VERİ MADENCİLİĞİ YÖNTEMLERİ KULLANILARAK OTEL YORUMLARININ DUYGU ANALİZİ

Duygu analizi, bir yazı parçasının olumlu veya olumsuz olup olmadığını belirleme yöntemidir. Bir konuşmacının fikrini veya tutumunu inceleyen fikir madenciliği veya metin madenciliği olarak da bilinir. Bu teknolojinin ortak kullanım noktası, insanların belirli bir konu hakkında nasıl hissettiklerini keşfetmektir.

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

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

Bu tez çalışmasında başlıca amaç elimizde olan veri setinin eğitilip veri analizin yapılması ve insanların fikirlerinin olumlu veya olumsuz olduğunu belirlemektir. İnsanların bir konu hakkında görüşlerinin hangi yönde olduğunu öğrenmek oldukça önemlidir. Kullandığımız veri setinde Tripadvisor sitesinde bulunan Türkiyedeki otel yorumlarını bulundurmaktadır. TripAdvisor popüler bir otel, seyahat ve restoran web sitesidir. Yapılan çalışmalar sonucu en uygulanabilir yöntemin makine öğrenmesi yöntemlerinden SMO ve Naive Bayes algoritması olduğu görülmüş ve bu algoritmaya göre elde edilen sonuçlar kullanılmıştır.

Anahtar Kelimeler : Veri analizi, Veri madenciliği, SMO, Naive Bayes, Makine Öğrenmesi.

SENTIMENT ANALYSIS OF HOTEL COMMENTS USING DATA MINING METHODS

Sentiment analysis is a method of determining whether a piece of writing is positive or negative. It is also known as idea mining or text mining that examines a speaker's idea or attitude. The common point of use of this technology is to discover how people feel about a particular subject.

It refers to the use of idea mining, natural language processing, text analysis, computational linguistics and biometrics to systematically identify, measure and examine emotional states and subjective information.

Generally expressed emotion analysis aims to determine the attitudes of a speaker, writer or other subject to a subject, general contextual polarity, or emotional response to a document, interaction or event. This attitude can be a judgment or evaluation or intended emotional communication.

The main aim of this thesis study is to educate the data set which is in our hands and to make data analysis and to determine whether the ideas of people are positive or negative. It is important to learn what direction people think about a subject. Emotion analysis study, a popular hotel, travel and restaurant web site, which is necessary data set was taken here. As a result of the studies performed, it was seen that the most applicable method was the SMO and Naive Bayes algorithm for machine learning methods and the results obtained according to this algorithm were used.

Keywords **:** Data analysis, Data mining, SMO, Naive Bayes, Machine Learning.

BÖLÜM 1

GİRİŞ

Her geçen gün atılan tweet'ler, yazılan bloglar, haberler, sosyal medya paylaşımları, filmlere ve ürünlere yapılan yorumlar, çevrimiçi ortamda ciddi miktarda verinin birikmesine neden olmaktadır. Bu verilerin alınıp en doğru şekilde işlenerek analiz edilmesi ve yorumlanması gerekmektedir. Amerika'da yapılan bir çalışmaya göre orada yaşayan insanların %73'ü çevrimiçi ortamda okudukları ürün yorumlarını ciddiye almaktadır. Özellikle büyük şirketler bu bilgilerin yorumlanıp, analiz edilmesine oldukça özen göstermektedirler. Şirketler, ürünleri ve kendileri hakkındaki bu yorumları internetten hızlı bir şekilde elde edip, analiz etmek için Duygu Analizini kullanmaya başlamışlardır. Duygu Analizi üzerinde daha çok pazarlama, insan ilişkileri ve reklam firmaları çalışmaktadırlar. Hatta Duygu Analizi, son zamanlarda politikada bile kendine yer bulmaktadır. Siyasiler, seçimlerden önce çevrimiçi ortamda kendileri hakkındaki yorumları önemsemekte ve seçim taktilerini bu yorumlara göre belirlemektedirler.

Son zamanların en popüler araştırma alanlarından bir olan Duygu Analizi hakkında yayınlanmış 7000'den fazla makale bulunmaktadır. Birçok yeni girişim şirketi ve köklü firma, yeni çözümler üretmek adına ciddi yatırımlar yapmakta ve yeni departmanlar kurarak Duygu Analizi konusunda çalışmalarına ağırlık vermektedirler.

Duygu Analizi (Sentiment Analysis / Opinion Mining) konusunda yapılan birçok çalışmada bir ürünün ya da servisin yorumlarını analiz etmek, açıklayabilmek için basit ifadeler kullanır. Buna rağmen diller arası farklılıklar ve bir kelimenin bir cümle içindeki kullanımına göre farklı anlamlara gelmesi, yazı dilindeki kültürel farklar ve kelime kullanımlarına göre içeriğin farklılaşması Duygu Analizi çalışmaları için birer problem haline gelebilmektedir. Çoğu zaman bir cümle içindeki kelimelerden anlam bütünlüğü çıkarmak bir dokümana bakıp yorumlamaktan daha zor hale gelebilmektedir. Duygu ve düşüncelerini metin 2 dokümanlarına yansıtan insanların yazdıklarını analiz etmek ve doğru kararı vermek bilgisayarlar için her zaman kolay olmamaktadır.

Veri kirliliği günümüzün en büyük sorunudur. Doğru veriyi bulmak, doğru şekilde işlemek ve analiz etmek çok kolay değildir. Çevrimiçi ortama göre İngilizce içerikli veriye ulaşmak ve onu işlemek diğer dillere nispeten daha kolay olmaktadır. İngilizce içerikli olarak yapılan Duygu Analizi çalışmaları, Türkçe içerikli yapılan çalışmalara göre sayıca daha fazladır. Bu durumun en temel nedenlerinden biri Türkçe metinler üzerinde dil bilgisi çalışmalarının azlığıdır. Üstünde akademik çalışmalar yapılacak Türkçe içerikli dokümanların yapısal olarak bozukluğu da çalışmaların azlığına neden olmaktadır. Bu tez çalışmasında İngilizce metinler üzerinde uygulanmış teknikler, Türkçe metinler üzerinde uygulanmıştır.

Duygu Analizi birçok çalışmanın bir araya gelmesi ile oluşmuştur. Asıl olarak Metin Madenciliği yöntemlerinden faydalanılsa da Duygu Analizinde Veri Madenciliği, Makine Öğrenmesi ve Doğal Dil İşleme yöntemleri oldukça sık kullanılmaktadır. Verinin bir kaynaktan çekilmesi, işlenmesi ise Metin Madenciliği çalışma alanlarının konusuna girmektedir. Ayrıca dokümanlardaki kelimelerin analiz edilmesi, Doğal Dil İşleme ve bu sürecin otomatik hale getirilerek bilgisayarlar tarafından yapılabilmesi Makine Öğrenmesinin çalışma alanına girmektedir.

Bu tez çalışmasında, Metin Madenciliği ve Makine Öğrenmesi Teknikleri kullanılarak Türkçe metinler üzerinde Duygu Analizi çalışmaları yapılmıştır. Veri Madenciliği'nin alt dallarından biri olan Metin Madenciliği, önceden bilinmeyen ancak kullanılabilir ve üzerinde çalışıldığı zaman anlamlı olacak bilgileri çok büyük veriler içinden çıkarma yöntemidir. Bu tez çalışmasında çevrimiçi ortamdan çekilen büyük ve kirli veriden, Metin Madenciliği yardımıyla anlamlı veriler elde edilmeye çalışılmış ve makine öğrenmesi algoritmaları kullanılarak da Duygu Analizi çalışmaları yapılmıştır.

Makine Öğrenmesi alanında iki yaygın yöntem; Naive Bayes, SMO yöntemleridir. Bu tez çalışmasında Naive Bayes ve SMO yöntemi kullanılarak Türkçe metinler üzerinde deneyler gerçekleştirilmiştir.

BÖLÜM 2

2.1 Duygu analizi

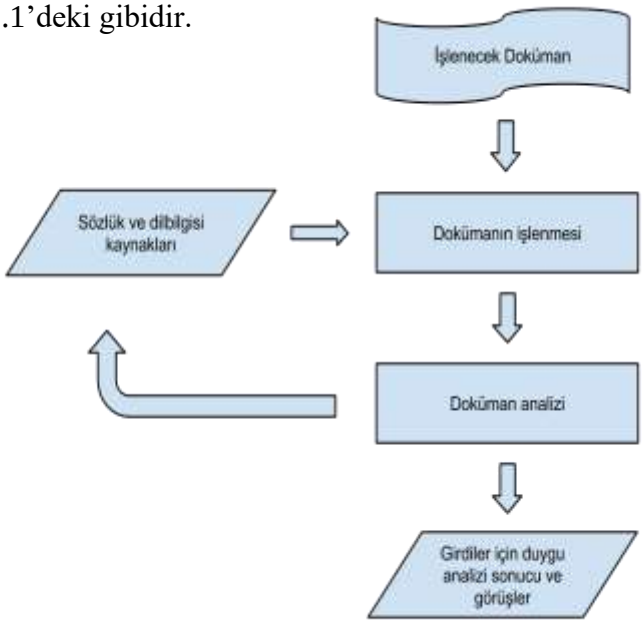
Duygu Analizi'nin amacı bir konu hakkında yapılan yorumların pozitif mi, negatif mi, tarafsız mı olduğunu belirlemektir. Bu yorumlar yazılı ya da sözlü olabilir. Dokümanların konusu hakkında ise herhangi bir sınırlama yoktur. Buradaki asıl amaç, makinelere bu yorumların pozitif mi negatif mi olduğunu, bir insan gibi tahmin ettirebilmek ve bunu otomatik olarak yapabilmesini sağlamaktır.

İnternetin gelişmesi ile Duygu Analizi konusunda çalışmalar son zamanlarda hız kazanmıştır. Özellikle Web 2.0 ile dinamik içerikli sayfaların üretilmesi ve veri tabanına olan önemin artması, veriyi daha anlamlı ve kullanışlı hale getirmiştir. 2002 yıllında Bo Pang ve Lillian Lee tarafından "Thumbs up? Sentiment Classification using Machine Learning Techniques" başlıklı çalışması Duygu Analizi konusunun yapı taşlarından biri olarak görülmektedir. Bu çalışmada Internet Film Veri Tabanı (IMDB Internet Movie Database)'dan veriler çekilmiş ve Naïve Bayes, Destek Vektör Makinesi (Support Vector Machine), Maksimum Entropi, Maksimum Entropi (Maximum Entropy Classification) Makine Öğrenmesi Algoritmaları ile Duygu Analizi çalışmaları yapılmıştır. İngilizce dışında başka dillerde farklı çalışmalar da gerçekleştirilmeye çalışılmıştır. Türkçe için Duygu Analizi konusunda şimdiye kadar yapılmış akademik çalışma sayısı çok azdır. Türkçe metinler üzerinde yapılan bir tez çalışmasında İngilizce çalışmalarda denenen yöntemler, iki yeni Türkçe veri seti üzerinde denenmiş ve %85 başarı sağlanmıştır.

