T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE SAHTE FELAKET TWEETLERİNİN TAHMİNİ

Fatma KURŞUN

YÜKSEK LİSANS TEZİ

İstatistik Anabilim Dalı

İstatistik Programı

Danışman

Prof. Dr. Fatma Noyan Tekeli

Aralık, 2022

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE SAHTE FELAKET TWEETLERİNİN TAHMİNİ

Fatma KURŞUN tarafından hazırlanan tez çalışması 12.12.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstatistik Anabilim Dalı, İstatistik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Fatma NOYAN TEKELİ
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Fatma NOYAN TEKELİ, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Gülhayat GÖLBAŞI ŞİMŞEK, Üye
Yıldız Teknik Üniversitesi _____

Doç. Dr. Seher ARIKAN TEZERGİL, Üye
Marmara Üniversitesi _____

Danışmanım Prof. Dr. Fatma NOYAN TEKELİ sorumluluğunda tarafımca hazırlanan Makine Öğrenmesi Yöntemleri ile Sahte Felaket Tweetlerinin Tahmini başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Fatma KURŞUN

İmza

TEŞEKKÜR

Bugüne kadar eğitim hayatım boyunca gerçek bilginin peşinde koşmamı sağlayan başta danışmanım Prof. Dr. Fatma NOYAN TEKELİ ve bütün hocalarıma, ilk günden beri yanımda olan ve destek veren arkadaşım Alihan ÖZTÜRK'e teşekkürü borç bilirim.

Her zaman benim arkamda olan ve her şeyden önce bana iyi bir insan olmayı öğreten annem, babam ve kardeşlerime sonsuz teşekkür ederim.

Fatma KURŞUN

İÇİNDEKİLER

KISALTMA LİSTESİ v
ŞEKİL LİSTESİ vi
TABLO LİSTESİ vii
ÖZET viii
ABSTRACT x
1 GİRİŞ 1
1.1 Literatür Özeti 1
1.2 Tezin Amacı 3
1.3 Hipotez 3
2 MAKİNE ÖĞRENMESİ 4
2.1 Makine Öğrenmesi Nedir? 4
2.2 Gözetimli Öğrenme 5
2.3 Gözetimsiz Öğrenme 5
2.4 Doğal Dil İşleme 6
2.5 Sınıflandırma Modellerinin Değerlendirilmesi 7
3 MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN UYGULANMASI 10
3.1 Problemin Anlaşılması 10
3.2 Verinin Toplanması 10
3.3 Veri Önişleme 10
3.4 Değişken Mühendisliği 18
3.5 Makine Öğrenmesi Modellerinin Kurulması 20
3.5.1.1 Lojistik Regresyon 20
3.5.1.2 Rasgele Orman Yöntemi 22
3.5.1.3 Naive Bayes Yöntemi 24
3.5.1.4 XGBoost Yöntemi 26
3.6 Modellerin Değerlendirilmesi 27
4 SONUÇ VE ÖNERİLER 29
KAYNAKÇA 33
TEZDEN ÜRETİLMİŞ YAYINLAR 34

KISALTMA LİSTESİ

AUC	Area Under the Curve
FN	False Negative
FP	False Positive
IDF	Inverse Document Frequency
NLP	Natural Language Processing
ROC	Receiver Operating Characteristic
TF	Term Frequency
TN	True Negative
TP	True Positive

ŞEKİL LİSTESİ

Şekil 2.1 Öğrenme Süreci 4
Şekil 2.2 Karışıklık Matrisi 7
Şekil 2.3 AUC 9
Şekil 3.1 Veri bilgileri 11
Şekil 3.2 Target değişkeninin sayısı 12
Şekil 3.3 Keyword değişkenin en çok kullanılan kelimeleri 13
Şekil 3.4 Gerçek felaket içeren tweetlerin keyword sıralaması 13
Şekil 3.5 Sahte tweetlerdeki keyword sıralaması 14
Şekil 3.6 Lokasyona göre en çok atılan sahte tweetlerin sayısı 14
Şekil 3.7 Lokasyon değişkeni için veri temizleme fonksiyonu 15
Şekil 3.8 Text değişkeninin içeriği 16
Şekil 3.9 Yeni üç değişkenin oluşturulması 16
Şekil 3.10 Metin madenciliği 17
Şekil 3.11 Felaket tweetlerinde en çok geçen kelimelerin sırası 18
Şekil 3.12 Count-Vector 18
Şekil 3.13 TF-IDF 19
Şekil 3.14 Eğitim verisinin son hali 19
Şekil 3.15 Verinin eğitim-test olarak ayrılması 20
Şekil 3.16 Lojistik regresyon model eğitimi 21
Şekil 3.17 Lojistik regresyon modeli tahminleri 21
Şekil 3.18 Lojistik regresyon modeli çarpıklık matrisi grafiği 22
Şekil 3.19 Rasgele Orman modeli eğitimi 23
Şekil 3.20 Rasgele orman modeli tahminleri 23
Şekil 3.21 Rasgele Orman modelinin çarpıklık matrisi grafiği 24
Şekil 3.22 Naïve bayes modeli eğitimi 24
Şekil 3.23 Naïve bayes modeli tahminleri 25
Şekil 3.24 Naïve bayes modelinin çarpıklık matrisi grafiği 25
Şekil 3.25 XGBoost modeli eğitimi 26
Şekil 3.26 XGBoost modeli tahminleri 27
Şekil 3.27 XGBoost modelinin çarpıklık matrisi grafiği 27
Şekil 3.28 Naïve Bayes modelinin ROC grafiği 28
Şekil 4.1 Test verisi tahmini 29
Şekil 4.2 Sonuçların Kaggle'a yüklenmesi 30
Şekil 4.3 Sonuç 31

TABLO LİSTESİ

Tablo 2.1 Metrikler 8
Tablo 2.2 ROC 9
Tablo 3.1 Verinin ilk 5 satırı 11
Tablo 3.2 Sonuçların değerlendirilmesi 27

ÖZET

Makine Öğrenmesi Yöntemleri ile Sahte Felaket Tweetlerinin Tahmini

Fatma KURŞUN

İstatistik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Fatma NOYAN TEKELİ

Bu araştırmada Twitter kullanıcılarının attığı yangın sel deprem vb. felaket tweetlerinin doğru ya da sahte olduğunun araştırılması yapılacaktır. Günümüzde sosyal medya mecraları özellikle de Twitter çok önemli bir iletişim haline gelmiştir. İnsanlar herhangi bir olay karşısında bu mecradan kendi fikirlerini dile getirmekte ve haberleri buradan takip edebilmektedir. Fakat her zaman doğru bir kaynak olamamaktadır. Bu tez kapsamında bir kullanıcının attığı bir tweetin gerçek ya da sahte felaket tweeti olmadığının tahmini makine öğrenmesi yöntemleri kullanılarak tahmin edilecektir. Örnek verilecek olursa; bir kullanıcı "Yanıyoruz. Bu ne sıcak!" gibi bir tweet attığında bunu bilgisayar gerçek bir felaket gibi algılayabilir. Burada makine öğrenmesi yöntemleri kullanılarak bu tweetin sahte bir felaket tweeti olduğu anlaşılmaya çalışılacaktır. Burada açık kaynaklı bir internet sitesi olan Kaggle'den alınan veri kullanılarak öncelikle veri temizleme, veri analizi, veri görselleştirmesi gibi ön işlemlerden geçirilecek ve daha sonrasında da makine öğrenmesi yöntemlerinden yararlanılarak model oluşturulacaktır. Daha sonra oluşturulan modelin değerlendirilmesi yapılacaktır. Modeller karşılaştırıldıktan sonra da en iyi model seçilecektir. Verinin hazırlaması, temizlenmesi, görselleştirilmesi ve modellerin kurulması sürecinde doğal dil

işleme yöntemleri kullanılacaktır. Buradaki temel amaç gerçekten bir felaket yaşandığında ve bununla ilgili bir tweet atıldığında bu tez içerisinde kurulan bir modelin doğru tahminin yapması ile acil yardım kurumlarının daha hızlı aksiyon almasını sağlamaktır.

Anahtar Kelimeler: Makine Öğrenmesi, Twitter, metin verisi, doğal dil işleme

ABSTRACT

Predicting Fake Disaster Tweets with Machine Learning Methods

Fatma KURŞUN

Department of Statistics

Master of Science Thesis

Supervisor: Prof. Dr. Fatma NOYAN TEKELİ

In this research, It will be investigated whether the disaster tweets on Twitter are real or fake. Today, social media channels, especially Twitter, have become a very important communication. In the face of any event, people can express their opinions and follow the news here. But it is not always the right source. Within the scope of this research, it will be estimated whether a tweet by a user is real or not. To give an example; One user said, “We're on fire. How hot is that!” When you tweet something like this, the computer may perceive it as a real disaster. Here, it will be tried to understand that this tweet is a fake disaster tweet by using machine learning methods. Here, using the data taken from an open-source website which is Kaggle, firstly data cleaning, data analysis, and data visualization will be pre-processed, and then a model will be created using machine learning methods. Then the created model will be evaluated. After the models are compared, the best model will be selected. The main purpose here is to ensure that the emergency aid agencies take faster action, with a model established in this thesis making the correct prediction when a disaster really occurs and a tweet

is made about it. Natural language processing will be used in data preprocessing, visualizing, and building machine learning models.

Keywords: Machine Learning, Twitter, text mining, natural language processing

1
GİRİŞ

1.1 Literatür Özeti

Twitter, kullanıcıların "tweet" olarak bilinen mesajları yayımladığı ve birbirleriyle etkileşimde bulunduğu bir sosyal ağ hizmetidir [1]. Günümüzde internetin kullanımı oldukça kolay ve yaygındır. Dünya nüfusunun 56%'sı internet ve 45%'i de sosyal medya kullanmaktadır [2]. Sosyal medya üzerinden neredeyse tüm bilgilere kolaylıkla ulaşılırken bilgi kirliliği de çok artmıştır. Bu sosyal medya mecralarından biri de Twitter'dır. Twitter, kullanıcılar tarafından sınırlı karakterle istedikleri yazıyı fotoğrafı veya bağlantıyı paylaşabildikleri bir sosyal medya aracıdır. Twitter, haber ajansları tarafından birçok habere ulaştıklarının yanı sıra bilgi kirliliğinin de fazla olduğu bir mecradır. Bu araştırma kapsamında twitter'da paylaşılan bazı felaket tweetlerinin sahte ya da gerçek olup olmadığının tahmini makina öğrenmesi yöntemleri kullanılarak yapılacaktır [2, 3]. Bu tez kapsamında twiterdan gelen veri ile makine öğrenmesi yöntemleri kullanılarak sınıflandırma modeli kurulacaktır. Literatürde NLP yöntemleri kullanılarak twitter verisi ile duygu analizi yapılmıştır. Makine öğrenimi, bilgisayarların, açıkça programlanmadan görevleri nasıl yerine getirebileceklerini keşfetmelerini içerir. Belirli görevleri yerine getirmeleri için sağlanan verilerden öğrenen bilgisayarları kapsar. Bilgisayarlara atanan basit görevler için, makineye eldeki sorunu çözmek için gereken tüm adımları nasıl uygulayacağını bildiren algoritmalar programlamak mümkündür; bilgisayar tarafında öğrenmeye gerek yoktur. Daha gelişmiş görevlerde insan için gerekli algoritmaları elle oluşturmak zor olabilir. Uygulamada, insan programcıların gerekli her adımı belirlemesinden ziyade, makinenin kendi algoritmasını geliştirmesine yardımcı olmak daha etkili olabilir.

