkçe Dolgu Kelimeler, `[https://github.com/ahmetax/trstop/]`.

EKLER

EK A.1 : Kodlar

EK A.1

Bu kısımda, tez çalışmam sürecinde yaptığım DA projesinde önemli yerlerin kaynak kodlarını olabildiğince sade ve anlaşılır şekilde anlatmaya çalışacağım.

```
import pandas as pd
import numpy as np
import nltk
from bs4 import BeautifulSoup
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from collections import Counter
from nltk.tokenize.toktok import ToktokTokenizer
from snowballstemmer import TurkishStemmer
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression,SGDClassifier
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score
from sklearn import metrics
import re
```

Şekil A.1 : Tez Çalışması Kapsamında Kullandığım Ana Kütüphaneler

Tez çalışmamda birçok metot ve kütüphane kullandım. Kullandığım kütüphaneler Şekil A.1'de gösterilmiştir. Burada önemli kütüphanelerin üzerinden tekrar geçmem gerekirse; DDİ kütüphanesi olan nltk, kök bulma işlemimi gerçekleştirdiğim SnowballStemmer ilk akla gelen kütüphanelerdir. Ayrıca veri setini eğitim ve test seti olarak ikiye bölmemde, vektörleştirme işlemlerimde, makine öğrenmesi sınıflandırıcı modellerinde, karmaşıklık matrisi ve başarım ölçütü hesaplamada kullandığım sklearn kütüphanesini de tez çalışmamda çok faydalı olmuştur.

```
class TwitterSentimentAnalyser:
    def __init__(self, consumer_key, consumer_secret, access_token, access_token_secret, keyword, tweetCount):
        self.keyword = keyword
        self.consumer_key = consumer_key
        self.consumer_secret = consumer_secret
        self.access_token = access_token
        self.access_token_secret = access_token_secret
        self.tweetCount = tweetCount

    def getTwitterData(self):
        tweets_list = []

        for tweet in api.searc(q = self.keyword, count = self.tweetCount, lang = "tr-tr"):
            tweets_list.append((tweet.created_at, tweet.id, tweet.text))

            self.tweets = pd.DataFrame(tweets_list, columns = ['Date', 'Id', 'Text'])


twst = TwitterSentimentAnalyser(consumer_key = consumer_keys, consumer_secret = consumer_secrets,
                                access_token = access_tokens, access_token_secret = access_token_secrets,
                                keyword = "turkcell", tweetCount = 100)

twst.getTwitterData()
```

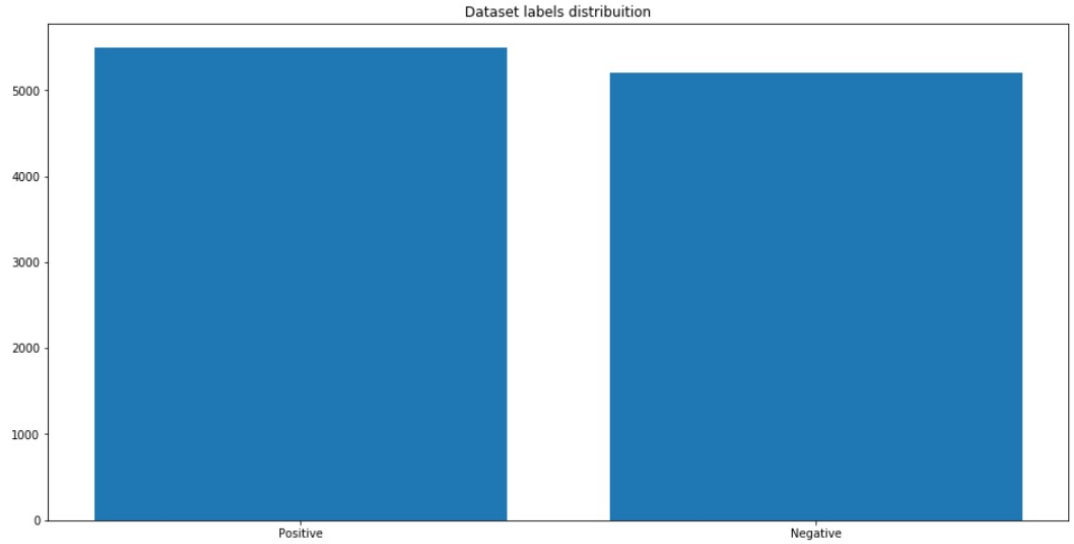
Şekil A.2 : Twitter API ile Veri Seti Oluşturma

Veri setimin bir kısmını Twitter API ile Tweepy kütüphanesi kullanarak oluşturdum. Şekil A.2'de bu kütüphane kullanılarak nasıl veri elde edildiği görülmektedir. Twitter API'ın sunduğu kısıtlı imkan dolayısıyla tweetCount = 100 olarak ayarlanmıştır. Çünkü API 100'den fazla veri çekilmesine izin vermez. Veri elde etmek için bir diğer yöntem olarak Twint projesi kullanılmıştır. Tez çalışmasına Tweepy kütüphanesi kullanılarak başladım fakat getirdiği kısıtlamalardan dolayı başka bir yolla devam ettim.

```
target_tdS = Counter(turkcell_data.Sentiment)

plt.figure(figsize=(16,8))
plt.bar(target_tdS.keys(), target_tdS.values())
plt.title("Dataset labels distribuition")
```

```
Text(0.5, 1.0, 'Dataset labels distribuition')
```