Genel yorumlar ve görüşler için Duygu Analizinin yapılması zor ve anlamsızdır. Kişisel görüşlerin belirtildiği dokümanları yorumlamak ve analiz etmek Duygu Analizi'nin çalışma alanına girer. Örneğin, bir kullanıcı bir cep telefonu hakkında "*Ciddi anlamda güzel bir telefon. Hem hafif, hem çok dayanıklı. Hep böyle bir telefona sahip olmak istemiştim. Fiyatı da performansına göre çok uygun. Bir kaç gün kullandıktan sonra batarya süresinin çok gitmediğini fark ettim. Böylesi bir telefonun batarya süresinin neden az olduğunu anlamıyorum. Ekran çözünürlüğünün diğer telefonlardan yüksek olması iyi ancak batarya süresinin kısalığı kullanımı zaman zaman zorlaştırıyor. Eğer yeni bir telefon almak istiyorsanız bu telefonu*

deneyebilirsiniz." yorumunu yapmış olabilir. Burada Duygu Analizi konusunda birçok çıkarım yapılabilir. Cep telefonu hakkındaki iyi ve kötü yorumlar bu metin içinde mevcuttur. Ancak cümle olarak bakıldığında, yorumların hem iyi hem de kötü olduğu görülmektedir. Ayrıca bahsedilen telefon diğer telefonlarla karşılaştırılmıştır. Telefonun bazı özellikleri kendi içindeki diğer özellikleri ile de karşılaştırılmıştır. Duygu Analizi çalışmalarını yapmadan önce doküman üzerinde ne gibi bir yaklaşım sergileneceğini belirlenirse daha iyi sonuç alınacağı kesindir.

Duygu Analizi çalışmalarına herhangi bir dosya türünde (pdf, html, xml, word vb.) doküman dizisi alınarak başlanır. Bu doküman girdisi, ön işleme yöntemlerinden olan hecelere ayırma, kelime grubu etiketleme, bilgi çıkarma ve kelimeler arası ilişki kurma kullanılarak sadeleştirilir. Dokümanı daha anlaşılır kılmak için dokümanın kendi diline ait sözlükler ve dil ile alakalı diğer kaynaklar kullanılabilir. Sadeleşen ve sözlükler üzerinden analizleri yapılan doküman üzerinde artık hangi Duygu Analizi Yaklaşımı uygulanacağına karar verilir. Bu yöntemler araştırmanın şekline göre farklılık gösterebilir. Uygulanan yöntemden sonra çıkan veri, son kullanıcının anlayacağı şekilde hazırlanır ve sunulur. Duygu Analizi çalışmalarının akış diyagramı Şekil 2.1'deki gibidir.



Şekil 2.1. Duygu Analizi Sistem Mimarisi

Duygu Analizi'nin alt çalışma alanlarını, Doküman Seviyesinde (Document-Level), Cümle Seviyesinde (Sentence-Level), Özellik Temelli (Aspect-Based) ve Karşılaştırmalı (Comparative) olarak sıralayabiliriz.

2.1.1 Veri sözlüğü tabanlı yaklaşımlar

Dünya üzerinde konuşulan veya yazılan her dilin kendisine göre kuralları ve yapıları vardır. Diller kelime tabanlı veya ek tabanlı olabilmektedirler. Bir dildeki kelimelerin en az bir anlamı vardır. Tüm kelimelerin anlamlarını olumlu, olumsuz veya belirsiz anlamlı olarak sınıflandırmak mümkündür.

Duygu analizinde kullanılan sözlük tabanlı yaklaşımlarda her kelimenin anlamları referans alınarak tüm metnin duygusu sınıflandırılmaktadır. Kelime tabanlı dillerde bu yöntemlerin kullanılması duyguların tespit edilmesinde başarılı sonuçlar verirken Türkçe gibi sondan eklemeli dillerde sözlük tabanlı duygu analizi çok iyi sonuçlar vermemektedir. Bunun temel nedeni ise kelimenin köküne gelen her yapım ekinin kelimenin duygu özelliği üzerinde büyük değişikliklere neden olmasıdır. Örneğin; Türkçe bir kelime birden fazla yapım veya çekim eki alabilmektedir. Her ek için kelimelerin duygularının derecelendirilmesi mümkün görülmemektedir.

Veri sözlüğü tabanlı yaklaşım, metni analiz etmek için kullanılan fikir veri sözlüğünü bulmaya bağlıdır. Bu yaklaşımda iki farklı metot vardır. Sözlük tabanlı yaklaşım eş anlamlılık ve zıt anlamlılıklardan oluşan sözlükleri araştırır ve fikir tohum kelimeleri bulmaya dayanan bir yaklaşımdır. Korpus tabanlı yaklaşım fikir kelimelerinin tohum listesi ile başlar ve sonra geniş bir korpusta diğer fikirleri bulur. Bu istatistiksel ve semantik metotlar kullanılarak yapılabilir.

2.1.2 Makine öğrenmesi yaklaşım

Makine öğrenmesi bilgisayarlara, programlama yapılmadan, öğrenme yeteneği sağlayan bir yapay zekâ tekniğidir. MÖ, kendilerini geliştirmek için eğitebilen, yeni veriler ile kendilerini değiştirebilen bilgisayar programlarının geliştirilmesi üzerinde durur. Bilgisayarlara karmaşık örüntüleri algılatma ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmak, MÖ araştırmalarının odaklandığı konudur. MÖ, istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma gibi alanlarla yakından ilintilidir.

Genel olarak bir sınıflandırma problemi, MÖ'nde denetimli veya denetimsiz öğrenme algoritmalarıdır. Denetimli sınıflandırma yapılırken öncelikle hangi sınıfa ait

oldukları belli, önceden etiketli yeterince büyük bir eğitim kümesinin olması gerekir. Denetimli sınıflandırma algoritması (Naive Bayes, KDM, Karar Ağaçları (KA) vb.) bu eğitim kümesindeki örüntüleri öğrenerek bir model üretir.

2.1.3 Denetimli öğrenme

Denetimli Makine Öğrenmesi, etiketli verilerin eğitim kümesini temel almaktadır. Denetimli yaklaşımlarda bir sınıflandırıcı eğitim verisi ile eğitilip, test verisi üzerinde sınıflandırma başarısı ölçülür. Makine öğrenmesi yaklaşımlarında en çok kullanılan yöntemler denetimli yaklaşımlardır.

Denetimli öğrenme modelinde etiketli girdi verilerinden istenilen çıktı verileri elde edilmesi ve oluşan çıktı verilerinin istenilen değerlere yakınlığı önemlidir. Denetimli öğrenmede oluşturulan model ile bir grup girdi değerine karşılık onlara ait hedef değerleri verilerek aralarındaki ilişkiyi öğrenmesi ve hedef değerlere en yakın çıktıların üretilmesi amaçlanır.

2.1.3.1 Naive bayes (NB)

NB makine öğrenimi yöntemleri arasında en temel olarak kabul edilen bir sınıflandırma yöntemlerinden biridir. Dayanak noktası olarak Bayesian sınıflandırma yöntemi temellerine dayanmaktadır. Bu yöntemin en bilinir özelliği olayları olasılık hesaplamaları yaparak değerlendirmesidir. Bunun içinse önceden etiketlenmiş olarak sistemdeki eğitim kümesi verileri kullanmaktadır. Yöntem üzerinden test edilecek bir sorgulamada önceden verilmiş olan eğitim kümesini kullanarak bir olasılık değeri elde edip, bu skora göre de test verisinin hangi kategoriye benzediğini belirlemektedir.
Naive Bayesian ile Bayesian yöntemlerinin (Eşitlik 2.1) farkı ise yeni gelen test verisindeki bir değerin eğitim kümesi içerisinde yer almaması durumunda oluşacak sıfır olasılık değerine olan bakış açılarıdır. NB bu durumu sıfır vermek yerine, belirlenmiş bir eşik değeri ekleyerek sonuçtaki hassaslık oranının daha da yükselmesini sağlamaktadır.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Eşitlik 2.1 Bayesian Yöntemi Denklemi

P(A|B) ; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır.

P(B|A) ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır.

P(A) ve P(B) ; A ve B olaylarının öncelikli olasılıklarıdır.

NB P(B\A) P(A) değerinin sıfır olması durumunda sıfır yerine uygun görülen bir eşik değeri konularak hesaba dahil edilir. Günümüzdeki literatür çalışmalarında en temel yöntem olarak kabul gören NB duygu analizi ve diğer türdeki sınıflamalar için yaygın olarak kullanılmaktadır.

2.1.3.2 K-nearest neighbors (K-NN)

Sınıflandırma problemlerinde etkin olarak kullanılmakta olan yöntemlerden bir tanesidir. Belirlenen "k" değerine göre verilen sorgunun "k" birim komşuluğunda yer alan vektörleri tespit ederek sınıflama işlemini gerçekleştirir. Bu yöntemin doğru uygulanabilmesi için iyi bir eğitim kümesi oluşturulması şarttır. Eğitim kümesi bu yöntemin başarısındaki en önemli faktördür. Uygulanmasının kolay olması nedeniyle *k-NN* sınıflama problemlerinde sık sık kullanılmaktadır. Bilgi kirliliğinin olduğu dokümanlarda sınıflama kabiliyeti güçlüdür.

k-NN algoritmasının çalışma mantığı her bir sorgunun ayrı ayrı hesaplanmasını gerektirdiğinden dolayı bu yöntemin hesaplanma maliyeti çok yüksektir. En yakın komşuluk bağıntısına dayandığı için vektör uzayında ifade edilen terimlerin birbirine olan uzaklıkları *Manhattan* yöntemi (Eşitlik 2.2), *Euclidean* yöntemi (Eşitlik 2.3) ya da Minkowski yöntemi (Eşitlik 2.4) yardımıyla hesaplanır. k-NN eşitlikleri içinde gösterilen "k" değeri komşuluk derecesini, "x" değeri kategorinin vektörünü "y" değeriyse sonucun vektörünü temsil etmektedir. Verilen sorgudaki terim benzerlik oranı "1" değerine en yakın olan kategoriye eklenir.

$$\sum_{i=1}^{k} |x_i - y_i|$$

Eşitlik 2.2 Manhattan Yöntemi

$$\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

Eşitlik 2.3 Euclidean Yöntemi

$$(\sum_{i=1}^{k} (|x_i - y_i|)^q)^{1/q}$$

Eşitlik 2.4 Minkowski Yöntemi

2.1.3.3 TF-IDF Ağırlıklandırma

IR ve metin madenciliği alanlarında veri setlerindeki bilgi kirliliğini azaltmak için sıkça kullanılan terim çıkartma yöntemlerinden bir tanesidir. Cümlelerde yer alan kelimelerin ne sıklıkla kullanıldığına yani frekansına ve diğer dokümanlarda geçme sıklıklarını birlikte hesaplayarak, kategoriler için en önemli kelimelerin tespitini sağlar. Böylelikle test verisi üzerinde analiz edilecek kelimelerin sayısını azaltarak hem işlem süresinin kısalmasına hemde cümlelerin içerisinde çok sık kullanılan ve sistemin sonucuna katkıda bulunmayan zamir, bağlaç gibi istenmeyen kelime yapılarının sistemin sonucuna etki etmesini önleyecektir.

Term Frequency (TF) burada terimin sıklığını ifade etmektedir. Hesaplanırken dokümanda o terimin geçme sayısının (f), o terimin toplam dokümanlarda kaç kere geçmiş ise o sayıya (df) bölünmesiyle elde edilir. Inverse Document Frequency (IDF) ise analiz edilen kelimenin ne kadar eşşiz (Unique) olduğunu hesaplanması (Eşitlik 2.6) işlemidir.