Makine öğrenimi disiplini, bilgisayarlara tam olarak tatmin edici bir algoritmanın bulunmadığı görevleri gerçekleştirmeyi öğretmek için çeşitli yaklaşımlar kullanır.

Çok sayıda olası yanıtın olduğu durumlarda, doğru yanıtlardan bazılarını geçerli olarak etiketlemek bir yaklaşımdır. Bu, daha sonra bilgisayarın doğru yanıtları bulmak için kullandığı algoritmayı/algoritmaları geliştirmede eğitim verisi olarak kullanılabilir [4].

Rubin ve arkadaşları haberlerdeki parodi ve komediyi tespit etmek için de bir model oluşturmuş ve incelemişlerdir. 360 alaycı haberi dört kategoride değerlendirmişlerdir: yurttaşlık bilgisi, bilim, iş dünyası ve "hassas haberler". Alaycı haberlere ilişkin analizlerine dayalı olarak 5 özelliğe dayalı olan bir destek vektör makineleri modeli inermişlerdir: Saçmalık, Mizah, Dilbilgisi, Olumsuz Etki ve Noktalama. Saçmalık, dilbilgisi ve noktalama işaretleri özelliklerini kullanarak en yüksek doğruluk düzeyi olan 90%, elde etmişlerdir [5].

Ayrıca Horne ve Adalı sahte ve gerçek içeriği ayırt etmenin ne kadar kolay olduğunu gçstermişlerdir. Sahte haber başlıkları, onların görüşüne göre, daha az durdurma kelime içerir, ancak daha fazla eylem kelimesi içerir. Onlar çok sayıda bileşeni aşağıdaki gibi üç kategoriye ayırmışlardır:

• Karmaşık vurgular, metnin karmaşıklığına ve tutarlılığına dikkat çeker.

• Duygusal ve basit kelimelerin kalitesi: örneğin beyinde kullanılan duygu sözcükleri ve rahat sözcüklerin miktarı gizlenen zihinsel etkileşimi ve bireysel kaygıları tanımlamak ve değerlendirmek için araştırma kompozisyonların içinde kullanılır.

• Eylem sözcüklerinin sayısı ve nesnelerin sayısı gibi kapsamlı özellikler, yazarların stili ve metnin dilsel yapısı [6].

Denetimli sınıflandırma yöntemleri açısından, Cobo ve arkadaşları sınıflandırma üzerinde çalışmıştır. Şili'deki bir depremle alakalı ve alakasız kelimeler üzerindeki çalışmalarında hesap takipçi ve profiline baktıkları kullanıcı bazlı ve içerik bazlı kelimelere baktılar ve metin madenciliğinin yapmışlardır. Modelleme için de Lojistik Regresyon Rastgele Orman, Destek Vektör Makinesi, Naive Bayes gibi çoklu sınıflandırıcılar kullanmışlar ve baktıkları modelleri değerlendirmek için kesinlik, hatırlama, F-skoru, eğri altındaki alan olan AUC'u kullanmışlardır [7].

1.2 Tezin Amacı

İnsanlar bir yazı okuduklarında bunun gerçek ya da sahte olduğunu çoğunlukla algılayabilmektedirler. Bazen kullanıcıların mecazi anlamda yazdıkları tweetlerin de sahte felaket tweeti olduğunu bilmektedirler. Ancak, bir kişinin sözlerinin gerçekten bir felaketi ilan edip etmediği her zaman net değildir. Bu araştırmanın amacı kullanıcıların attıkları sahte felaket tweetlerini önceden tahmin ederek bilgi kirliliğini ortadan kaldırmaktır [8]. Böylelikle herhangi bir felakette yetkililer duruma müdahale edebileceklerdir. Bir örnek vermek gerekirse, ülkemizde son zamanlarda olan deprem gibi bazı felaketlerde twitter üzerinden yapılan paylaşımlarla birçok insanın hayatı kurtarılmıştır. Bu sebeple bu araştırmanın doğru modellenmesi büyük önem taşımaktadır.

1.3 Hipotez

Bu tez çalışmasının hipotezi makine öğrenmesi modelleri kullanılarak kullanıcılar tarafından atılan tweetlerin sahte veya gerçek olarak tahmin edilebileceğini savunmaktadır. Bu araştırmada doğal dil işleme araçlarından yararlanılarak önce tweetdeki metinlerin ön işlemesi- temizlemesi yapılmış sonrasında da model kurularak sahte felaket tweetleri önceden tahmin edilmiştir.

Kullanılan veriye bakıldığında çıktı 1 (gerçek felaket tweeti) veya 0 (sahte felaket tweeti)'dır. Bu sebeple kullanılacak olan makine öğrenmesi yöntemi sınıflandırma yöntemidir. Gözetimli öğrenmenin bir dalı olan sınıflandırma yönteminin algoritmaları kullanılarak modellenmiş ve değerlendirmesi yapılmıştır.

2
MAKİNE ÖĞRENMESİ

2.1 Makine Öğrenmesi Nedir?

Son yıllarda makine öğrenmesi teknoloji alanında en önemli yapay zeka dallarından biri haline gelmiştir. Her zaman yeni ve daha güçlü araçlar ve sonuçlarla uygulamalarının her iş sektöründe her geçen gün daha da yaygınlaşması şaşırtıcı değildir. Her ay yayınlanan yüzlerce makaleyle birlikte açık kaynak, üretime hazır çerçeveler, BT tarihindeki en yaygın demokratikleşme süreçlerinden birine katkıda bulunmaktadır. Peki makine öğrenmesi neden bu kadar önemlidir [9]? Makine öğrenmesinin amacı, eldeki örnek değerlerle birlikte veriye birkaç örnek vererek (bir görevin nasıl yapılacağı veya yapılmayacağını) görevleri yerine getirmeyi öğretmektir [9]. Peki öğrenme süreci nasıldır?

Makine öğrenmesinde öğrenme süreci vardır. Temel öğrenme süreci inan veya makine ayırt etmeksizin çalışmaktadır. Aşağıdaki gibi üç bileşene ayrılabilir:

• Veri girişi: Daha fazla akıl yürüterek olgusal bir temel sağlamak için gözlem, bellek depolama ve geri çağırmayı kullanır.

• Soyutlama: Verilerin daha geniş temsillere dönüştürülmesini içerir.

• Genelleme: Olay için bir temel oluşturarak soyutlanmış veriyi kullanır [10].

Şekil 2.1 Öğrenme Süreci

2.2 Gözetimli Öğrenme

Gözetimli bir senaryo, ana görevi aracıya hatasının kesin bir ölçümünü (çıktı değerleriyle doğrudan karşılaştırılabilir) sağlamak olan bir öğretmen veya denetçi kavramı ile karakterize edilir. Gözetimli öğrenmedi bir girdi ve bir çıktı bulunmaktadır. Eğitilecek olan model bu girdi ve çıktı arasındaki ilişkiyi öğrenip buna göre eğitilmektedir. Buradaki amaç global kayıp fonksiyonun büyüklüğünü azaltmak için parametreleri düzeltmektir. Böylelikle tahmin edilen değer ile beklenen değer arasındaki fark azalarak sıfıra yaklaşacaktır. Amaçlardan biri ise daha önce hiç görmediği örneklerle de çalışması gereken bir sistemi eğitmektir [9]. Gözetimli öğrenme regresyon ve sınıflandırma olarak ayrılmaktadır. Regresyon algoritmasında çıktı sürekli bir değişkenken iken sınıflama algoritmasında çıktı kategorik değişkenlerdir.

Yaygın gözetimli öğrenme uygulamaları;

Regresyon veya kategorik sınıflandırmaya, tahmine dayalı analiz

Spam algılama

Doğal Dil İşleme

Duygu analizi

Otomatik görüntü sınıflandırması [3]

Bu tezde gözetimli öğrenme problemi olan sınıflandırma algoritmaları kullanılacaktır.

2.3 Gözetimsiz Öğrenme

Bu yaklaşım, herhangi bir denetimin olmamasına ve dolayısıyla mutlak hata ölçütlerine dayanmaktadır; bir dizi öğenin benzerliklerine (veya uzaklık ölçülerine) göre nasıl gruplanabileceğini (kümelenebileceğini) öğrenmek gerektiğinde yararlıdır [6].Burada çıktı değişkeni yoktur. Gözetimsiz öğrenme modelleri üç ana dala ayrılır: kümeleme, ilişkilendirme ve boyut azaltma. Bazı kullanım alanları şöyledir;

Haber Bölümleri: Google Haberlerden bahsedilecek olursa; aynı haber değerini taşıyan haber içeriklerini kategorilere ayırmak için denetimsiz öğrenme araçlarından biri olan kümeleme yöntemini kullanmaktadır. Örneğin, Amerika'daki başkanlık seçiminin sonuçları "ABD" haberleri etiketi altında kümelenebilir.

Bilgisayarla görme: Nesne tanıma gibi görsel veriye dayalı tahminlerde kullanılır. Yeni nesil akıllı arabalarda olan şerit takibi veya akıllı telefonlardaki yüz tanıma sistemi bunlara örnektir.

Anormallik tespiti: Denetimsiz öğrenme modelleri, büyük miktarda veri ile çalışıp o verideki anormal veri noktalarını yakalayabilmektedir. Örneğin bir hava durumu veri setinde anormal bir sıcaklık denetimsiz makine öğrenmesi yöntemleri ile tespit edilebilmektedir.

Müşteri kişilikleri: Satın alma alışkanlıklarını anlayabilmek adına müşteri kişilikleri tanımlanmalıdır. Böylelikle müşterilerin ortak özellikleri belirlenebilmektedir. Denetimsiz öğrenme ile benzer alışkanlıklara sahip müşteriler sınıflandırılarak işletmelere büyük kolaylık sağlanır.

Öneri Motorları: Satın alma davranışlarından yola çıkılarak o işletmeye yeni katılan bir müşteriye denetimsiz öğrenme yöntemleri ile satın alınabilecek ürünler önerilebilmektedir. Böylelikle müşterinin işletme içinde kalması olasılığı artırılmaktadır. Netflix ve YouTube gibi büyük teknoloji şirketleri öneri motorları modellerini kullanmaktadır [11].