Şekil A.3 : Veri Setinin Dağılımı

Şekil A.3'te oluşturulan veri setinin etiketlenmesi sonucundaki dağılımı görülmektedir. 10700 adet verinin, 5500 tanesi pozitif, 5200 tanesi negatiftir. Veri setinin oldukça dengeli olduğu gözüküyor.

```
def strip_html(text):
        soup = BeautifulSoup(text, "html.parser")
    return soup.get_text()

def remove_between_square_brackets(text):
    return re.sub('\[[^]]*\]', '', text)

def denoise_text(text):
    text = strip_html(text)
    text = remove_between_square_brackets(text)
    return text

def stemWord(text):
    return text.lower()

def removeUsernames(text):
    return re.sub('@[^\s]+', '', text)

def removeHashtags(text):
    return re.sub(r'#[^\s]+', ' ', text)

def removePunctuation(text):
    return re.sub(r'[^\w\s]', ' ', text)

def singleCharacterRemove(text):
    return re.sub(r'(?:^| )\w(?:$| )', ' ', text)

def stripEmoji(text):
    emoji = re.compile('[\U00010000-\U0010ffff]', flags=re.UNICODE)
    return emoji.sub(r'', text)
```

Şekil A.4 : Veri Seti Üzerinde Yapılan Bazı Ön İşlemler

Şekil A.4'te veri seti üzerinde yapılan ön işleme adımlarının bazıları görülmektedir. Bu adımlar her metin sınıflandırma işleminde yapılması gereken ilk adımlardır. Linklerin kaldırılması, ifadelerin silinmesi, kullanıcı adı ve sayıların kaldırılması, Twitter'a özgü '@', '#' işaretlerinin silinmesi, metinlerin normalizasyonun sağlanması gibi gürültü giderici işlemler yapılmıştır.

```
cv=CountVectorizer(min_df=0,max_df=1,binary=False,ngram_range=(1,3))

cv_train_tweets=cv.fit_transform(X_train)

cv_test_tweets=cv.transform(X_test)

#vocab=cv.get_feature_names()-toget feature names


tv=TfidfVectorizer(min_df=0,max_df=1,use_idf=True,ngram_range=(1,3))
tv_train_tweets = tv.fit_transform(X_train)
tv_test_tweets = tv.transform(X_test)
```

Şekil A.5 : CouuntVectorizer ve TfidfVectorizer

Şekil A.5'te CountVectorizer ve TfidfVectorizer görülmektedir. Metinlerin vektörleştirilmesinde bu iki yöntem kullanılmış. İki yöntemde de parametre olarak (1,3) gram kullanılmıştır.

```
lr=LogisticRegression(penalty='l2',max_iter=500,C=1.1,random_state=42)

lr_bow=lr.fit(cv_train_tweets,y_train)
print(lr_bow)

lr_tfidf=lr.fit(tv_train_tweets,y_train)
print(lr_tfidf)

lr_bow_predict=lr.predict(cv_test_tweets)
print(lr_bow_predict)

lr_tfidf_predict=lr.predict(tv_test_tweets)
print(lr_tfidf_predict)

lr_bow_score=accuracy_score(y_test,lr_bow_predict)
print("lr_bow_score :",lr_bow_score)

lr_tfidf_score=accuracy_score(y_test,lr_tfidf_predict)
print("lr_tfidf_score :",lr_tfidf_score)

lr_bow_report=classification_report(y_test,lr_bow_predict,target_names=['Positive','Negative'])
print(lr_bow_report)

lr_tfidf_report=classification_report(y_test,lr_tfidf_predict,target_names=['Positive','Negative'])
print(lr_tfidf_report)

cm_bow=confusion_matrix(y_test,lr_bow_predict,labels=[1,0])
print(cm_bow)

cm_tfidf=confusion_matrix(y_test,lr_tfidf_predict,labels=[1,0])
print(cm_tfidf)
```

Şekil A.6 : Lojistik Regresyon Model Kurulumu

Şekil A.6'da Lojistik Regresyon modelinin kurulması gösterilmektedir. Bu model kurulurken, penalt = "l2", $\max_iter = 500, C = 1.1 verandom_state = 42 seilmitir. Modelineniyihangiparametrelerdebaarßverdiillmemitir.$

Şekil A.7'de Lojistik Regresyon sınıflandırıcısının karmaşıklık matrisi görünmektedir.

	precision	recall	f1-score	support
Positive	0.71	0.42	0.53	1053
Negative	0.60	0.84	0.70	1087
accuracy			0.63	2140
macro avg	0.66	0.63	0.61	2140
weighted avg	0.66	0.63	0.62	2140

	precision	recall	f1-score	support
Positive	0.71	0.36	0.48	1053
Negative	0.58	0.85	0.69	1087
accuracy			0.61	2140
macro avg	0.64	0.61	0.59	2140
weighted avg	0.64	0.61	0.59	2140

Şekil A.7 : Lojistik Regresyon Karmaşıklık Matrisi

ÖZGEÇMİŞ

Ad Soyad: Ramazan ÖLMEZ

Doğum Tarihi ve Yeri: 13.02.1995 - Uşak

E-Posta: ramazanolmez@outlook.com.tr

ÖĞRENİM DURUMU:

- **Lise:** 2013, Şehit Abdülkadir Kılavuz Anadolu Öğretmen Lisesi
- **Lisans:** Pamukkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- Staj I:AdresGezgini A.Ş, 2020