IDF hesaplanırken “N” değeri toplam doküman sayısını temsil eder. “df” ise hesabı yapılan kelimenin kaç dokümanda yer aldığını temsil eder. Cümledeki *IDF* değerinin yüksek olması demek o kelimenin kullanılmasının, kategorinin belirlenmesi için çok değerli olduğunu gösterir. Örneğin “çok” kelimesi bir cümlede tek başına bir duygu ifadesi içermemektedir. Bu nedenle hem olumlu cümle hem de olumsuz cümlenin içerisinde sık olarak geçebilir. *TF-IDF* (Eşitlik 2.7) yöntemi tam bu esnada bu kelimenin ağırlığını düşük bularak önemsiz olduğunun tespitinde büyük rol oynar.

$$IDF = \log\left(\frac{N}{df}\right)$$

Eşitlik 2.6 IDF Hesaplanması

$$TF = \frac{f}{df}$$

Eşitlik 2.5 IDF Hesaplanması

$$TF - IDF = TF * IDF$$

Eşitlik 2.7 TF-IDF Hesaplanma Yöntemi

2.1.4 Denetimsiz öğrenme

Denetimsiz makine öğrenmesi etiketsiz verileri kullanarak sistemin eğitilmesini sağlar. Denetimsiz öğrenmede amaç veri setindeki örneklerin çıkışları bilinmediği için tanıma ve sınıflandırma değildir. Genellikle kümeleme, olasılık yoğunluk tahmini, öznitelikler arasındaki ilişkilerin bulunması ve boyut indirgeme gibi amaçlarla kullanılır.

Denetimsiz öğrenme modelinde kullanılan etiketsiz verilerin elde edilmesi, denetimli öğrenmede kullanılan etiketli verilerden daha kolaydır. Denetimsiz öğrenme metotları etiketli eğitim dokümanlarının bulunması zor olduğu durumlarda kullanılır. Denetimsiz öğrenmede kullanıcının sisteme herhangi bir müdahalesi söz konusu değildir. Sadece girdi verileri sisteme girilir, çıkış değerine karşılık gelen sonuçlar yani sınıflar bilinmez. Denetimsiz öğrenmede temel amaç eldeki verilerden ortaya bir model çıkarmak için sistemin eğitilmesidir.

2.2. Makine Öğrenimi Yöntemleri ile Yapılan Çalışmalar

İstatistiksel yöntemler ve sınıflayıcılar ile farklı diller ve farklı veri kümeleri için yapılmış çalışmalar mevcuttur. Pang ve arkadaşları Naïve Bayes (NB), Maksimum Entropy (MEf) ve Support Vector Machine (SVM) (Destek Vektör Makinesi) algoritmaları yardımıyla, elle seçtikleri öznitelikler ve 1.400 yorumlu veri kümeleri ile duygu analizi yapmışlar ve ortalama %82 başarı elde etmişlerdir.

Pang ve Lee çoklu sınıf SVM regresyon algoritmaları ile %60 civarında başarı elde etmişlerdir. Çalışmada yıldız ve sembollerle etiketlenmiş verilerde sınıflama yapma problemi irdelenmiştir.

Hu ve Liu ticari ürünler hakkında yapılan yorumları derlemiş olumlu ve olumsuz olmak üzere sınıflamışlardır. Yaptıkları sınıflama sonucunda %70 civarı başarı elde etmişlerdir.

Turney ise web üzerinden topladığı 4 farklı konu hakkındaki 410 yorumu cümlelerin polaritesine göre PMI-IR algoritması ile sınıflamıştır. Turney çalışma sonucunda ortalama %74 başarı elde etmiştir.

Pang ve Lee önce veriden öznel ifadeleri çıkarmış sonra bu ifadelerin polaritelerini hesaplamış ve SVM, NB sınıflayıcılar ile ortalama %85 başarı elde etmişlerdir.

2.3. Sözlüksel benzeşim yöntemleri ile yapılan çalışmalar

Sözlüksel benzeşim yöntemlerinde ise iki farklı bakış açısı vardır; duygu sözlüğü ve duygu derlemi. İstatistiksel yöntemlerden makine öğrenmesi ile duygu analizi yapılabilmesi için gerekli olan öznitelikleri elde etmek için bir derleme sahip olmak gerekmektedir. Bu derlemi oluştururken her sınıfa ait tohum kelimeler, her bağlam için yeniden oluşturulmaktadır. Duygu derlemi bakış açısı ile sadece bir bağlam için duygu içeren kelime listesi oluşturulmaktadır. Oluşturulan listelerin konu bağımlı olması, tohum listenin el yordamıyla hazırlanmasından dolayı için eksik ya da hatalı olma riski, vakit ve iş gücü kısıtlarından ötürü; kapsamlı ve doğru bir duygu sözlüğü oluşturma ihtiyacı ortaya çıkmıştır.

Hu ve ekibinin %84 başarı elde ettikleri çalışmalarında el yordamıyla toplanmış bir miktar olumlu ve olumsuz tohum kelimeler listelenir. Zıt ve eş anlamları gösteren WordNet gibi bir sözlük alınır ve liste tohum kelimelerin eş ya da zıt anlamlıları bulunarak genişletilir. Sonra yine el yordamıyla liste arındırılır. Buna benzer bazı araştırmalarda ise, son aşamada listedeki hatalı elemanları temizleme işlemi, kelimelerin temsil yeteneğini tespit eden istatistiksel yöntemler yardımıyla yapılmaya çalışılmıştır. Kelimelere gelen eklerle zıt anlam tespit etmeyi deneyen Mohamed ve ekibi bazı zıt anlamları listeye eklemeyi başarmıştır. Yine başka bir teknikte ise Kamps ve ekibi tıpkı Williams ve ekibinin çalışmalarında olduğu gibi mesafe tabanlı algoritmalarıyla her kelimenin duygu yönünü, olumlu ve olumsuz elemanları bulunan bir tohum listesine olan mutlak uzaklıklarına göre tespit etmeyi denemiştir.

Steinberger ve arkadaşları çalışmalarında birçok dilde duygu sözlüğü oluşturmak için yarı otomatik bir yöntem sunmaktadır. İki farklı dil için oluşturulmuş olan sözlükler otomatik yöntemlerle üçüncü dillere çeviri yapılmıştır. Çalışmada İngilizce, İspanyolca, Arapça, Çekçe, Fransızca, Almanca, İtalyanca ve Rusça üzerine çalışmalar yapılmıştır. Türkçe için denenmiş tamamlanamamıştır. Yaklaşık 2.000 civarı kelimelik sözlükler insan eliyle kontrol edilmiş ve ortalama %76 başarı sağlanmıştır.

2.4 Türkçe için yapılmış duygu analizi çalışmaları

Türkçe duygu analizi tüm bu gelişmelerden birkaç yıl sonra gündeme gelmiş hazır sözlükler ve derlemler olmadığı için önce makine öğrenmesi yöntemleri ile yapılmış ve başarılı olmuştur. Alandaki ilk çalışmada Eroğul makine öğrenmesi algoritmalarından SVM ve doğal dil işleme teknikleri ile film yorumlarını N-gram model kullanarak olumlu ve olumsuz kategorilerde sınıflamış ve %85 başarı elde etmiştir.

Nizam ve arkadaşları çalışmalarında gözetimsiz makine öğrenmesi yöntemleri ile duygu sınıflaması yapılmaktadır. Twitter verisi üzerinde çalışılmış tüm kelimeler öznitelik olarak seçilmiştir. Ngram özellikleri kullanılarak yapılan sınıflama sonucunda eşit dağılımlı veri kümesi ile yapılan deney dengesiz veri kümesiyle yapılan

deneyden daha başarılı olmuş ve en fazla %72 başarı elde edilmiştir. Boynukalın [İngilizce bir veri kümesini insan eliyle Türkçeye çevirip ve bir takım Türkçe verileri de elle etiketleyerek elde ettiği veriler üzerinde makine öğrenmesi teknikleri ile sınıflamalar yapmış ortalama %80 civarı başarı elde etmiştir. Vural ve ekibi yaptıkları çalışmada İngilizce için yapılmış ve birçok dile çevrilmiş olan SentiStrength altyapısını kullanarak duygu analizi yapmıştır. Uygulamanın kullandığı yaklaşık 1000 kelimelik sözlüğü elle çevirmiş ve makine öğrenimi yöntemleri ile sınıflama yaparak duygu analizini gerçekleştirmiştir.

Meral ve Diri yaptıkları çalışmada 8.500 civarı Twiti elle etiketlemiş, kelime yakalama bakış açısı ve makine öğrenmesi yöntemlerinden Rastgele Orman (*Random Forest*), Destek Vektör Makinesi (SVM) ve Naïve Bayes (NB) sınıflayıcılarıyla Duygu Analizi yapmışlardır. Korelasyon tabanlı öznitelik seçimi yaparak %90 civarında başarı elde etmişlerdir.

Mayda ve Aytekin çalışmalarında sosyal medyada rekabet analizi için karşılaştırma görevine yönelik bir fikir madenciliği modeli geliştirmiştir. Bu amaçla karşılaştırma siteleri, YouTube ve teknoloji forumlarından iz sürme tekniği ile karşılaştırma ifadesi içeren 100 yorum manuel olarak derlenmiş ve bu yolla bir test veri tabanı oluşturulmuştur. Sadece anma metriği ile sunulan sonuçlara göre sistem %70 civarı başarılı olmuştur.

Çakmak ve arkadaşları çalışmalarında Türk dili için kelime kökleri ve cümle bazında duygu ilişkilerini incelemiştir. Bulanık mantık kullanılmış ve hazır bazı kelime listeleri Türkçeye çevrilmiştir. Bazı istisnalar dışında kelime kökleri ve cümleler arasında yüksek ilişki olduğu tespit edilmiştir. Çakmak ve arkadaşları yine bir başka çalışmada deneyleri bulanık mantık tip 2 ye göre yeniden yapmışlar ve aynı sonuçları almışlardır.

Kaya ve arkadaşları çalışmalarında farklı kaynaklardan topladıkları haber yorumlarını birden çok makine öğrenimi yöntemiyle duygu analizi yapmış ve ortalama %77 başarı sağlamışlardır. Politika haber yorumları bağlamında duygu analizi yaparken yaşanan özel sorunları ele almışlar ve bazı çözümler üretmişlerdir. Akba ve ekibi ise öznitelik seçme algoritmaları ve SVM yardımıyla film veri kümesi üzerinde tamamen gözetimsiz makine öğrenmesi teknikleri ile duygu sınıflaması yapmış ve yaklaşık %84 başarı elde etmişlerdir.

Balahur ve arkadaşları çalışmalarında analiz yapılacak veriler makine çevirisi ile İngilizceye çevrilmiş ve İngilizce için hazır olan makine öğrenmesi yöntemleri ve öznitelik seçme yöntemleriyle analiz yapılmıştır. Verilerin bir kısmı ana dili Türkçe olan katılımcılar tarafından İngilizceye çevrilmiş ve aynı sonuçlar alınmıştır. İster makine çevirisi isterse insan çevirisi ile İngilizceye çevrilen verilerin barındırdıkları duyguyu koruduğu tespit edilmiştir. Türkçe duygu analizi sonucunda elde edilen başarı %60'a yakındır.

Akbaş yaptığı çalışmada Twitter üzerinden topladığı verilerle makine öğrenimi yöntemleri kullanarak duygu analizi yapmıştır. Öznitelik seçiminde, oluşturduğu duygusal kelime listesini kullanarak hibrit bir yöntem kullanmıştır. Çalışma sonucunda pozitif ve negatif duygu sınıflamasında %85 civarı başarı elde etmiştir.