2.4 Doğal Dil İşleme

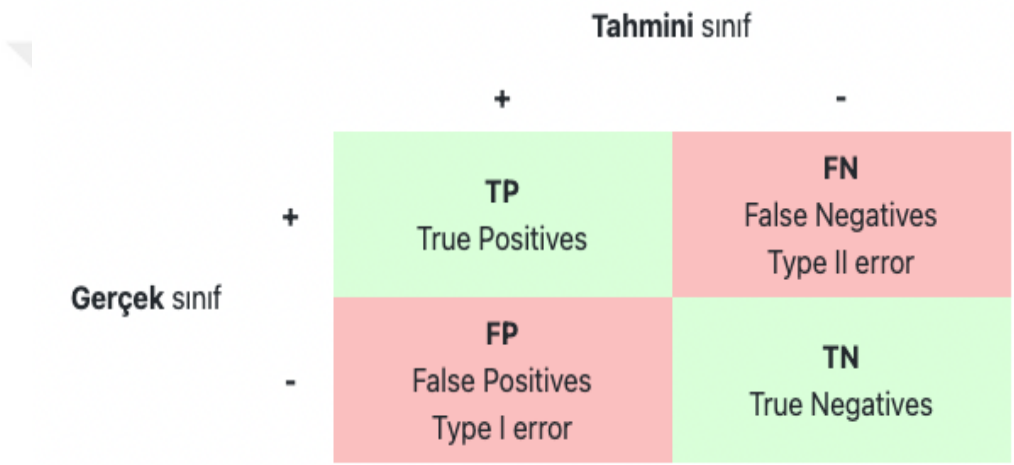
Doğal dil işleme, iç yapıları ve kelimelerin dağılımını göz önünde bulundurarak metin belgeleriyle çalışmayı sağlayan bir dizi makine öğrenme tekniğidir. Bu bölümde, metinleri toplamak, onları parçalara bölmek ve sayısal vektörlere dönüştürülmektedir. Özellikle kelimeleri simgeleştirme – her kelimeyi ayırmak, kelimeleri filtrelemek, kelimelerin köklerini bulmak için özel uygulamalar yaparak ve son olarak da bir kelime dağarcığı oluşturmak için farklı yöntemler kullanmaktadır. Bu kelime dağarcığını kullanarak kümeleme veya sınıflama amaçlarıyla kullanılabilecek özellik vektörü oluşturmak mümkündür. Doğal dil

işleme yöntemleri kullanılarak bir metin verisi makine öğrenmesi yöntemleri için hazır hale getirilmektedir [9].

2.5 Sınıflandırma Modellerinin Değerlendirilmesi

Bir sınıflandırma problemi birçok farklı metrik ile değerlendirilebilmektedir. İkili bir sınıflandırma durumunda, modelin performansını değerlendirmek için gerekli olan ana metrikler aşağıda verilmiştir [12].

Karışıklık matrisi, tahmin sonuçlarını doğru ve yanlış tahminleri göstererek ortaya koymaktadır. Aşağıdaki şekilde tanımlanmıştır [12]:



Şekil 2.2 Karışıklık Matrisi

Doğru pozitif (TP): 1 olarak sınıflandırılan ve gerçekten de 1 olan değerlerin sayısıdır.

Yanlış pozitif (FP): 1 olarak sınıflandırılan fakat gerçekte 0 olan değerlerin sayısıdır.

Doğru negatif (TN): 0 olarak sınıflandırılan ve gerçekten de 0 olan değerlerin sayısıdır.

Yanlış negatif (FN): 0 olarak sınıflandırılan fakat gerçekte 1 olan değerlerin sayısıdır [3]

Ana metrikler- Sınıflandırma modellerinin performansını değerlendirmek için aşağıda verilen metrikler yaygın olarak kullanılmaktadır [12]

Tablo 2.1 Metrikler

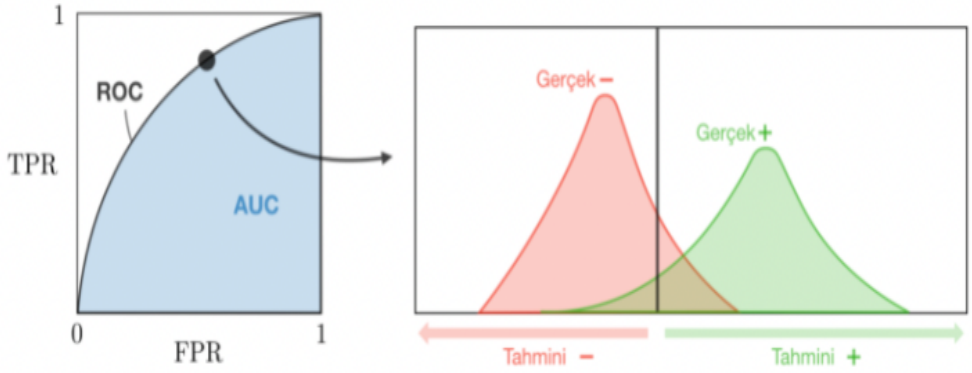
Metrik	**Formül**	**Açıklama**
Doğruluk	$\frac{TP + TN}{TP + TN + FP + FN}$	Modelin genel performansı
Kesinlik	$\frac{TP}{TP + FP}$	Doğru tahminlerin ne kadar kesin olduğu
Geri Çağırma	$\frac{TP}{TP + FN}$	Gerçek pozitif örneklerin oranı
Özgünlük	$\frac{TN}{TN + FP}$	Gerçek negatif örneklerin oranı
F1 Skor	$\frac{2TP}{2TP + FP + FN}$	Dengesiz sınıflar için yararlı hibrit metrik

İşlem Karakteristik Eğrisi (ROC)- İşlem Karakteristik Eğrisi, eşik değeri değiştirilerek Doğru Pozitif Oranı-Yanlış Pozitif Oranı grafiğidir. Bu metrikler aşağıdaki tabloda özetlenmiştir [12]

Tablo 2.2 ROC

Metrik	**Formül**	**Eşdeğer**
Gerçek Pozitif Oranı TPR	$\frac{TP}{TP + FN}$	Geri çağırma
Yanlış Pozitif Oranı FPR	$\frac{FP}{TN + FP}$	1-specificity

Eğri Altında Kalan Alan (AUC)- Aynı zamanda AUC veya AUROC olarak belirtilen işlem karakteristik eğrisi altındaki alan, aşağıdaki şekilde gösterildiği gibi İşlem Karakteristik Eğrisi (ROC)'nin altındaki alandır [12]:



Şekil 2.3 AUC

ROC bize modelin gerçek pozitif oranı ile yanlış pozitif oranı cinsinden ne kadar iyi ayrım yapabildiğini açıklar. AUC 0 ile 1 arasındadır ve ROC eğrisinin altında kalan alanı vermektedir. AUC eğer 0 olursa bütün tahminler yanlış demektir. Gerçek pozitif oranı kısaca; durum gerçekte pozitifken modelin bu pozitiflerin kaçını pozitif tahmin ettiğini gösterir, yanlış pozitif oranı ise durum gerçekte negatifken model bunların kaçını pozitif tahmin ettiğini göstermektedir. Buna yanlış alarm da denmektedir [12].

Bu tez genelinde kullanılan makine öğrenmesi yöntemleri bu metrikler kullanılarak değerlendirilmiştir.

3
MAKİNE ÖĞRENMESİ YÖNTEMLERİNİN UYGULANMASI

3.1 Problemin Anlaşılması

Teknolojinin gelişmesi ve akıllı telefonların çok yaygınlaşması sebebiyle Twitter, acil durumlarda önemli bir iletişim kanalı haline gelmiştir. Twitterda atılan tweetler bazen insanların gözlemledikleri bir acil durumu gerçek zamanlı olarak duyurmalarını sağlamaktadır. Bu nedenle, günümüzde daha fazla haber ajansları veya yardım kuruluşları Twitter'ı programlı olarak izlemekle ilgilenmektedir [8]. Bu tezde, Twitterdan toplanan veri ile gerçek ya da sahte felaket olaylarının tahmini yapılacaktır. Twitterdan toplanan veri ile model kurma aşamalarına başlanmıştır. Burada toplanan veri metin tipindedir.

3.2 Verinin Toplanması

Veri açık kaynak olarak https://www.kaggle.com/c/nlp-getting-started/data adresinden alınmıştır. Herkesin erişebileceği bir yerdir. Kaggle bu veri seti ile bir yarışma gerçekleştirmiştir. Burada eğitim ve test verisi ayrılmış haldedir ve 7613 adet eğitim, 3263 adet test verisi bulunmaktadır. Kaggle'ın paylaştığı eğitim verisinde "target" değişkeni (tahmin edilen değişken) mevcuttur fakat test verisinde bu bilgi yer almamaktadır. Bu tweetler 25'ten fazla ülkelerden toplanmıştır.

Burada paylaşılan eğitim verisi kullanılarak modellenecek ve daha sonrasında modelin hiç görmediği test verisi kullanılarak tahmin yapılacak ve kaggle platformuna yüklenerek modelin gerçek başarı performansı görülecektir [8].

3.3 Veri Önişleme

Bu bölüm veri temizleme, görselleştirme ve veri analizi adımlarını içermektedir. Python programlama dili kullanılarak veri önişleme adımları yapılmıştır.

Tablo 3.1 Verinin ilk 5 satırı

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

Tablo3.1'de eldeki eğitim verisinin bir örneği vardır. Burada id her bir tweeti, keyword o twitteki anahtar kelimeleri, location o tweetin nereden atıldığını, text tweet'in içeriğini ve target da o tweetin gerçek bir felaket olup olmadığını ifade etmektedir. Target değişkeni kurulacak olan modeldeki hedef (tahmin edilecek) değişkendir. 1 gerçek 0 sahte tweetleri temsil etmektedir. Burada yapılacak olan ilk işlem veride eksik verilerin kontrolü ve eğer gerekli görülürse burada eksik verileri doldurma veya veriden atma işlemi uygulanacaktır.

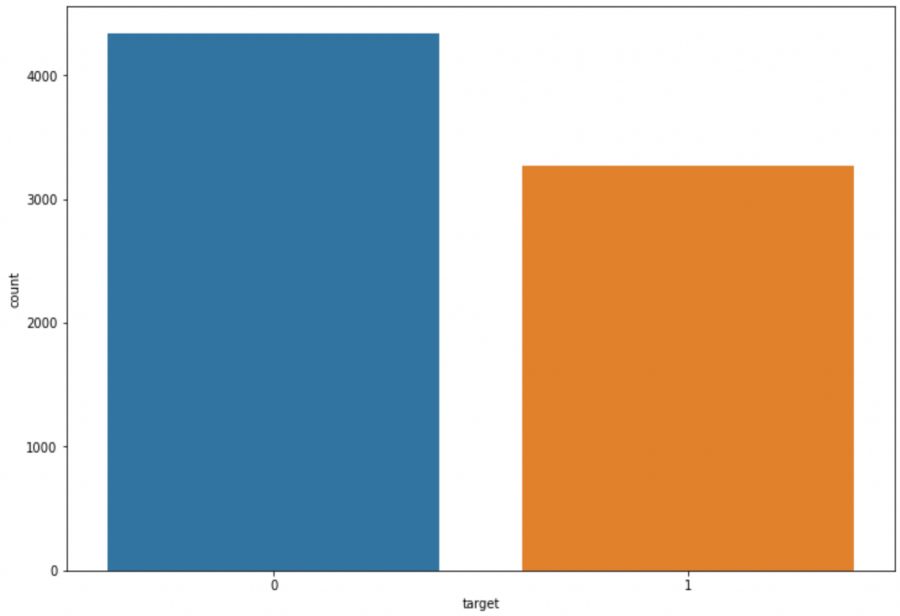
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        7613 non-null   int64
 1   keyword   7552 non-null   object
 2   location  5080 non-null   object
 3   text      7613 non-null   object
 4   target    7613 non-null   int64
dtypes: int64(2), object(3)
memory usage: 297.5+ KB
```

Şekil 3.1 Veri bilgileri

Şekil 3.1'de görüldüğü üzere keyword ve location değişkenlerinde eksik değer olduğu görülmektedir. Bu eksik değerleri 'None' değeri ile doldurma işlemi uygulanmıştır. Çünkü burada anahtar kelimelerin ve lokasyonun nerede olduğunun bilinmesi önemli bir bilgi olabilir.

Bu bölümden itibaren target değişkeni ile diğer ilgili değişkenler arasındaki bazı istatistiksel çıkarımlara bakılmış ve verinin analizi ve görselleştirmesi yapılmıştır.

```
sns.countplot(train['target'])
plt.show()
print(train['target'].value_counts())
```

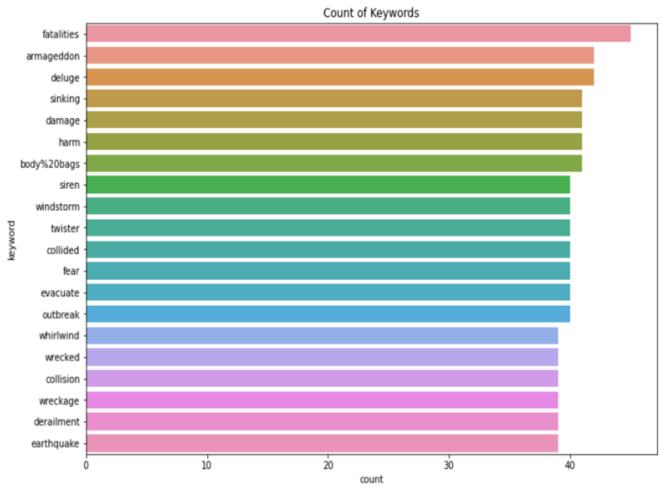



```
0    4342
1    3271
Name: target, dtype: int64
```

Şekil 3.2 Target değişkeninin sayısı

Şekil3.2'de görüldüğü üzere eğitim verisinde 4342 adet 0(sahte) 3271 adet 1(gerçek) değer bulunmaktadır. Burada hedef değişkeninde çok fark bulunmadığı için veri dengeli denilebilmektedir.

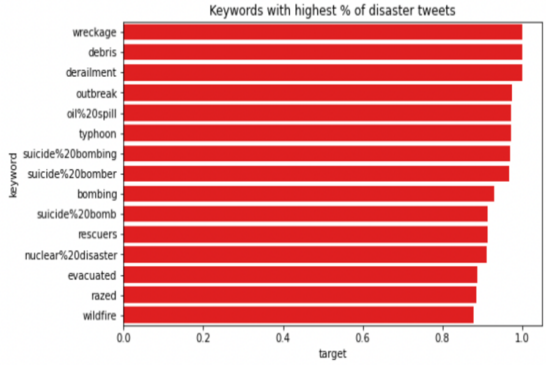
```
sns.countplot(y = train.keyword,order = train['keyword'].value_counts().sort_values(ascending=False).iloc[0:20].index)
plt.title("Count of Keywords")
plt.show()
```



Şekil 3.3 Keyword değişkenin en çok kullanılan kelimeleri

Şekil 3.3'e bakıldığında keyword değişkeninde en çok geçen kelimelerin sırası görülmektedir. Burada hangi kelimenin sahte veya gerçek tweetlerde geçtiği görülebilir. Bundan sonraki adımda gerçek ve sahte olan tweetleri ayırıp bunlarda hangi kelimelerin ne sıklıkla geçtiğine bakılmıştır.

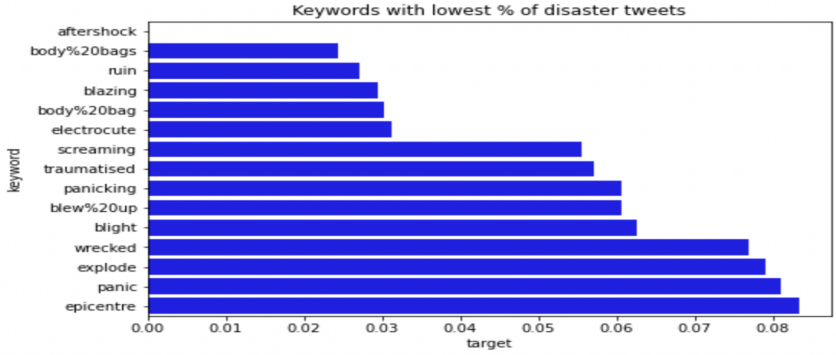
```
# Count of keywords for real disaster;
disastered_tweet = train.groupby('keyword')['target'].mean().sort_values(ascending=False).head(15)
non_disasterd  = train.groupby('keyword')['target'].mean().sort_values().head(15)

plt.figure(figsize=(8,5))
sns.barplot(disastered_tweet, disastered_tweet.index, color='red')
plt.title('Keywords with highest % of disaster tweets')
plt.show()
```



Şekil 3.4 Gerçek felaket içeren tweetlerin keyword sıralaması

Disaster ve non-disaster diye iki farklı değişken oluşturulmuş ve bu değişkenler içindeki keywordlerin sıralamasına bakılmıştır. Şekil 3.4'te gerçek felaket olan değerlerin sıralamasına bakılmaktadır ve şekilde görüldüğü üzere wreckage(enkaz) kelimesi en çok geçen kelimedir. Bu kelime gerçek felaket için anlam taşımaktadır. Sahte felaket tweetlerindeki kelimelere bakıldığında ise;

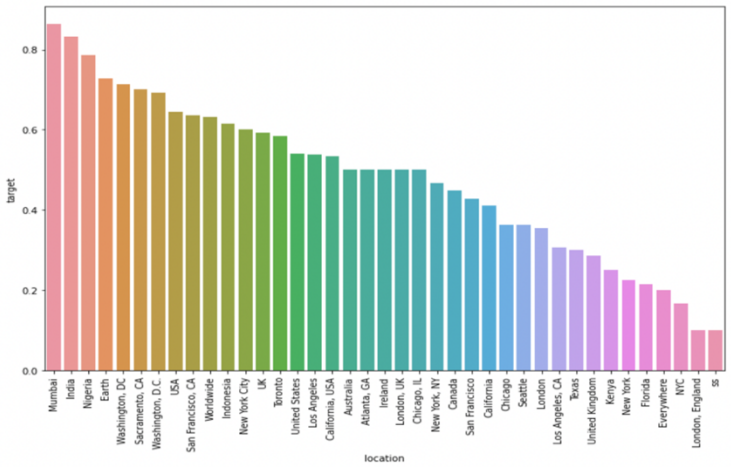
```
#Count of eywords for Non-Disasters
plt.figure(figsize=(8,5))
sns.barplot(non_disasterd, non_disasterd.index, color='blue')
plt.title('Keywords with lowest % of disaster tweets')
plt.show()
```



Şekil 3.5 Sahte tweetlerdeki keyword sıralaması

En çok kullanılan kelimenin “aftershock” olduğu gözlemlenmektedir. Makine öğrenmesi modeli kurulduğunda bu değişkenlerin önemi ortaya çıkacaktır.

```
raw_loc = train.location.value_counts()
top_loc_disaster = list(raw_loc[raw_loc>=10].index)
top_only_disaster = train[train.location.isin(top_loc_disaster)]

top_location = top_only_disaster.groupby('location')['target'].mean().sort_values(ascending=False)
sns.barplot(x=top_location.index, y=top_location)
plt.xticks(rotation=90)
plt.show()
```

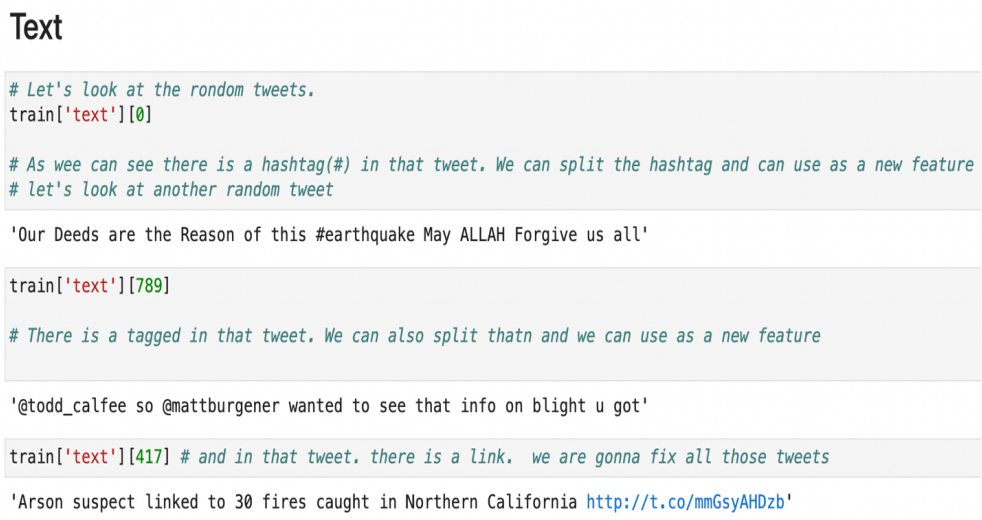


Şekil 3.6 Lokasyona göre en çok atılan sahte tweetlerin sayısı

Lokasyon değişkeninin analizi yapıldığında en çok sahte felaket tweetlerinin Mumbaiden yapıldığı görülmektedir. Şekil 3.6’da bir başka göze çarpan durum ise bu değişkenin kirliliğidir. Değişken içerisinde hem New York City hem NYC, hem de New York değerleri vardır. Bu şehirler aynı olduğu halde farklı değerlermiş gibi görülmektedir burada bir veri temizleme işlemi ile bir düzeltme yapılacak ve tekrar kontrol edilmiştir. Bu işlemden önce keyword ve location değişkenlerindeki eksik değerler “None” değişkeni ile doldurulmuştur.