Çetin ve Amasyalı yaptıkları çalışmada Türkçe Twitter verisi üzerinde birçok deney gerçekleştirmiş ve deney sonuçlarını karşılaştırmışlardır. Sonuç olarak eğiticili yöntemlerin daha başarılı olduğunu tespit etmişlerdir. Ortalama %60 civarı başarı elde etmişlerdir. Çetin ve Amasyalı yine başka bir çalışmalarında ise makine öğrenimi yöntemlerinden NB ile sınıflama esnasında eğitim kümesinin sayısını %50 azaltıp aktif öğrenme algoritmaları uygulamıştır. Tüm eğitim kümesine göre daha başarılı olmuşlar ve %64 başarı elde etmişlerdir. Özsert ve Özgür yaptıkları çalışmada duygu analizinde çok önemli olan kelime polaritelerini belirlemek üzere birtakım deneyler yapmıştır. Kelime polaritelerini belirlemek için yarı otomatik bir yöntem geliştirmişlerdir. İngilizce için oluşturulmuş olan tohum kelimeleri Türkçeye çevirmiş adım adım öğrenme metodu ile kelimelerin polaritelerini tespit etmişlerdir. Tespit ederken elde ettikleri başarı yaklaşık %90 civarındadır.

Tüm bu bilgiler ışığında hala Türkçe için açık kaynak bir duygu sözlüğü ve deneysel bir veri kümesi yoktur.

2.4.1 Türkçe ve duygu analizinde karşılaşılan zorluklar

Doğal dil işlemede üzerine sıklıkla çalışılan dillerden yapısal olarak farklı olan dillerde duygu analizi, özellikle de hedef tabanlı duygu analizi alanında çalışmalar yapma çok daha zordur. Türkçe bu tür dillere önemli bir örnek teşkil etmektedir. Bunun en büyük sebeplerinden biri Türkçe'nin sondan eklemeli ve zengin

biçimbilimsel yapısıdır. Batılı dillerde birçok kelimeden oluşan bir cümle, Türkçede tek bir kelime ile ifade edilebilmektedir. (Örneğin "he/she can not bring"-"getiremez"). Diğer dillerde sentaktik ilişkileri belirleyen (ör: from, to vb.) sözcükler, Türkçe'de biçimbilimsel seviyede ortaya çıkmaktadırlar. Bu özellikleri sebebiyle veriye dayalı (corpus based) tekniklerde veri seyrekliği problemi ortaya çıkmaktadır. Bu da aynı ifadenin temsil oranını düşürmekte, eğitim kümesinde yer almayan bazı sözcük görünüş biçimleri (word surface forms) için çıkarım yapılamamasına yol açmaktadır. Türkçe'nin diğer bir önemli özelliği olan öğelerin cümle içi serbest dizilimi, duygu analizi çalışmalarında zorluklara yol açmaktadır. İngilizce gibi dillerde kelimeler arası ilişkileri incelerken sıralama önem ifade ederken, Türkçede serbest dizilim yüzünden kelimeler arası ilişkileri bulmak güçleşmektedir. Hedef tabanlı duygu analizi problemi özelinde bir örnek vermek gerekirse, bir hedef terimin duygusunu bulabilmek için yalnızca dizilim esasına göre kelimeye komşu diğer kelimelerden faydalanmak yeterli gelmemektedir.

Türkçe	İngilizce
Okula gidiyorum.	I'm going to school

Şekil 2.8 Türkçe ve İngilizce Cümle Yapıları

2.5. Veri madenciliği

Veri Madenciliği, birçok disiplinin bir araya gelmesi ile oluşmuş bir araştırmasahasıdır. Bu disiplinler arasında; Yapay Zekâ, Makine Öğrenmesi, İstatistik ve Veri Tabanı Sistemlerine Erişim Yöntemleri yer almaktadır. Son zamanlarda çevrimiçi ortamdaki verinin öneminin artmasından dolayı Veri Madenciliği'ne olan ilgi de aynı ölçüde artmış ve artmaya devam etmektedir.

Karmaşık veriden anlamlı bir bilgi çıkarmak için izlenmesi gereken adımlar şunlardır;

1. Veri Seçimi: Üzerinde çalışılacak verinin veri tabanından ya da herhangi bir kaynaktan alınmasıdır.

2. Veri Entegrasyonu: Eğer veri birden çok kaynaktan alınıyorsa, bu verilerin birbirleri ile olan birleştirme işlemidir.
3. Veri Temizleme: Veri içindeki tutarsızlıkların ve gürültünün giderilmesidir.
4. Veri Dönüştürme: Verinin özetleme veya derleme işlemlerine tabi tutularak, kullanıma uygun hale getirilmesidir.
5. Veri Madenciliği: Veri örüntülerini ortaya çıkarmak için akıllı yöntemlerin uygulandığı önemli bir süreçtir.
6. Örüntü Değerlendirmesi: Bilgiyi temsil eden ilginç örüntülerin özel ölçümlere dayanarak belirlenmesi işlemidir.
7. Bilgi Sunumu: Ortaya çıkarılan bilginin görselleştirme ve bilgi sunum yöntemleri kullanılarak kullanıcıya gösterilmesi adımıdır.

1. ve 4. adımlar arası "Veri Ön İşleme Süreci" olarak adlandırılmaktadır. Bu süreç, üzerinde madencilik yapılacak verinin temizlenmesi ve işleme hazır hale getirilmesi için gereken adımları içermektedir. "Veri Madenciliği" adımı ise kullanıcı ile etkileşimli bir şekilde gerçekleştirilebilir. Bu adımdan sonraki örüntülerin değerlendirilmesi aşamasında, ilgi alanı dışındaki örüntüler belirli ölçümlerle ayıklanır. Önemli olanlar ise bir sonraki aşamada kullanıcıya değişik yollarla gösterilebilir. Veri Madenciliği, çok büyük veri kümelerinde standart yöntemlerle görülemeyecek bilgi ve örüntüleri ortaya çıkardığı için önemlidir. Veri Madenciliği genellikle küçük veri kümeleri ile ilgilenmez.

2.6. Metin madenciliği

Metin Madenciliği, farklı yazılı kaynakların bir araya getirdiği verinin, otomatik olarak alınması ve o veri içinde yeni bir bilgi keşfetme işidir. Asıl amacı belli bir metin üzerinde belli bir yapısı olan veriyi bulup, o verinin ilgili metin içinden çıkarılmasıdır. Metin Madenciliği Teknikleri dört temel kategoriye ayrılır: Sınıflandırma (Classification), Birliktelik Analizi (Association Analysis), Bilgi Çıkarım (Information Extraction) ve Kümeleme (Clustering). Sınıflandırma işlemi, nesnelerin daha önceden bilinen sınıflara ya da kategorilere dahil edilmesidir.

Birliktelik Analizi ise sıklıkla birlikte yer alan ya da gelişen sözcük veya kavramların belirlenmesini amaçlar. Böylece doküman içeriğinin ya da doküman kümelerinin anlaşılmasını sağlar. Bilgi Çıkarım Teknikleri yardımlarıyla dokümanların içerisindeki yararlı veri ya da ifadeler bulunmaya çalışılır. Kümeleme Analizi, doküman kümelerinin temelini oluşturan yapıların keşfedilmesi amacıyla uygulanmaktadır.

Metin Madenciliği çalışmaları, metin kaynaklı literatürdeki diğer bir çalışma alanı olan Doğal Dil İşleme (Natural Language Processing, NLP) çalışmaları ile çoğu zaman beraber yürütülmektedir. Doğal Dil İşleme çalışmaları daha çok Yapay Zeka altındaki dil bilimine dayalı çalışmaları kapsamaktadır. Metin Madenciliği çalışmaları ise daha çok istatistiksel olarak metin üzerinden sonuçlara ulaşmayı hedefler. Metin Madenciliği çalışmaları sırasında çoğu zaman Doğal Dil İşleme Teknikleri kullanılarak, özellik çıkarımı yapılmaktadır.

Genel olarak metin madenciliği dört adımdan oluşmaktadır;

2.6.1 Metin koleksiyonu oluşturma

Metin madenciliğinde atılacak ilk adım ilgili dokümanların toplanmasıdır. Metin madenciliği ‘derlem’ olarak da adlandırılan doküman koleksiyonu ile başlamaktadır. En basit şekliyle derlem, metne dayalı dokümanların herhangi bir grubu olarak tanımlanabilir. Geleneksel veri tabanı ile kıyaslandığında metin koleksiyonu, yapısal olmayan ham verilerden oluşmaktadır. Ham veriler, günümüzde özellikle internet ortamları kullanılarak toplanmaktadır.

Dokümanların oluşturduğu derlemlerin yapısı statik veya dinamik olabilmektedir. Derlemlerin eğer başlangıçtaki durumları değişmeden kalıyorsa statik yapıda oldukları söylenebilir. Buna karşın zaman içerisinde yeni dokümanlar ekleniyor veya dokümanlar güncelleniyorsa, derlemin dinamik olduğunu söylemek mümkün olacaktır. Örneğin, belirli tarihler arasında atılan e-maillerin kaydedilmesi gerekli olabilir.

2.6.2 Metin koleksiyonu oluşturma

Yapılandırılmamış ham veri üzerinde çalışılması elde edilen sonuçlar üzerinde farklılıklar meydana getirebileceği gibi analiz sürecinin de uzamasına neden olacaktır. Bu sebeple ham verinin metin madenciliğine hazırlanması için gereksiz kelimelerden arındırılması, yazım hatalarının düzeltilmesi, köklerine ayrılarak yanlış kullanılmış kelimelerin düzeltilmesi gibi önişleme tekniklerine ihtiyaç duyulmaktadır.

Veri kalitesinin iyi olması hatasız veya en az hatalı sonuçlar almayı mümkün kılmaktadır. Bu nedenle metin önişleme, veriden daha anlamlı bilgi üretebilmek için metin madenciliğinin en önemli adımıdır. Metin önişleme sürecinde, veri temizlemenin yanı sıra veriyi uygun formata dönüştürme gerçekleşir. Bu aşamanın sonunda metinler yapılandırılmış formata dönüştürülmüş olmaktadır.

Türkçe gibi sondan eklemeli dillerde kelimeye eklenecek her bir ek anlamı değiştirmekte ve aynı gövdenin ek almış hallerinin farklı anlamlarda olması değerlendirmeyi zorlaştırmaktadır. Bunun yanında tek bir Türkçe kelimeden çok sayıda farklı anlamda kelimeler oluşabilmektedir. Bu durumda farklı metin önişleme teknikleri gerekmektedir. Metin önişleme teknikleri sırasıyla açıklanmıştır:

İşaretleme: Metin ile ilgili çalışmalarda atılacak ilk adım işaretleme işlemidir. Ham metin verilerinde bulunan bütün tümcelerin işaretlere bölünmesidir. Elimizdeki ham verinin daha kaliteli hale getirilmesi, veri boyutunun da küçültülerek işlem kabiliyetimizin artması adına metnin sadeleştirilmesi gerekmektedir. Bu sebeple her bir kelimeyi ayrıştırabilmemiz için toplam metni sadeleştirmek ve işaretlememiz gerekmektedir. İşaretleme işleminde dokümanlar bölüm, kısım, paragraf ve hatta hecelere ayrılabilir. Metin içerisinde bulunan noktalama işaretleri, tek başına boşluk karakterinden fazla olan boşluklar ve diğer metine konu olmayan karakterlerin temizlenmesi şeklinde işaretleme gerçekleşir. Böylelikle metin olarak geriye kelimeler ve kelimeler arası birer boşluklar halindeki sade metin kalır.