```python
def clean_location(x):
    if x == 'None':
        return 'None'
    elif x == 'Earth' or x =='Worldwide' or x == 'Everywhere':
        return 'World'
    elif 'New York' in x or 'NYC' in x:
        return 'New York'
    elif 'London' in x:
        return 'London'
    elif 'Mumbai' in x:
        return 'Mumbai'
    elif 'Washington' in x and 'D' in x and 'C' in x:
        return 'Washington DC'
    elif 'San Francisco' in x:
        return 'San Francisco'
    elif 'Los Angeles' in x:
        return 'Los Angeles'
    elif 'Seattle' in x:
        return 'Seattle'
    elif 'Chicago' in x:
        return 'Chicago'
    elif 'Toronto' in x:
        return 'Toronto'
    elif 'Sacramento' in x:
        return 'Sacramento'
    elif 'Atlanta' in x:
        return 'Atlanta'
    elif 'California' in x:
        return 'California'
    elif 'Florida' in x:
        return 'Florida'
    elif 'Texas' in x:
        return 'Texas'
    elif 'United States' in x or 'USA' in x:
        return 'USA'
    elif 'United Kingdom' in x or 'UK' in x or 'Britain' in x:
        return 'UK'
    elif 'Canada' in x:
        return 'Canada'
    elif 'India' in x:
        return 'India'
    elif 'Kenya' in x:
        return 'Kenya'
    elif 'Nigeria' in x:
        return 'Nigeria'
    elif 'Australia' in x:
        return 'Australia'
    elif 'Indonesia' in x:
        return 'Indonesia'
    elif x in top_location:
        return x
    else:
        return 'Others'

train['location'] = train['location'].apply(lambda x: clean_location(str(x)))
test['location'] = test['location'].apply(lambda x: clean_location(str(x)))
```

Şekil 3.7 Lokasyon değişkeni için veri temizleme fonksiyonu



Şekil 3.8 Text değişkeninin içeriği

Şekil3.8'de görüldüğü gibi text değişkeni olay etiketi, kişi etiketi ve linkler içermektedir. Bazı etiketler önem taşımaktadır. İlk örnekte görüldüğü üzere deprem kelimesi etiketlenmiştir ve bu tweet eğer gerçekse bu etiket makine öğrenmesi modeli için kolaylık sağlamıştır. Olay etiketi, kişi etiketi ve linkler yeni birer değişken olarak veriye katılacak ve girdi olarak makine öğrenmesi modellerine eklenecektir.

```
import re

# We are going to split the hashtag,link and tagged
def created_feature(train):
    train['hashtags'] = train['text'].apply(lambda x: " ".join([match.group(0)[1:] for match in re.finditer(r"#\w+", x)]) or 'no_hashtag')
    train['tagged'] = train['text'].apply(lambda x: " ".join([match.group(0)[1:] for match in re.finditer(r"@\w+", x)]) or 'no_tagged')
    train['link'] = train['text'].apply(lambda x:" ".join([match.group(0)[:] for match in re.finditer(r"https?://\S+", x)]) or 'no_link')
    return train
```

```
train = created_feature(train)
test = created_feature(test)
```

Şekil 3.9 Yeni üç değişkenin oluşturulması

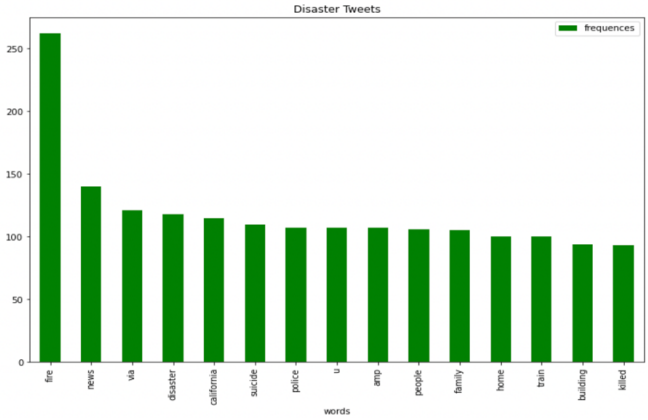
Veriye üç yeni değişken eklenmiştir ve ardından tekrar değişkenler içerisinde bazı veri temizleme işlemleri uygulanarak makine öğrenmesi modelleri için uygun hale getirilmiştir.

```
## Text Mining
import nltk
nltk.download("stopwords")
!pip install textblob
nltk.download("wordnet")

#Upper lower convert
train['text'] = train['text'].apply(lambda x:" ".join(x.lower() for x in x.split()))
test['text'] = test['text'].apply(lambda x:" ".join(x.lower() for x in x.split()))

# punctuation marks
train['text'] =train['text'].str.replace('[^\w\s]','')
test['text'] =test['text'].str.replace('[^\w\s]','')

# numbers
train['text'] = train['text'].str.replace('[\d]','')
test['text'] = test['text'].str.replace('[\d]','')

from nltk.corpus import stopwords
sw = stopwords.words('english')
train['text'] =train['text'].apply(lambda x: " ".join(x for x in x.split() if x not in sw))
test['text'] =test['text'].apply(lambda x: " ".join(x for x in x.split() if x not in sw))

#lemmi
from textblob import Word
train['text'] = train['text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
test['text'] = test['text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))


train['text'] = train['text'].str.replace('rt','')
test['text'] = test['text'].str.replace('rt','')
```

Şekil 3.10 Metin madenciliği

NLTK pythonda; metin madenciliği ve doğal dil işleme konularında değişken değerlerini düzenlemek için kullanılan bir kütüphanedir. Şekil10'da yapılan işlemlerde text değişkenindeki hatalı ve gereksiz değerleri kaldırmak ya da başka değerlerle değiştirmek için kullanılmıştır. Bu temizleme işlemlerinden sonra text değişkenindeki frekanslara -kelimelerin yoğunluğuna bakılmıştır. Buradan gerçek felaketleri içeren tweetlerde en çok hangi kelimelerin kullanıldığı öğrenilmiş ve bu değişkenler modele girdi olmuştur.

```
#Most used words dor disasters
freq_df = train[train['target']==1]['text'].apply(lambda x:pd.value_counts(x.split(" "))).sum(axis = 0).reset_index()
freq_df.columns = ['words', 'frequences']
freq_df.sort_values('frequences',ascending=False)

top_freq_disaster = freq_df.sort_values('frequences',ascending=False)[0:15]
top_freq_disaster.set_index('words',inplace=True)
top_freq_disaster.plot.bar(color ='g')
plt.title("Disaster Tweets")
plt.show()  #Fire and news are most used words in the disasters tweets.
```



Şekil 3.11 Felaket tweetlerinde en çok geçen kelimelerin sırası

Şekil3.11'de görüldüğü üzere gerçek felaket tweetlerinde en çok geçen kelime "fire" (yangın) kelimesidir. Bu da model için anlam ifade etmektedir.

3.4 Değişken Mühendisliği

Bu bölümde count-vector ve tf-idf açıklanacak ve yeni değişkenler eklenerek veri son haline getirilecektir.

Sayma Vektörü (count vector): Dokümanlardaki geçme sıklığına göre oluşturulan vektörlerdir [8].

Count Vectors

```
import category_encoders as ce

# Target encoding
features = ['keyword', 'location']
encoder = ce.TargetEncoder(cols=features)
encoder.fit(train[features],train['target'])

train = train.join(encoder.transform(train[features]).add_suffix('_target'))
test = test.join(encoder.transform(test[features]).add_suffix('_target'))
```

```
from sklearn.feature_extraction.text import CountVectorizer

# Links
vec_links = CountVectorizer(min_df = 5, analyzer = 'word', token_pattern = r'https?://\S+') # Only include those >=5 occurrences
link_vec = vec_links.fit_transform(train['link'])
link_vec_test = vec_links.transform(test['link'])
X_train_link = pd.DataFrame(link_vec.toarray(), columns=vec_links.get_feature_names())
X_test_link = pd.DataFrame(link_vec_test.toarray(), columns=vec_links.get_feature_names())

# Tagged
vec_tag = CountVectorizer(min_df = 5)
tag_vec = vec_tag.fit_transform(train['tagged'])
tag_vec_test = vec_tag.transform(test['tagged'])
X_train_tag = pd.DataFrame(tag_vec.toarray(), columns=vec_tag.get_feature_names())
X_test_tag = pd.DataFrame(tag_vec_test.toarray(), columns=vec_tag.get_feature_names())

# Hashtags
vec_hash = CountVectorizer(min_df = 5)
hash_vec = vec_hash.fit_transform(train['hashtags'])
hash_vec_test = vec_hash.transform(test['hashtags'])
X_train_hash = pd.DataFrame(hash_vec.toarray(), columns=vec_hash.get_feature_names())
X_test_hash = pd.DataFrame(hash_vec_test.toarray(), columns=vec_hash.get_feature_names())
```

Şekil 3.12 Count-Vector

Şekil3.12'de bazı değişkenleri geçme sıklığına göre değişiklikler yapılmıştır.

TF (t) = (Bir belgedeki terimin sıklığı) / (bir belgedeki toplam terim sayısı)

IDF (t) = log_e (Toplam belge sayısı) / (içinde t terim bulunan belge sayısı) [7]

TF-IDF

```
# Tf-idf for text
from sklearn.feature_extraction.text import TfidfVectorizer

vec_text = TfidfVectorizer(min_df = 10, ngram_range = (1,2), stop_words='english')
text_vec = vec_text.fit_transform(train['text'])
text_vec_test = vec_text.transform(test['text'])
X_train_text = pd.DataFrame(text_vec.toarray(), columns=vec_text.get_feature_names())
X_test_text = pd.DataFrame(text_vec_test.toarray(), columns=vec_text.get_feature_names())
print (X_train_text.shape)
```

```
(7613, 1567)
```

```
train = train.join(X_train_link, rsuffix='_link')
train = train.join(X_train_tag, rsuffix='_tagged')
train = train.join(X_train_hash, rsuffix='_hashtag')
train = train.join(X_train_text, rsuffix='_text')

test = test.join(X_test_link, rsuffix='_link')
test = test.join(X_test_tag, rsuffix='_mention')
test = test.join(X_test_hash, rsuffix='_hashtag')
test = test.join(X_test_text, rsuffix='_text')

print (train.shape)
```

```
(7613, 1708)
```

Şekil 3.13 TF-IDF

```
train.head()
```

	id	keyword	location	text	target	hashtags	tagged	link	keyword_target	location_target	...	youtube video	yr	yr old	yyc_text	zone	û_	ûïwhen	ûïwhen saw	ûò	ûó
0	1	None	None	deed reason eahquake may allah forgive u	1	earthquake	no_tagged	no_link	0.688525	0.424398	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	4	None	None	forest fire near la ronge sask canada	1	no_hashtag	no_tagged	no_link	0.688525	0.424398	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5	None	None	resident asked shelter place notified officer ...	1	no_hashtag	no_tagged	no_link	0.688525	0.424398	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	6	None	None	people receive wildfire evacuation order calif...	1	wildfires	no_tagged	no_link	0.688525	0.424398	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	7	None	None	got sent photo ruby alaska smoke wildfire pour...	1	Alaska wildfires	no_tagged	no_link	0.688525	0.424398	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 1708 columns

Şekil 3.14 Eğitim verisinin son hali

Count-vector ve TF-IDF işlemlerinden sonra eğitim verisinin son hali Şekil 3.14'teki gibidir. Toplamda 1708 değişken sayısına ulaşılmıştır. Veri makine öğrenmesi modelleri için son haline getirilmiştir. Bu işlemden sonra eğitim verisi tekrar eğitim-test olarak bölünecektir ve hali hazırdaki test verisi de doğrulama için en son değerlendirme olarak kullanılacaktır.

Şekil 3.15 Verinin eğitim-test olarak ayrılması

Şekil 3.15'e bakıldığında veri bölünmüş ve modelleme için hazır hale getirilmiştir. 1700 adet değişken girdi olarak kullanılacak ve target değişkeni de hedef değişkeni olarak kullanılacaktır. 5329 adet eğitim, 2284 adet de test verisi vardır. Verinin 70% eğitim 30% test için ayrılmıştır.

3.5 Makine Öğrenmesi Modellerinin Kurulması

Veri ön işlemlerden geçirildikten ve bölündükten sonra modelleme için hazır hale getirilmiştir. Bu bölümde 4 farklı makine öğrenmesi modeli eğitilecek ve modeller arasında en iyi doğruluk oranını veren model kullanılacaktır. Bu araştırmadaki hedef değişkeni kategorik bir değişkendir. Bu sebeple sınıflandırma modelleri kullanılacaktır.

3.5.1.1 Lojistik Regresyon

Lojistik regresyon en çok kullanılan sınıflandırma algoritmalarından biridir. Lojistik regresyon hedef değişkeni kategorik değişken olduğu zamanlarda kullanılmaktadır. Lojistik regresyon aslında sınıflandırma problemlerinde çalışabilen bir doğrusal regresyondur. Lojistik regresyon ile lineer regresyon arasındaki fark, lojistik regresyonda çıktı 0 ile 1 arasındayken lineer regresyonda sürekli değişkendir. Ayrıca lineer regresyonun aksine lojistik regresyon girdiler ve çıktı değişkenleri arasında doğrusal bir ilişki olması koşulu yoktur. Bunun nedeni, oran oranına doğrusal olmayan bir log dönüşümü uygulanmasıdır [9].

$$Logistic\ function = \frac{1}{1+e^{-x}} \tag{3.1}$$

Lojistik fonksiyonun formülü yukarıdaki gibidir. Burada x girdi değişkenlerini ifade etmektedir.

Uygulama:

```
from sklearn import linear_model
log = linear_model.LogisticRegression()
log_model = log.fit(train_x, train_y)
```

Şekil 3.16 Lojistik regresyon model eğitimi

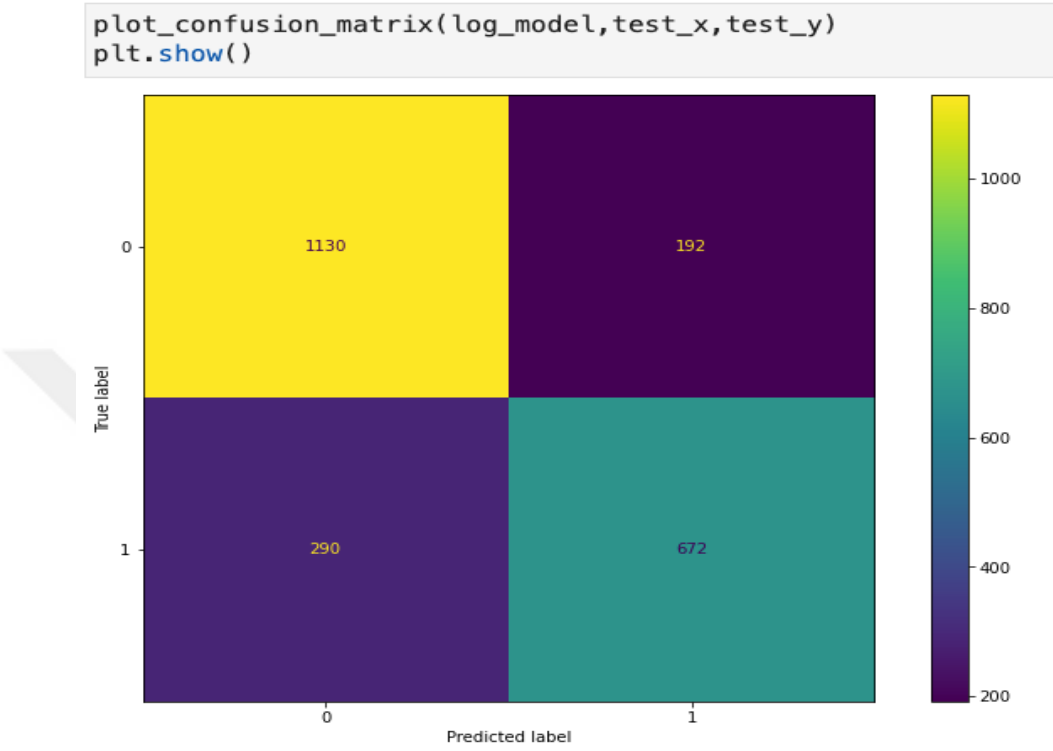
Veri eğitim ve test olarak ayrıldıktan sonra yapılması gereken tek iş modelin eğitimidir. Lojistik regresyon modeli python'da scikit-learn kütüphanesinden çekilip kullanılmaktadır. Eğitim tamamlandıktan sonra test olarak ayrılan veri tahmin için kullanılmıştır.

```
predictions = log_model.predict(test_x)
acc_score = accuracy_score(test_y,predictions)
pre_score = precision_score(test_y,predictions)
rec_score = recall_score(test_y,predictions)
print('Accuracy_score: ',acc_score)
print('Precision_score: ',pre_score)
print('Recall_score: ',rec_score)
print('--'*35)
cr = classification_report(test_y,predictions)
print(cr)
```

```
Accuracy_score:  0.7889667250437828
Precision_score:  0.7777777777777778
Recall_score:  0.6985446985446986
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.80      0.85      0.82      1322
           1       0.78      0.70      0.74       962

    accuracy                           0.79      2284
   macro avg       0.79      0.78      0.78      2284
weighted avg       0.79      0.79      0.79      2284
```

Şekil 3.17 Lojistik regresyon modeli tahminleri

Eğitimin tamamlanmasından sonra test verisi kullanılarak tahmin yapılmış ve doğruluk sonucu 0.79 çıkmıştır. Makine Öğrenmesi bölümünde açıklanan sınıflandırma raporu Şekil 3.17'de gösterilmektedir. Yaklaşık 80%'lik bir doğruluk oranı sınıflandırma algoritmaları için kabul edilebilir bir orandır.



Şekil 3.18 Lojistik regresyon modeli çarpıklık matrisi grafiği

Çarpıklık matrisi grafiği Şekil 3.18'de görülmektedir. Çarpıklık matrisi Bölüm 2.5'te anlatılmıştır. Buradan anlaşılacağı üzere sahte tweetlerin tahmini biraz daha zor denebilir.

3.5.1.2 Rasgele Orman Yöntemi

Rastgele orman, bir dizi karar ağacından oluşmaktadır. Böyle bir modelde en iyi seçimi aramak yerine, her bir ağaç için rasgele bir özellik alt kümesi kullanılır ve verileri en iyi ayıran sınır bulunmaktadır. Böylelikle rasgele ormanda birden fazla eğitilmiş karar ağaçları vardır ve her bir karar ağacı farlı bir tahmin üretmektedir. Bu sonuçları yorumlamanın iki yolu vardır; daha yaygın olan yaklaşım çoğunluk oylamasına dayanmakta ve en çok oy alan sınıf doğru kabul edilmektedir. Bununla birlikte, scikit-learn, sonuçların ortalamasını almaya dayalı bir algoritma uygular

ve bu da çok doğru tahminler vermektedir. Teorik olarak farklı olsalar bile, eğitilmiş bir rastgele ormanın olasılıksal ortalaması, tahminlerin çoğundan çok farklı olamaz (aksi takdirde, farklı kararlı noktalar olmalıdır); bu nedenle iki yöntem genellikle karşılaştırılabilir sonuçlara götürür [3].

Uygulama:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf_model = rf.fit(train_x,train_y)
```

Şekil 3.19 Rasgele Orman modeli eğitimi

Rasgele orman modeli scikit-learn kütüphanesinden çekilerek eğitilmiştir.

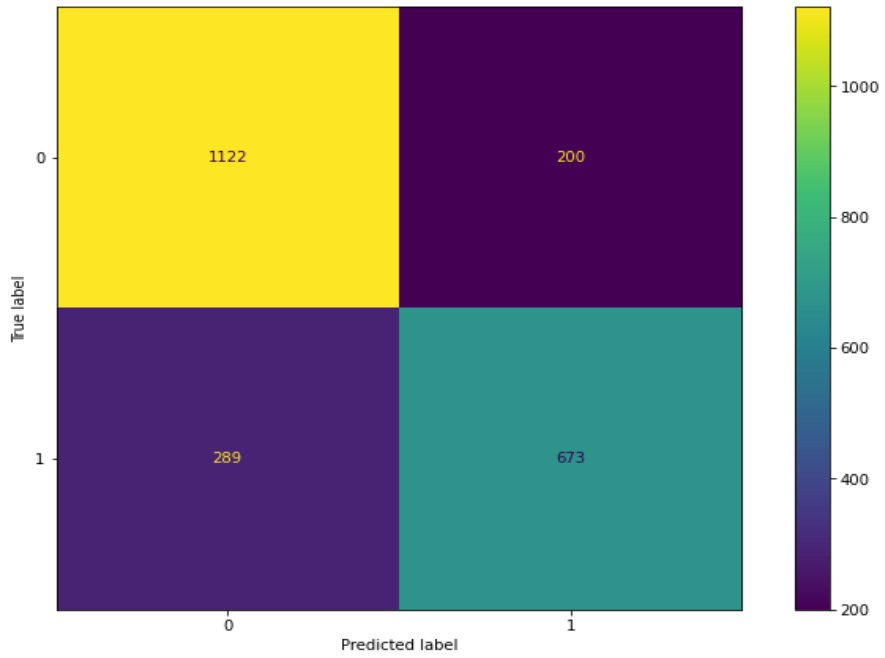
```
predictions = rf_model.predict(test_x)
acc_score = accuracy_score(test_y,predictions)
pre_score = precision_score(test_y,predictions)
rec_score = recall_score(test_y,predictions)
print('Accuracy_score: ',acc_score)
print('Precision_score: ',pre_score)
print('Recall_score: ',rec_score)
print('--'*35)
cr = classification_report(test_y,predictions)
print(cr)
```

```
Accuracy_score:  0.7859019264448336
Precision_score:  0.7709049255441008
Recall_score:  0.6995841995841996
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.80      0.85      0.82      1322
           1       0.77      0.70      0.73       962

    accuracy                           0.79      2284
   macro avg       0.78      0.77      0.78      2284
weighted avg       0.78      0.79      0.78      2284
```

Şekil 3.20 Rasgele orman modeli tahminleri

Rasgele orman modeli 78%'lik bir doğruluk oranı ile çalışmıştır.

```
plot_confusion_matrix(rf_model,test_x,test_y)
plt.show()
```



Şekil 3.21 Rasgele Orman modelinin çarpıklık matrisi grafiği

3.5.1.3 Naive Bayes Yöntemi

Bir Bayes sınıflandırıcısı, nedenlerin koşullu bağımsızlığını ima eden saf bir koşula dayandığı için böyle adlandırılır. Değişkenler arasında kuvvetli bir ilişki olduğunda bunu tahminlemesi bazen çok zor olabilmektedir. Örneğin, spam filtrelemede, 50 karakterden daha kısa bir metin görüntünün bulunma olasılığını artırabilir veya etki alanı aynı spam e-postalarını milyon kullanıcıya göndermek için zaten kara listeye alınmışsa, belirli anahtar kelimeleri bulma olasılığı yüksektir. Başka bir deyişle, olaylar bazen birbirinden bağımsız değildir [9].