Gövdeleme: İşaretlemeler belirlendikten sonra bu işaretlerin her birinin standart forma çevrilme işlemidir. Gövdeleme işleminin gerekli olup olmaması uygulamaya bağlıdır. Bazı uygulamalarda fayda sağlayacağı gibi gerek duyulmayan gövdeleme işlemi fazladan yapılmış olabilmektedir.

Sözlük Oluşturma: Genel anlamıyla sözlük, kelimeleri ve işaretleri bir arada barındıran, sözcüğün kökünü esas alan eserlerdir. Sözlükte yer alan kelime sayısından çok kelimenin niteliğini önemlidir. Sözlük oluştururken yer kaplayacak gereksiz kelimelerin alınmaması performans açısından büyük önem taşımaktadır. Örneğin bir kelimenin hem tekil hem çoğulunu sözlüğe dahil etmek yerine gövdeleme işlemi ile sözlük boyutunda büyük bir azalma sağlanabilmektedir. Sözlük oluşturma da köke kadar gövdeleme işlemi hafif anlam kaymalarına neden olabilmektedir. Joker kelimeler sözlüklerin kelime sayısını azaltmakta böylece işlem süreçleri kısalıp daha başarılı sonuçlar elde edilebilmektedir.

2.6.3 Veri analizi

Yapılandırılmış formata dönüştürülmüş olan metinler, geleneksel analiz yöntemleri ile analiz edilebilmektedir. Veri analizinde kullanılan yöntemler uygulamanın içeriğine göre değişebilir. Önemli olan konuyla ilgili toplanan veriyi uygun yöntemle özetlemek ve araştırma ile ilgili sağlıklı çıkarımlarda bulunabilmektir. Toplanan veriler araştırma tasarımının türüne ve araştırmanın amaçlarına göre farklı analiz yöntemleri gerektirir. Yöntemlerin seçimi analiz sonuçlarında önemli rol oynamaktadır.

2.6.4 Değerlendirme ve yorumlama

Verilerin analizinden elde edilen sonuçların değerlendirilip, uygun ve anlaşılır bir şekilde sunulmasıdır. Veri analizinde birden çok yöntem kullanılmış ise sonuçlar karşılaştırmalı olarak değerlendirilebilir. Sonuçların kullanıcıya sunulması tablo, şekil gibi araçlar kullanılarak değerlendirme daha açıklayıcı hale getirilebilmektedir.

2.6.5 Sosyal medyada metin madenciliği

Sosyal medya bilgi ve deneyimleri paylaşmak amacıyla tek yönlü paylaşımdan çift taraflı ve eş zamanlı paylaşımlara ulaşılmasını sağlayan sosyal etkileşim

alanlarının bütünüdür. Sosyal medya üzerinde yapılan tüm paylaşım, diyalog ve bilgi içerikleri sosyal medyayı oluşturur. Michael Mandiberg (2012: 2) sosyal medya kavramının birden fazla kavramla ilişkili olduğunu savunurken, sosyal medyayı kullanıcı tarafından oluşturulan kurumsal medya olarak tanımlamaktadır.

Sosyal ağ kullanıcısının artmasıyla birlikte sosyal medya üzerindeki veri akışı da artmaktadır. Bu veri artışına bağlı olarak işlenmemiş, yapısal olmayan veri miktarı da artış göstermektedir. Bu verilerin anlamlı hale dönüştürülmesi ve bilgi çıkarımıyapılabilmesi için işlenmesi gerekmektedir. Bu da veriden anlam çıkarabilme özelliği olan duygu ifadelerinin analizi yöntemlerini sosyal medya verilerine yöneltmiştir.

Tüketici verileri, ürün hakkında bilgi, duygu ve düşünce takibi, yorum ve şikayetler, bilgi paylaşımı gibi birçok iletişimin sosyal ağlar üzerinden yapılması sosyal medyanın, üzerinde en çok veri barındıran platform olmasına neden olmuştur ve metin madenciliği çalışmalarını kaçınılmaz hale getirmiştir.

Sosyal medyadaki veriler çeşitlilik ve ulaşılabilirlik açısından kolaylık sağlarken, birçok zorluğu da beraberinde getirmektedir. Sosyal medyada kullanılan dilin zaman içerisinde değişmesiyle kısaltmalar, sosyal medyaya özgü terimler ve birçok yazım hataları metin analizi için zorlayıcı hale gelmiştir. Sosyal medya üzerindeki duygu ve düşünce ifade eden terimlerin değerlendirilmesi ve yorumlanması çok uzun zaman alabilmektedir. Bir sosyal medya kullanıcısının bir ürün hakkında yapılan yüzlerce yorumu okuması ve buna bir karar vermesi zaman alan ve zahmetli biri süreçtir. Bu kapsamda doğal dil işleme yöntemleri kullanılarak elde edilen düzgün metinler metin madenciliği teknikleri ile hızlı ve verimli analizler sağlayacaktır. Bu tez kapsamında sosyal medya ağı olan Twitter üzerinden alınan veriler, önce doğal dil işleme teknikleri kullanılarak kelimeler düzenlenmiş daha sonra düzenlenen veriler kullanılarak veri madenciliği teknikleri ile hızlı ve verimli bir şekilde duygu analizi yapılması amaçlanmıştır.

BÖLÜM 3

Veri madenciliği aşamalarında ön işleme aşamalarını kolaylıştırmak için C# programlama dili kullanılarak görsel arayüz uygulaması geliştirilmiştir. Böylelikle ön işleme aşamalar kolaylıkla yapılabilecektir. Uygulama arayüzünde sunulan farklı menü seçenekleri ile kullanıcı istediği seçenekleri kullanarak ön işleme aşamalarını gerçekleştirebilecektir.

3.1 Weka

Makine Öğrenmesinin en popüler araçlarından biri olan WEKA (Waikato Environment for Knowledge Analysis) hemen hemen birçok makine öğrenmesi algoritmasını bünyesine barındırır. Java Programlama Dili ile geliştirilmiştir ve açık kaynak kodludur.

WEKA, tamamen modüler bir tasarıma sahip olup, içerdiği özelliklerle veri kümeleri üzerinde görselleştirme, veri analizi, iş zekası uygulamaları gibi işlemler yapabilmektedir. WEKA yazılımının kendisine özgü olarak bir .arff dosya biçimi vardır. Ayrıca WEKA yazılımının içerisinde CSV dosyalarını da ARFF dosya formatına çevirmeye yarayan özellikler mevcuttur.

Temel olarak aşağıdaki üç Veri Madenciliği işlemi WEKA ile yapılabilir:

• Sınıflandırma (Classification)

• Kümeleme (Clustering)

• Birliktelik Kuralı Analizi (Association Rule Analysis)

Ayrıca yukarıdaki işlemlere ilave olarak, Veri Ön İşleme (Data Pre-Processing), Görselleştirme (Visualization) yardımıyla veri kümeleri üzerinde ön ve son işlemler yapılabilir. Son olarak WEKA kütüphanesinde veri kümelerini içeren dosyalar üzerinde çalışan çok sayıda hazır fonksiyon bulunmaktadır.

```
@relation 'DuyguTest'

@attribute Text string
@attribute Duygu {T, F}

@data

'Çok yeni bir yapı. Çok etkili ve nazik personel.İyi kaliteli odalar ve restoranda yemekler güzel.',T
'Yemekler, Servis, Temizlik hepsi 10 numaraydı. Herşey için teşekkür ederiz. Ebru Yaman.',T
'Asla 5 yıldızı haketmiyor. Odalar cok eski ve bakımsız. Servis kötü.',F
```

Şekil 3.1 Arff Veri Dosyası Örneği

ARFF (Attribute Relationship File Format) dosya yapısı, WEKA'ya özel olarak geliştirilmiştir ve dosya metin yapısında tutulmaktadır. Dosyanın ilk satırında, dosyadaki ilişki tipi (relation) tutulmakta olup, ikinci satırdan itibaren de veri kümesindeki özellikler (attributes) ve türleri yazılmaktadır. Özelliklerin hemen ardından veri kümesi yer alır ve veri kümesindeki her satır bir örneği (instance) ifade etmektedir. Veri kümesindeki her örneğin her özelliği arasında virgül ayıracı kullanılmaktadır.

Şekil 3.1'deki örnek kodda, eğitim kümemiz için kullanılan otel yorumlarının olumlu, olumsuz değerleri bir dosya içerisinde değerlendirilmiş örnek içerecek şekilde gösterilmiştir. Bu değerler, tip olarak sözel değerler olduğundan "string" olarak ifade edilmiştir. String dışındaki değerler aşağıdaki gibi tiplerde de olabilir:

• Küme Değerleri: Tahmin değeridir ve bir tanım kümesi alır. Örneğin, tahmin şeklinde tanımlanan bir değer, tanım kümedeki {güneşli, yağmurlu, sisli} değerlerinden birisini alabilir.

• Real: [Reel Sayılar] kümesinden bir değer verileceğinde kullanılır. Örneğin, sıcaklık değeri 22,8 şeklinde ondalıklı değer olarak ifade edilmek istenirse, tip olarak nümerik yerine reel kullanabiliriz.

• String: Veri kümesinin bu özelliğinin serbest yazı şeklinde olabileceğini ifade eder. Özellikle Metin Madenciliği çalışmaları için sıkça kullanılan bir tiptir.

• Date: Veri kümesinin bu özelliğinin tarih olduğunu ifade eder. Örneğin, veri kümesindeki kişilerin doğum tarihi veya örneklerin toplanma tarihi gibi özelliklerin tutulmasında kullanılabilir.

3.2. Otel yorumları üzerinde duygu analizi

Çevrimiçi ortamlarda gerek sosyal medyalarda gerek otel sitelerinde Türkçe yorum sayısının hızla artması ve buna bağlı olarak şirketlerin kendi otelleri hakkında yapılan yorumların ne gibi bir düşünce içerdiğine (olumlu/olumsuz) önem vermesi bu

tezin motivasyonu olmuştur. Yapılan otel yorumalarının İngilizce içerik yüzdesi göz ardı edildiğinde, Türkçenin çevrimiçi ortamdaki yerinin Türkçe adına yapılan çalışmaların ışığında geliştirilen yeni yöntemler ile Duygu Analizi hakkında yapılan çalışmaların sayısı arttırılmaya çalışılmıştır.

3.2.1 Çalışmada kullanılan veri

Bu tez çalışması için gerekli olan olumlu ve olumsuz içerikli verinin, otele yapılan yorumların bulunduğu sitelerden elde edilebileceği düşünülmüştür. Türkiye'nin en büyük web sitelerinden biri olan tripadvisor.com.tr de buluann otellere, kullanıcıların oteller hakkında yaptıkları yorumlar Json formatında alınmıştır.

Kullanıcılar tripadvisor.com.tr'in üzerinde ilgilendikleri otellerin yorumların üzerlerine yorumlarını yapmış ve bu yorumlar Json formatında saklanmıştır.

Bu yorumların daha sonra olumlu, olumsuz olarak değerlendirilmiştir. Yani bu tezin konusu için uygun bir veri kaynağı olduğunu göstermektedir.