Uygulama:

```
nb= naive_bayes.MultinomialNB()
nb_model = nb.fit(train_x, train_y)
```

Şekil 3.22 Naïve bayes modeli eğitimi

Naïve bayes modeli kendi kütüphanesinden çekilerek eğitilmiştir.

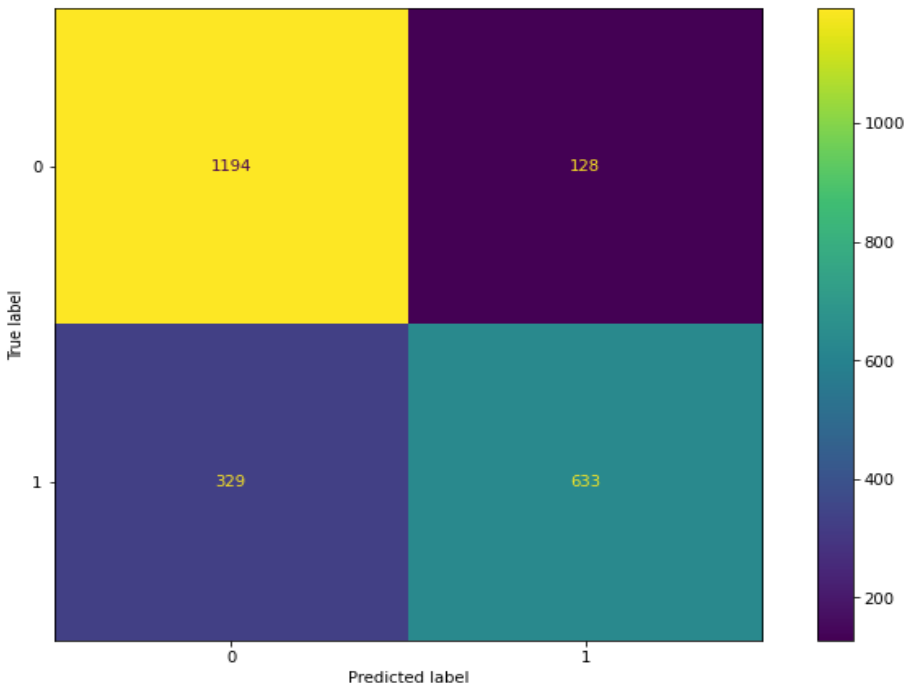
```
predictions = nb_model.predict(test_x)
acc_score = accuracy_score(test_y,predictions)
pre_score = precision_score(test_y,predictions)
rec_score = recall_score(test_y,predictions)
print('Accuracy_score: ',acc_score)
print('Precision_score: ',pre_score)
print('Recall_score: ',rec_score)
print('--'*35)
cr = classification_report(test_y,predictions)
print(cr)
```

```
Accuracy_score:  0.7999124343257443
Precision_score:  0.8318002628120894
Recall_score:  0.658004158004158
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.78      0.90      0.84      1322
           1       0.83      0.66      0.73       962

    accuracy                           0.80      2284
   macro avg       0.81      0.78      0.79      2284
weighted avg       0.80      0.80      0.80      2284
```

Şekil 3.23 Naïve bayes modeli tahminleri

Naïve bayes 79%'luk bir başarı göstermiştir.

```
plot_confusion_matrix(nb_model,test_x,test_y)
plt.show()
```



Şekil 3.24 Naïve bayes modelinin çarpıklık matrisi grafiği

Çarpıklık modeline bakıldığında gerçek tweetlerinin tahmini diğer modellere göre daha kötü sonuçlandığı görülmektedir.

3.5.1.4 XGBoost Yöntemi

XGBoost veya aşırı gradyan artırma, sıralı karar ağaçları tabanlı makine öğrenme algoritmalarında gelişmiş performans ve hıza sahip iyi bilinen gradyan artırma tekniklerinden biridir. XGBoost, Tianqi Chen tarafından oluşturulmuş ve başlangıçta Derin Makine Öğrenimi Topluluğu grubu tarafından sürülmüştür. Yarışmalarda uygulamalı makine öğrenmesi için kullanılan en yaygın algoritmadır ve yapılandırılmış ve tablosal verilerde kazanan çözümlerle popülerlik kazanmıştır. Açık kaynaklı yazılımdır [13].

Uygulama

```
import xgboost
xgb = xgboost.XGBClassifier()
xgb_model = xgb.fit(train_x,train_y)
```

Şekil 3.25 XGBoost modeli eğitimi

Python'da xgboost diye bir kütüphane geliştirilmiştir ve uygulamada kendi kütüphanesi kullanılarak eğitim yapılmıştır.

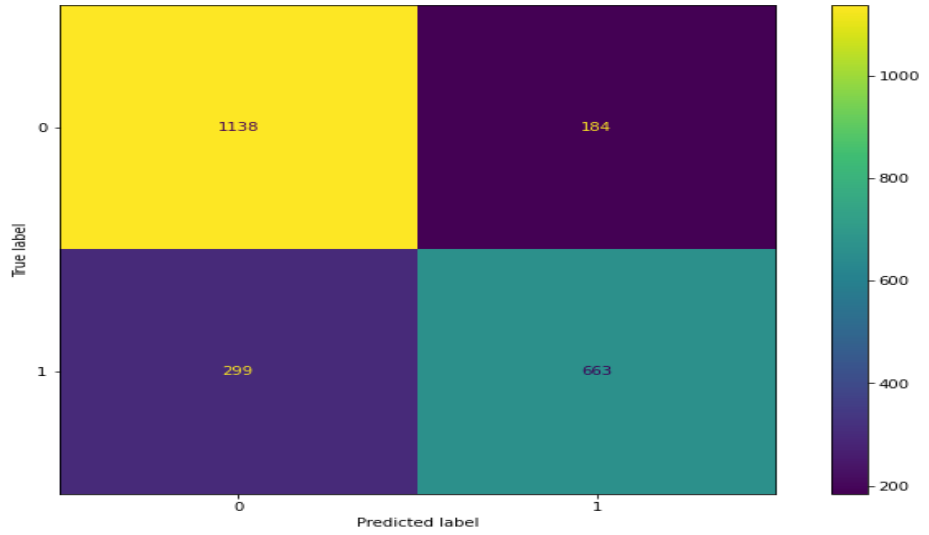
```
predictions = xgb_model.predict(test_x)
acc_score = accuracy_score(test_y,predictions)
pre_score = precision_score(test_y,predictions)
rec_score = recall_score(test_y,predictions)
print('Accuracy_score: ',acc_score)
print('Precision_score: ',pre_score)
print('Recall_score: ',rec_score)
print('--'*35)
cr = classification_report(test_y,predictions)
print(cr)
```

```
Accuracy_score:  0.7885288966725044
Precision_score:  0.7827626918536009
Recall_score:  0.6891891891891891
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.79      0.86      0.82      1322
           1       0.78      0.69      0.73       962

    accuracy                           0.79      2284
   macro avg       0.79      0.78      0.78      2284
weighted avg       0.79      0.79      0.79      2284
```

Şekil 3.26 XGBoost modeli tahminleri

Şekil 3.26'da da görüldüğü üzere xgboost model 79%'luk bir başarım göstermiştir. XGBoostun en iyi tarafı çok hızlı bir algoritma olmasıdır. Diğer algoritmalara göre eğitim süresi çok daha kısa sürmüştür.

```
plot_confusion_matrix(xgb_model,test_x,test_y)
plt.show()
```



Şekil 3.27 XGBoost modelinin çarpıklık matrisi grafiği

Kullanılan 4 tane farklı algoritma sonucunda 4 algoritma arasında çok yüksek bir fark olduğu söylenemez.

3.6 Modellerin Değerlendirilmesi

Modellerin değerlendirilmesi bölümünde kullanılan metrikler doğruluk ve kesinlik metrikleridir. Kesinlik metriğinin kullanılma sebebi bu tezde 1(gerçek doğal afet tweetleri) olan hedef değişkeninin tahmininin daha önemli olmasıdır.

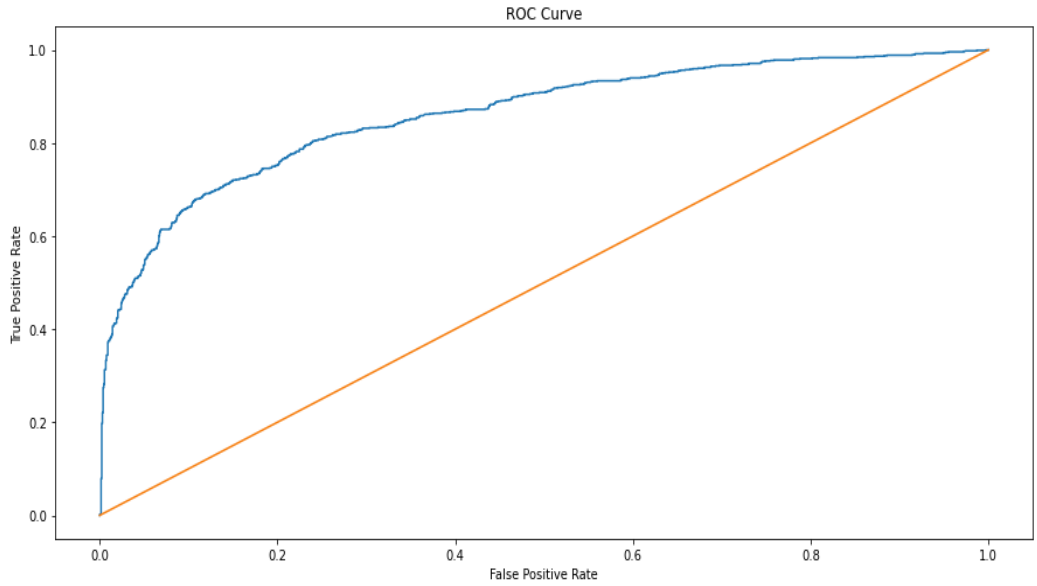
Tablo 3.2 Sonuçların değerlendirilmesi

```
results
```

	Models	Accuracy	Precision
0	Logistic Regression	0.788967	0.777778
1	Naive Bayes	0.799912	0.831800
2	Random Forest Classifier	0.785902	0.770905
3	XGBoost Classifier	0.788529	0.782763

Yukarıdaki tabloda da görüldüğü üzere Naïve bayes en iyi sonucu veren modeldir. Bu tezde kullanılan metriklere bakıldığında en iyi sonucu Naïve bayes modeli verdiği için bu modelin kullanılmasına karar verilmiştir.

```
predictions_probability = nb_model.predict_proba(test_x)
fpr,tpr,thresholds = roc_curve(test_y,predictions_probability[:,1])
plt.figure(figsize=(15,7))
plt.plot(fpr,tpr)
plt.plot([0,1])
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```



Şekil 3.28 Naïve Bayes modelinin ROC grafiği

Bölüm 2.5'te de anlatıldığı gibi ROC eğrisi pozitif tahminlerin doğruluğunu göstermektedir. Yukarıdaki şekilde 80% doğrulukla pozitif tahminlerin başarılı olduğu görülmektedir. Bu doğruluk bir sınıflandırma modeli için geçerlidir.

4
SONUÇ VE ÖNERİLER

Naïve bayes yönteminin kullanılmasına karar verildikten sonra test verisi kullanılarak modelin hiç görmediği bir veri seti ile tahmin yapılmıştır. Test verisine 3. Bölümdeki Yöntem kısmında anlatılan ön işleme adımları uygulanmış ve test verisi tahmin için hazır hale getirilmiştir.