3.2.2 Verinin çekilmesi

Bu tez çalışmasında kullanılacak veri, siteye yapılan yorumlar Json formantıda alınmıştı. Bu yorumlar C# sorguları ile çekilerek ayıklanmış ve yorumların olduğu yerlerden metinler elde edilmiştir ve sonra veritabanına kayıt edilmiştir. Alınan veriler belirlendikten sonra yorum ID'lerine göre sıralanmıştır ve yorumlar tablosu oluşturulmuştur. C# Programlama Dili ile çeşitli kodlar şekil 3.2.'deki gibi yazılarak veritabanına kaydedilmiştir.

Şekil 3.2 Yorumların Veritabanına Kayıt Edildiği Sorgular

```
foreach (var item in jsonYorum)
{

    /*otelList.Add(System.Net.WebUtility.HtmlDecode(item.OtelName));*/
    for (int i = 0; item.Yorumlar.Count > i; i++)
    {
        yrm = item.Yorumlar[i].Yorumu;//otelin i. yorumu
        yrm = System.Net.WebUtility.HtmlDecode(RemoveHtml(yrm));//html decode etme ve html tag temizleme islemi
        string kucuk = yrm.ToLower();

        foreach (char c in kucuk)
        {
            foreach (char cc in myChar)
            {
                if (cc == c)
                {
                    flag = 1;
                    break;
                }
            }
            if (flag == 0)
            {
                break;
            }
        }
        if (flag == 1)
        {
            /*
```

Yaklaşık 7000 adet yorum dosyası oluşturulmuştur. Bu dosyalar tek tek incelenmiş, hangi yorumun olumlu, hangisinin olumsuz olduğuna karar verilmiş ve yorumlar ayrıştırılmıştır. Bu yöntemde yorumlar tamamen bizim tarafımızdan okunmuş ve ayıklanmıştır. Burada amaç bir insanın bir yoruma verdiği kararı, bir makinenin ne kadar yüksek doğrulukta verebileceğini test etmek olduğu için verinin önce bizim tarafımızdan ayıklanması doğru bulunmuştur. Bütün yorumların olumlu ve olumsuz olmadığını düşündüğümüzde, pozitif ya da negatif kutbuna karar verilemeyen yorumlar tarafsız olarak değerlendirilmiştir. Daha sonra tarafsız seçtiğimiz yorumları elenmiştir. Örnek yorumlar şekil 3.3 ve şekil 3.4'te görüldüğü gibi

Çok yeni bir yapı. Çok etkili ve nazik personel.İyi kaliteli odalar ve restoranda yemekler güzel.

Şekil 3.3 tripadvisor.com'dan Alınmış Pozitif İçerikli Bir Yorum

Odalar temiz değil, eski ve bakımsız. Banyo akıyor duştan sonra tüm banyo gölet oluyor. Kahvaltı çeşitsiz özensiz

Şekil 3.4 tripadvisor.com'dan Alınmış Negatif İçerikli Bir Yorum

3.2.3 Verinin ön işlemesi (Pre-processing)

Çevrimiçi ortamdan çekilen verinin, çok dağınık ve karmaşık bir yapıda olduğu için verinin ön işlemeye sokulması ve temizlenmesi gerekliliği doğmuştur. Ön İşleme sadece verinin temizlenmesi değil, çalışmaya uygun hale getirilmesidir. Çalışmaya uygun şekilde bir verinin oluşturulması için öncelikle çalışmada nasıl bir veri kullanmak gerektiğine karar verilmiştir. tripadvisor.com.tr'den alınan yorumlar bir araya getirilerek eğitim kümesi oluşturulmuştur, bir dosya halinde deney durumlarına göre ön işleme (eğitim için) tabi tutulacak yorumların 7000 tanesi eğitilmiştir.

3.2.4 Algoritma çalışmaları

Dökümanları ön işleme sürecinden geçirdikten sonra WEKA ile analiz yapılmıştır. Daha önce de belirtildiği gibi WEKA ile analiz yapabilmek için dosyaların formatının .arff olması gerekmektedir. Eldeki .txt doküman dosyalarını .arff uzantılı şekil 3.5'deki gibi bir dosya haline çevirmek için C# Programlama Dili ile çeşitli kodlar şekil 3.6'daki gibi yazılarak, WEKA için bir format hazırlanmıştır.

```
@relation 'KategoriAnaliz'

@attribute Text string
@attribute Yeme {T,F}
@attribute Havuz {T,F}
@attribute Personel {T,F}

@data

' Alakart yemek muhteşemdi, şimdiye kadar en iyi yemek, kesinlikle tavsiye ederim',T,F,F,F,F,F,F,F,
'Bence bu tatil için tek kötü tarafı o kadar uzun olduğu için transfer ama kesinlikle buna değer, ot
' Bulunduğu bölgede her yere yakın, konaklamak için sessiz, exclusive ve panoramik',F,F,F,F,F,T,F,F,
' Hiç bir yenileme yapılmamış, tarihi doku bozulmasın diye olabilir fakat odadaki herşey çok eski',F
```

Şekil 3.5 Yorum.txt Dosyası

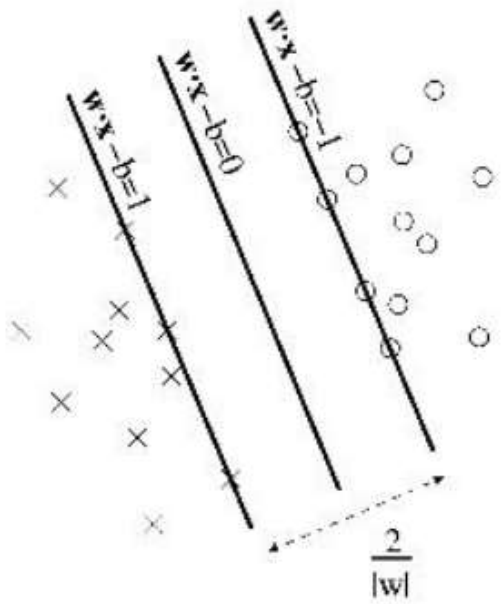
```
class Program
{
    static void Main(string[] args)
    {
        string filepath = @"D:\Tez\arff\Ciktilar\Anim.txt";

        YorumAnalizEntities db = new YorumAnalizEntities();
        FileStream fileStream = new FileStream(filepath, FileMode.OpenOrCreate, FileAccess.Read);
        StreamReader sw = File.OpenText(filepath);

        string satir;
        int x = 0;
        byte[] bytes;
        Encoding utf_8 = Encoding.UTF8;
        while ((satir = sw.ReadLine()) != null)
        {
            x++;
            char ayrac = ' ';
            string[] satirr = satir.Split(ayrac);
            satirr = satirr.Where(val => val != "" && val != "*" && val != "+" && val != "(normalized)").ToArray();
            AnimasyonSMO kayit = new AnimasyonSMO();

            bytes = Encoding.UTF8.GetBytes(satirr[1]);
            string myString = Encoding.UTF8.GetString(bytes);
            kayit.Kelime = myString;
            kayit.Agirlik = Convert.ToDouble(satirr[0]);
            db.AnimasyonSMO.Add(kayit);
            db.SaveChanges();
            Console.WriteLine(x + "satir eklendi");
        }
```

Şekil 3.6 C# ile Düzenlenen Yorum.txt Dosyası Sorguları

Bütün bu işlemler tüm yorum için tek tek yapılmıştır. Buradaki amaç WEKA ile kullanılan Naïve Bayes ve SVM (SMO) algoritmaları kullanılarak yapılan deneylerde iki sınıf için SMO %83.9, Naïve Bayes %72.5 doğruluk oranı en yüksek olarak SMO ile elde edilmiştir. Ancak SMO algoritmasında Confusion Matrix de FP Rate Naive Bayes'e göre yüksek çıkmıştır. Bunun sonucunda SMO'nun TP Rate değerini ve Naive Bayes'in FP Rate değerini değerlendirip hibritleşme yapılmıştır.

Bu tezdeki deneylerde, temelde SVM ve Naïve Bayes sınıflandırıcı yönteminden bahsetmek gerekirse:

SVM (Support Vector Machine): Sınıflandırma (Classification) konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. İşte SVM bu sınırın nasıl çizileceğini belirler. Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir. Örneğin aşağıdaki şekildeki iki grubu ele alalım:



Bu şekilde iki grup iki boyutlu bir düzlem üzerinde gösterilmiştir. Bu düzlemi ve boyutları birer özellik olarak düşünmek mümkündür. Yani basit anlamda sisteme giren her girdinin (input) bir özellik çıkarımı (feature extraction) yapılmış ve sonuçta bu iki boyutlu düzlemde her girdiyi gösteren farklı bir nokta elde edilmiştir.

Bu noktaların sınıflandırılması demek, çıkarılmış olan özelliklere göre girdilerin sınıflanması demektir.

Naïve Bayes (NB): Bu teknik adını İngiliz matematikçi Thomas Bayes'den (yak. 1701- 7 Nisan 1761) alır. Naïve Bayes, sınıflandırıcı örüntü tanıma problemine kısıtlayıcı bir önerme getiren olasılıkçı bir yaklaşımdır. Bu önerme, örüntü tanımada kullanılacak her bir tanımlayıcı nitelik ya da parametrenin istatistik açıdan bağımsız olması gerekliliğidir. Her ne kadar bu önerme; Naïve Bayes Sınıflandırıcısının kullanım alanını sınırlasa da genelde istatistik bağımsızlık koşulu esnetilerek kullanıldığında daha karmaşık Yapay Sinir Ağları gibi metotlarla karşılaştırabilir sonuçlar vermektedir. Her özellik için sınıflar içinde bulunma olasılıkları ve sınıfların veri üzerinde görülme olasılıklarını hesaplayarak karar veren bir modeldir. "Koşullu Bağımsızlık Kabulü" ile bir özelliğin bir sınıfta belirli bir olasılıkla geçmesi, bir başka özelliğin aynı sınıfta geçiş olasılığından etkilenmez ve o olasılığı etkilemez.

3.2.5 Tez çalışmasında kullanılan başarım ölçütü terimleri

Yapılmış olan çalışmaların tez vasfını kazanabilmesi için elde edilen bulguların benzer ya da daha önceden yapılmış çalışmalara olan üstünlüklerinin veya zayıf kalan yanlarının tespit edilebilmesi gerekmektedir. Bu nedenle tez çalışmasında elde edilen deney sonuçlarının tüm bilim insanları tarafından kabul görmüş başarım hesaplama yöntemleri kullanılarak bu değerlerin ifade edilmesi gerekmektedir. Bu tez çalışmasındaki deney sonuçlarınca elde edilmiş olan bulguların başarısını ifade etmek için kullanılmış olan başarım ölçütleri ve hesaplanma yolları alt başlıklar şeklinde açıklanmıştır.

Karışıklık matrisi

MÖ yöntemlerinin etkinliğini değerlendirmek amacıyla ise özellikle iki sınıflı sınıflandırma problemlerinde kullanılabilecek birçok metrik önerilmiştir. Ancak kesinlik (precision), duyarlılık (recall) ve doğruluk (accuracy) metrikleri yaygın olarak kullanılan değerlendirme yöntemleridir. Bu değerlerin elde edilebilmesi için iki sınıflı bir sınıflandırma probleminde muhtemel olan dört farklı durumun bilinmesi

gerekmektedir. Bu durumlar şekil 3.7'de verilen karışıklık (confusion) veya olasılık (contingency) tablosu ile elde edilmektedir.

Kategori (c)		Örnek Kategorisi	
		Evet	Hayır
Sınıflandırıcı Kararı	Evet	TP	FP
	Hayır	FN	TN

Şekil 3.7 Karışıklık matrisi

Duyarlık Değeri (Precision)

Deneyler sonucunda yapılmış olan ölçümlerin birbirine ne derecede yakın olduğunu gösteren başarım ölçütü terimidir. Bu değer hesaplanırken doğru sınıflandırılmış pozitif örnek sayısının (TP), toplam pozitif örnek sayısına (TP+FP) bölünmesiyle bu değere ulaşılır (Eşitlik 1.1). Bu değer her zaman 0-1 aralığında olmaktadır.

$$Duyarlık = \frac{TP}{TP + FP}$$

Anma Değeri (Recall)

Anma değeri hedefi tutturma oranı olarak bilinmektedir. Yani gerçekte ulaşılması gereken bilgilere ne oranda ulaşılmış olduğunu gösteren bir değerdir. Bu değer hesaplanırken doğru sınıflandırılmış ilgili pozitif örnek sayısının (TP), toplam ilgili belgelerin sayısına (TP + FN) bölünmesiyle bu değere ulaşılır (Eşitlik 1.2).

$$Anma = \frac{TP}{TP + FN}$$

Doğruluk (Accuracy)

Doğruluk değeri deneyde yapılan analizin gerçek değere ne kadar yakın olduğunu gösterir. Yani aynı şartlarda bu deney tekrarlanırsa yeni sonucun, önceki sonuca ne derecede benzer olacağını gösterir. Bu değer hesaplanırken doğru sınıflanmış pozitif ve negatif değerlerin sayısının toplamı (TP + TN), sınıflanan verilerin tümünün sayısına (TP + FP + TN + FN) bölünmesiyle elde edilir (Eşitlik 1.3).

$$Doğruluk = \frac{TP + TN}{TP + FP + TN + FN}$$

3.2.6 Kullanılan algoritmalraın Weka çıktıları

```
Accuracy0,895312694268308
TP rate0,948066126013724
TN rate 0,687921520539546
Sensitivity 0,948066126013724
specificity 0,687921520539546
precision  0,9480661

---------Confusion Matrix-------

       a    b     <-- classified as
    6079    333|    a = T
     509   1122|    b = F
```

Şekil 3.8 NaiveBayes Weka'daki Sonuçları

```
Accuracy0,900907621534253
TP rate0,957891453524641
TN rate 0,676885346413243
Sensitivity 0,957891453524641
specificity 0,676885346413243
precision  0,9578915

---------Confusion Matrix-------

       a    b     <-- classified as
    6142    270|    a = T
     527   1104|    b = F
```

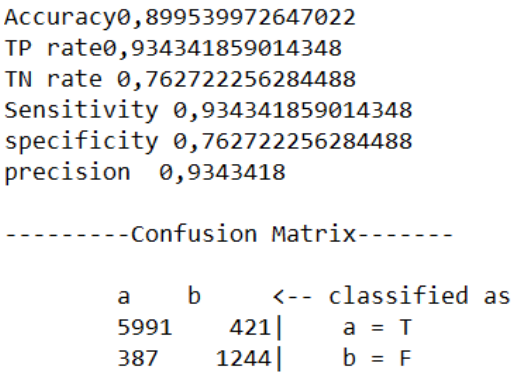
Şekil 3.9 SMO Weka'daki Sonuçları

WEKA Sınıflandırma Yöntemi ile Şekil 3.8 ve Şekil 3.9'daki gibi çıktılar üretilmektedir. Bu tezdeki deneylerde de bu ve benzeri çıktılar üretilmiştir. Ayrıca deneylerde F-Skor değeri dikkate alınmıştır. Bu değerin 1'e olan yakınlığı, deneyin o kadar başarılı olduğunu ifade etmektedir. F-Measure değerini bulmak için Kesinlik (Precision) ve Hassasiyet (Recall) değerlerinin bilinmesi gerekmektedir.

Alınan sonuçlar neticesinde Naïve Bayes ile sınıflandırdığımızda Şekil 3.8'deki gibi çıktılar üretmiştir. Burda Naïve Bayes için TN değeri SMO'nun değerine göre biraz daha iyi çıktığı görülmektedir ve yanlışı bulma olasılığının Smo'ya göre

biraz daha fazladır. Aynı şekilde eğitim kümesi Smo ile sınıflandırıldığında alınan sonuç TP değerinin NaïveBayes göre daha iyi çıktığı ve doğruyu bulma olasılığının NaïveBayes'e göre daha iyi olduğu tespit edilmiştir.

Bu iki algoritmaları birleştirerek (hibrit yöntemi) daha sağlıklı sonuçlar alınacağı gözlemlenmiştir. Smo algoritmasının Naïve Bayes algoritmasına göre TP'yi bulma olasılığı daha iyi olduğu veya diğer bir anlamda Smo'nun positif bir metnin doğru tutturma olasılığı daha iyi olduğu gözlemlenmiştir. Aynı şekilde Naïve Bayes algoritmasının TN'yi bulma olasılığı fazla olduğu için hibrit yapılmıştır. Hibrit yapıldığında daha da iyi sonuç alındığı Şekil.310'daki gibi görülmektedir.



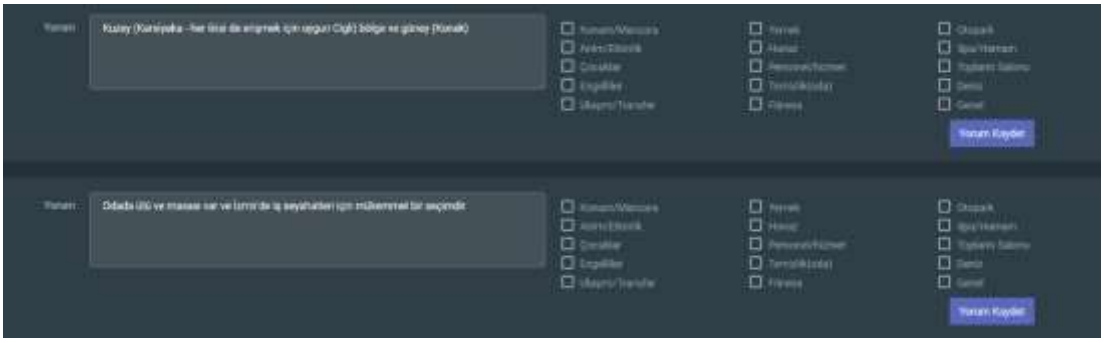
Şekil 3.10 Hibrit Sonucu

3.2.7 Çalışmada yapılan iyileştirmeler

Yorumlar ön işleme sürecinden geçirdikten sonra WEKA ile analiz yapılmıştır. Daha önce de belirtildiği gibi WEKA ile analiz yapabilmek için dosyaların formatının .arff olması gerekmektedir.

NaïveBayes ve SMO algoritmalarının olumlu taraflarının birleştirilmesinin sonucu başarı çıksa da yorumları metin olarak değil cümle bazlı okumanın daha sağlıklı olacağı gözlemlenmiştir. Bir metinde yer alan cümleler tek tek parçalanmıştır. Parçalanan cümleler sırasıyla veritabanına aktarılmıştır. Daha sonra cümleler için veritabanında 15 tane kategori oluşturulmuştur. Kategoriler şunlardır; Deniz, Havuz, Temizlik, Konum, Ulaşım, Personel, Spa, Yeme, Animasyon, Çocuk, Otopark, Fitness, Toplantı.

Bu kategoriler kolayca değerlendirilmesi için Asp .Net ile arayüz oluşturulmuştur. Kategorilenmemiş cümleler arayüze çekilmiş olup değerlendirme yapılırken bahsi geçen kelimeler kategorilerde işaretlenmiştir. Oluşturulan veribatanı Şekil 3.11'deki gibi ve arayüz Şekil 3.12'deki gibidir.



Şekil 3.11 Oluşturulan Arayüz

Id	Cumle	Yemek	Havuz	Personel	Deniz	Temizlik	Konum	Fitness	Anim	Cocuklar	Engelliler	Ulasim	Otopark	Spa	Toplanti	Genel
287	Varış zamanı kontrol etmek için bize bağlanış ve p...	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
288	Bizim çok sayıda valiz tüm odalar getirildi ya da kal...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
289	Tüm otel modern, bizim evin göre daha temiz ve ç...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
290	Personel çok titiz dikkatlerini, gerçekten çok; gaşrt...	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
291	Sahile ve İzmir Saat Kulesi'ni sadece 10 - 12 dakik...	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
292	Ayrıca, Pamukkale gitmek için planlıyorsanız veya ...	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
293	Harika ve çok ekonomik seyahat edenler için bütü...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
294	İzmir'e bu fazla için yeterli değil ama bir gece için g...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
295	Kahvaltı oldukça sıradan ve konumu iyi, tren istasy...	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
296	Tamamen yenilenmiş (Ben kaldı burada önce renos...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
297	Bu yıl iki ziyaret ve eğin ve ben her ikisi sut. Agora...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Şekil 3.12 Veritabanındaki Kategoriler

Okunup kategorileri değerlendirilen cümleler Şekil 3.12'deki gibi veritabanına kaydedilmiştir. Bu değerlendirilen yorumlar daha sonra .arff dosyasına çevrilerek her kategori sırasıyla Weka'da test edilmiştir. Test edilen kategorilerden bazıları FN değeri düşük çıkmıştır. Aşağıdaki örnekte deniz kategorisinde olduğu gibi. Bu gibi durumlarda Cost Sensitive Classifier kullanımının daha iyi sonuc verdiği gözlemlenmiştir.

```
=== Confusion Matrix ===

   a    b   <-- classified as
 134  123 |    a = T
  69 6674 |    b = F
```

Şekil 3.13 Deniz Kategorisi Test Sonucu

Precision ve Recall sonucuna bakmak gerekirse:

Precision= 134/(134+69)=0.660
Recall= 134/(134+123)=0.521
F-measure=2*0.660*0.521/(0.660+0.521)=0.583

Amacımız Precision ve recall değerlerini iyileştirmek. Dolayısıyla ikisinin harmonik ortası olan f-measure da iyileşmeli. Cost Sensitive Classifier sınıflandırması bize dengesiz yapıdan kaynaklanan düşük precision ve f-measure sorunun çözümüdür. Bu sınıflandırma biraz olsun dengesiz kısma ağırlık vererek dengeyi sağlamaya çalışmaktadır.

```
=== Confusion Matrix ===

   a    b   <-- classified as
 160   97 |    a = T
  84 6659 |    b = F
```

Şekil 3.14 Cost Sensitive Classifier Kullanılarak Deniz Kategorisinin Test Sonucu

Precision ve Recall sonucuna bakmak gerekirse:

Precision= 160/(160+84)=0.656
Recall= 160/(160+97)=0.624
F-measure=2*0.656*0.624/(0.656+0.624)=0.6396

Görüldüğü üzere Precision ve Recall değeri eski hesaplamaya göre daha iyi sonuç verdi. Bu bağlı olarak Cost Sensitive Classifier negatif olarak FN değeri artmıştır. Bunun sebebi FN'ye ağırlık verilmesinden dolayıdır.