```
pred = nb_model.predict(test)
submission = pd.DataFrame({"id": Id, "target": pred})
submission.to_csv("submission.csv", index=False)
```

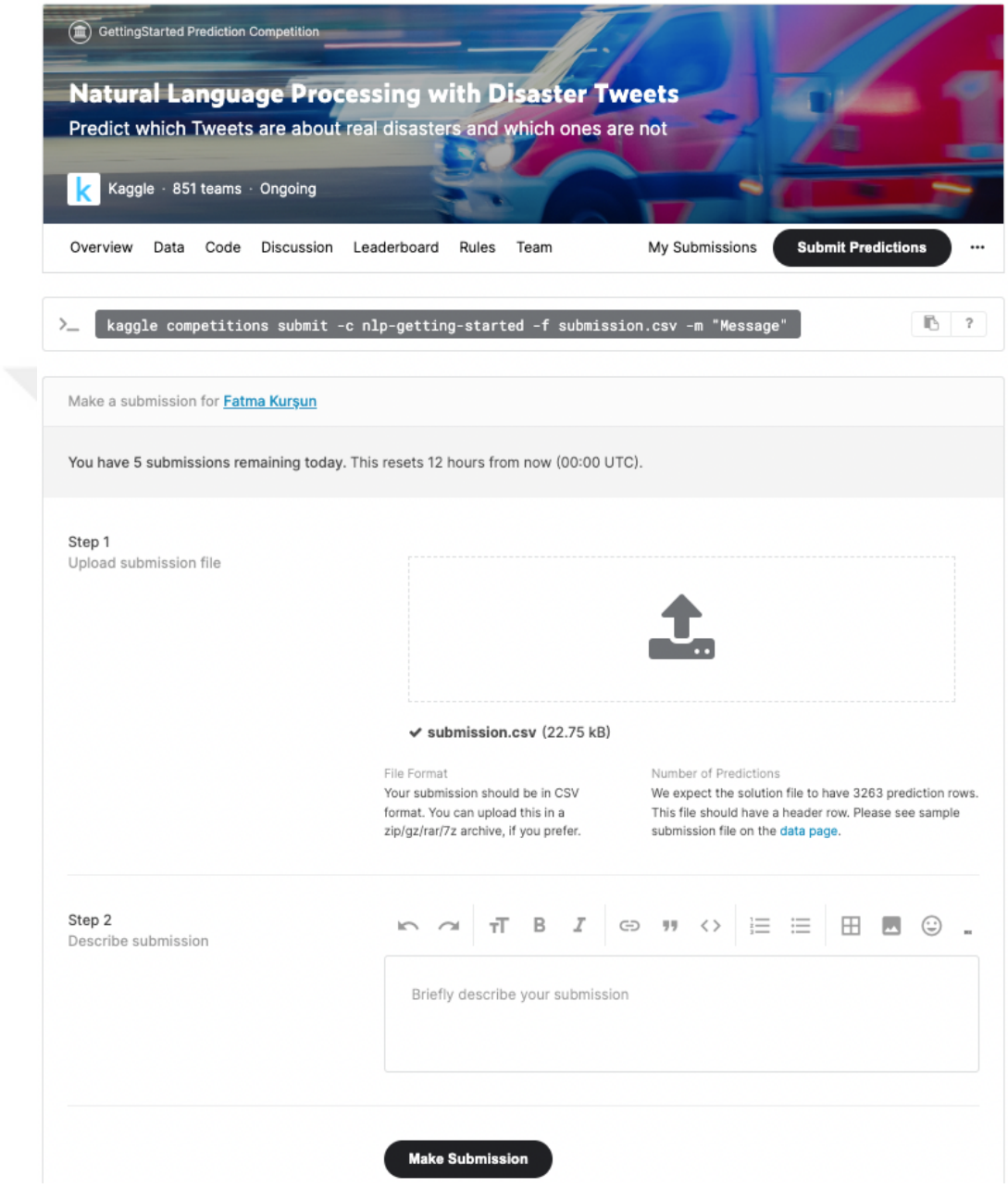
```
submission.head(10)
```

	id	target
0	0	1
1	2	1
2	3	0
3	9	1
4	11	1
5	12	1
6	21	0
7	22	0
8	27	0
9	29	0

Şekil 4.1 Test verisi tahmini

Şekil 4.1'de görüldüğü üzere modele daha önce görmediği bir veri ile tahminleme yaptırılmıştır. Daha sonra sonuçlar bir csv dosyasına aktarılmıştır. Bu tez

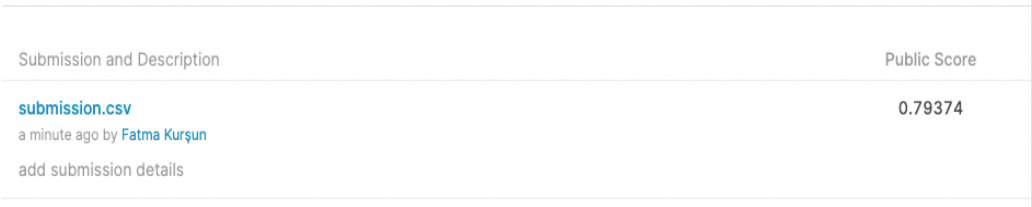
kapsamında veri Kaggle internet sitesinden alınmıştı ve şimdi test verisinin sonuçları bu siteye yüklenerek sonucun doğruluğu gözlemlenebilecektir. Bu adımdan sonra bu csv dosyası Kaggle sitesine yüklenerek sonuçlarına bakılmıştır.



Şekil 4.2 Sonuçların Kaggle'a yüklenmesi

Yukarıdaki resim Kaggle'a aittir ve test seti daha önce modelin görmediği bir veridir. Bu veri kullanılarak tahminler yapılmış ve sonucu görmek için tahmin sonuçları siteye yüklenmiştir. Şekil 4.2'de de görüldüğü üzere tahminler siteye

yüklenerek yarışma sahipleri tarafından değerlendirilmesi yapılmıştır. Bu adımdan sonra sonuçlara bakılacaktır.



Şekil 4.3 Sonuç

Şekil 4.3'te görüldüğü üzere tahmin dosyası yüklendiğinde doğruluk oranı test verisi üzerinde 79% çıkmıştır. Bu sonuç bize modelin test verisi üzerinde de doğru çalıştığını göstermektedir. Naïve bayes modeli kullanılarak bir kullanıcının attığı tweetin 80% ihtimalle gerçek olup olmadığı tahmin edilebilmektedir.

Bu çalışmada önemli olan nokta eğer bir tweet gerçek bir felaket tweeti ise bu hayati bir önem taşımaktadır. Atılan bir tweetle bir insan hayatı kurtulabilir. Fakat bunun yanı sıra eğer bir sahte felaket tweeti için birimler (ambulans, polis itfaiye vb.) harekete geçerse en basit sonucu ile bu büyük bir zaman kaybı olacaktır ama daha büyük sonuçlara sebep de olabilir. Tüm bu sebepler, bu modelin doğru çalışmasının önemini göstermektedir.

En basit hali ile model şöyle çalışacaktır;

- Kullanıcılar tarafından içerisinde felaket haberi geçen bir tweet atılacak,
- Bu tweet uygulanan ön işlemlerden geçirilecek (NLP)
- Ön işlemden sonra model harekete geçecek ve tweetin gerçek veya sahte olduğunun tahminini yapacak
- Eğer model gerçek felaket tweeti tahmini yapılırsa harekete geçilecek ve gerekli birimlere haber verilecek

Bu çalışmanın devamı adına bir uygulama geliştirilerek, ambulans itfaiye ve polis gibi acil durumlarda ulaşılması gereken birimlerle iletişim kurulabilir. Bu uygulama kullanılama modeli içerip bir tweet atıldığında o tweetin gerçek olup

olmadığının tahmini yapılarak eğer gerçekse atılan tweetin konumu bu birimlere gönderilerek daha hızlı reaksiyonlar alınabilir.

KAYNAKÇA

[1] Wikipedia, https://en.wikipedia.org/wiki/Twitter (Erişim zamanı: 12,02, 2021).

[2] A. H. Hoss Belyadi, *Machine Learning Guide for Oil and Gas Using Python,* UK: Joe Hayton, 2021.

[3] Analytics in Diamag, https://analyticsindiamag.com/xgboost-internal-working-to-makedecision-trees-and-deduce-predictions/.,(Erişim zamanı: 10,11, 2020).

[4] Wikipedia, https://en.wikipedia.org/wiki/Machine_learning., (Erişim zamanı: 03,01, 2022).

[5] Rubin, V.R., Conroy, N. J., Chen, Y. and Cornwell, S.*, Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News*, 2016.

[6] B. and. A. S. Horne, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news.," In Proceedings of the international AAAI conference on web and social media, 2017.

[7] A. and. P. D. and. N. J. Cobo, *Identifying Relevant Messages in a Twitter-based Citizen Channel for Natural Disaster Situations,* 2015.

[8] Kaggle, https://www.kaggle.com/competitions/nlp-getting-started., (Erişim zamanı: 11,12, 2020).

[9] G. Bonaccorso, *Machine Learning Algorithms*, India: Packt Publishing Ltd., 2017.

[10] B. Lantz, *Machine Learning with R*, UK: Packt Publishing Ltd., 2013.

[11] IBM, https://www.ibm.com/topics/unsupervisedlearning?mhsrc=ibmsearch_a&mhq=what%20is%20unsupervised%20learning., (Erişim zamanı: 05,11, 2021).

[12] Medium, https://medium.com/deep-learning-turkiye/regresyon-ve-s%C4%B1n%C4%B1fland%C4%B1rmada-hata-metrikleri-143a40c6b656, (Erişim zamanı: 01,31, 2020).

[13] Wikipedia, https://en.wikipedia.org/wiki/XGBoost., (Erişim zamanı: 09,24, 2020).

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. F. KURŞUN ve F. NOYAN TEKELİ, “NLP Kullanımı ile Sahte Felaket Tweetlerinin Tahmini” Icesser 5th Internatıonal Congress On Empirical Social Sciences, Economy, Management and Educatıon Researches, ICESSER 5th International Congress, June 19, 2022 Rome- s.182