Bu sınıflandırma 3 kategori dışında iyileşme sağlamamıştır. İyileşme sağlanan kategoriler (Deniz, Havuz, Temizlik). Bu kategoriler dışındakiler özel yöntem kullanılarak ve çıktının orijinal halini kullanarak devam edilmiştir. Orijinal halini kullanacağımız kategoriler (Konum, Ulaşım, Personel, Spa, Yeme).

Özel yöntem olarak kullacağımız kategoriler (Animasyon, Çocuk, Otopark, Fitness, Toplantı). Bu kategoriler wekadan aldığımız sonuçlar içerisindeki o kategoriye ait kelimelerin ağırlığı belirgin olan kelimeler alınarak o sınıfa atanmıştır. Böylelikle elimizde olan veri azlığından kaynaklanan sorundan böylelikle iyileşme gerçekleşmiştir.

3.3 JSON

JSON nedir?

Json, Javascript uygulamaları için oluşturulmuş bir veri formatıdır. Javascript Object Notation'ın kısaltmasıdır. Json'ın çıkış amacı veri transferlerinde verilerin XML'den daha az yer kaplamasını sağlamaktır. Şu an sadece Javascript uygulamalarında değil, yazılım geliştirmede kullanılan birçok teknolojide Json formatındaki veriler tercih edilmektedir.

Json Veri Formatı

Json türündeki veriler iki parçadan oluşur: key (anahtar) ve value (değer). Anahtar'da nesnenin hangi özelliğinin olduğu (koddaki değişken ismi gibi düşünülebilir) tanımlanırken değerde ise anahtar özelliğinin değeri (değişkenin değeri) tanımlanır. Nesnelerdeki anahtar ve değerler string türünde tanımlanır. Aşağıda basit bir json nesnesi örneği bulunmaktadır.

Şekil 3.15 JSON Formatı

Yukarıdaki Şekil 3.15'de çalıştığımız JSON dosyasının bir kısmı bulunmakta.8 tane anahtar ve 8 tane değer var: "Şehir" anahtarının değeri "İstanbul" olarak tanımlanmıştır. 2 key value'ya sahip Json nesnesi GoogleMap: "Latitude" anahtarının değeri: "41.009285", "Longitude" anahtarının değeri "28.975676" olarak tanımlanmıştır.

3.3.1 JSON yapısına uygun sınıfların oluşturulması

```
class OldJsonDto
    {
        public string OtelName { get; set; }
        public string Şehir { get; set; }
        public string Şehir_Sıralaması { get; set; }
        public List<Yorum> Yorumlar { get; set; } //Yorum tipinde yorumlar
listesi
        public string Link { get; set; }
        public string Adres { get; set; }
        public Map GoogleMap { get; set; }
        public List<Adrs> GoogleAddress { get; set; }
        public class Map
        {
            public string Latitude { get; set; }
            public string Longitude { get; set; }
        }
        public class Adrs
        {
            public string long_name { get; set; }
            public string short_name { get; set; }
            public List<string> types { get; set; }
        }
        public class Yorum
        {
            public string Yorumu { get; set; } //Yorumların yorumu

        }
        public string Oda_Sayısı;
        public string Otel_Sınıfı;
        public string Fiyat_Aralığı;
        public List<string> Otel_Tarzı;

    }
```

JSON dosyamızdaki anahtarlarla aynı isimde uygun değişkenlerle OldJsonDto adında bir sınıf oluşturulmuştur.

```
class NewJsonDto
    {
        public string OtelName { get; set; }
        public string Şehir { get; set; }
        public string Şehir_Sıralaması { get; set; }
        public List<Yorum> Yorumlar { get; set; } //Yorum tipinde yorumlar
listesi
        public string Link { get; set; }
        public string Adres { get; set; }
        public OldJsonDto.Map GoogleMap { get; set; }
```

```
        public List<OldJsonDto.Adrs> GoogleAddress { get; set; }

        public class Yorum
        {
            public string Yorumu { get; set; } //Yorumların yorumu
            public Dictionary<string, double> KategoriOlumluluk { get; set; }
            public double GenelOlumluluk { get; set; }

        }
        public string Oda_Sayısı;
        public string Otel_Sınıfı;
        public string Fiyat_Aralığı;
        public List<string> Otel_Tarzı;
        public Dictionary<string, double> KategoriOlumluluk_Otel { get; set; }
        public double GenelOlumluluk_Otel { get; set; }
    }
```

JSON nesnemiz için bizden istenen alanlar eklenerek NewJsonDto adında bir sınıf oluşturulmuştur.

3.3.2 JSON dosyasının okunması

```
using (StreamReader _StreamReader = new
StreamReader(@"D:\Tez\yorum_duzenlenmis.json"))//dosyamızı okuyoruz
            {
                var jsonData = _StreamReader.ReadToEnd();

                var jsonYorum =
JsonConvert.DeserializeObject<List<OldJsonDto>>(jsonData);
                foreach (var item in jsonYorum)
                {
                    if(item == null)
                    {
                        Console.WriteLine("null değer");
                        continue;
                    }
                    OldJsonDto otel = item;
                    liste.Add(otel);
                }
            }
```

Oluşturduğumuz OldJsonDto sınıfı türünde bir liste ile Json verisini liste nesnesine dönüştürdük.

3.3.3 Cümlenin olumlu olup olmadığının belirlenmesi

```
class IsCommentNegativeorPositive
    {
        private static YrmAnalizEntities db = new YrmAnalizEntities();
        private static List<PNSMO> SmoKelime = db.PNSMO.OrderBy(y =>
y.Id).ToList();
```

```
        private static List<PNNaiveBayes> NaiveKelime =
db.PNNaiveBayes.OrderBy(y => y.Id).ToList();

        public static bool YorumAnaliz(string cumle)
        {
            cumle = WebUtility.HtmlDecode(RemoveHtml(cumle));
            bool SmoSonuc, NaiveSonuc;
            NaiveSonuc = NaiveBayes(cumle);
            SmoSonuc = SMO(cumle);
            if (SmoSonuc ==NaiveSonuc)
            {
                return SmoSonuc;
            }
            return false;
        }

        private static bool SMO(string cumle)
        {
            double sonuc = 0;
            var yrm = Regex.Replace(cumle, @"(\p{P})", " ");
            yrm = new String(yrm.Where(c => c != '-' && (c < '0' || c >
'9')).ToArray());
            char ayrac = ' ';
            string[] kelimeler = yrm.Split(ayrac);
            foreach (var kelime in kelimeler)
            {
                foreach (var KelimeL in SmoKelime)
                {
                    if (kelime == KelimeL.Kelime)
                    {
                        sonuc += KelimeL.Agirlik;
                    }
                }
            }
            sonuc -= 0.5258;
            if (sonuc < 0)
            {
                sonuc = 0;
                return true;
            }
            sonuc = 0;
            return false;
        }

        private static bool NaiveBayes(string cumle)
        {
            double Tsonuc = 1.0, Fsonuc = 1.0;
            double T = 0.7796960486322189;
            double F = 0.22030395136778116;

            var yrm = Regex.Replace(cumle, @"(\p{P})", " ");
            yrm = new String(yrm.Where(c => c != '-' && (c < '0' || c >
'9')).ToArray());
            char ayrac = ' ';
            string[] kelimeler = yrm.Split(ayrac);
            foreach (var kelime in kelimeler)
            {
                foreach (var Kelime in NaiveKelime)
                {
                    if (kelime == Kelime.Kelime)
                    {
```

```
                    Tsonuc *= Kelime.True;
                    Fsonuc *= Kelime.False;
                }
            }
        }
        Tsonuc *= T;
        Fsonuc *= F;

        if (Tsonuc > Fsonuc)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public static string RemoveHtml(string text)
    {
        return Regex.Replace(text, @"<(.|\n)*?>", string.Empty);
    }

}
```

Veritabanımızda bulunan PNSMO ve PNNaiveBayes tablolarını static bir liste nesnesine aktardık. Bu tablolarda algoritmalarımıza ait SMO için kelime ağırlıkları, Naive Bayes için True ve False ağırlıkları bulunmaktadır.

YorumAnaliz methodumuzda cümlemizi html taglarından temizleyerek SMO ve NaiveBayes methodlarına gönderdik ve bu methodlardan gelen boolean değeri bir değişkende tuttuk. Eğer sonuçlar aynıysa (örneğin ikisi de true döndüyse) sonucumuzda bu değer oluyor. Şayet birbirinden farklıysa yaptığımız çalışmalar neticesinde aldığımız en iyi sonuçlara göre cümlemizin olumsuz olduğunu kabul ediyoruz.

SMO ve NaiveBayes methodlarının her ikisindede cümlemizi noktalama işaretleri ve sayılardan temizledik. Temizlenmiş cümleri kelimelere bölerek algoritmanın gerektirdiği işlemleri uyguladık.

3.3.4 Cümlenin kategorilerinin belirlenmesi

```
public static YorumCikti Analiz(string yorum)
        {
            yorum = WebUtility.HtmlDecode(RemoveHtml(yorum));

            double OlumlulukYuzde;
            string[] cumleler = yorum.Split('.');


            foreach (var cumle in cumleler)
            {
                if (cumle == "")
                    continue;
```

```
                else
                {
                    Cumle++;
                    bool IsAnim = AnimasyonOZEL(cumle);
                    bool IsCocuk = CocuklarOZEL(cumle);
                    bool IsTemizlik = TemizlikSMO(cumle);
                    bool IsKonum = KonumSMO(cumle);
                    bool IsDeniz = DenizSMO(cumle);
                    bool IsFitness = FitnessSMO(cumle);
                    bool IsHavuz = HavuzSMO(cumle);
                    bool IsSpa = SpaSMO(cumle);
                    bool IsToplanti = ToplantiSMO(cumle);
                    bool IsOtopark = OtoparkSMO(cumle);
                    bool IsPersonel = PersonelSMO(cumle);
                    bool IsYemek = YemekSMO(cumle);
                    bool IsUlasim = UlasimSMO(cumle);

                    if (IsAnim == true)
                        AnimC++;
                    if (IsCocuk == true)
                        CocukC++;
                    if (IsTemizlik == true)
                        TemizC++;
                    if (IsKonum == true)
                        KonumC++;
                    if (IsDeniz == true)
                        DenizC++;
                    if (IsFitness == true)
                        FitnessC++;
                    if (IsHavuz == true)
                        HavuzC++;
                    if (IsSpa == true)
                        SpaC++;
                    if (IsToplanti == true)
                        ToplantiC++;
                    if (IsOtopark == true)
                        OtoC++;
                    if (IsPersonel == true)
                        PersonelC++;
                    if (IsYemek == true)
                        YemeC++;
                    if (IsUlasim == true)
                        UlasimC++;

                    bool IsPositive =
IsCommentNegativeorPositive.YorumAnaliz(cumle);
                    if (IsPositive == true)
                    {
                        Olumlu++;
                        if (IsAnim == true)
                            AnimO++;
                        if (IsCocuk == true)
                            CocukO++;
                        if (IsTemizlik == true)
                            TemizO++;
                        if (IsKonum == true)
                            KonumO++;
                        if (IsDeniz == true)
                            DenizO++;
                        if (IsFitness == true)
                            FitnessO++;
```

```
                        if (IsHavuz == true)
                            HavuzO++;
                        if (IsSpa == true)
                            SpaO++;
                        if (IsToplanti == true)
                            ToplantiO++;
                        if (IsOtopark == true)
                            OtoO++;
                        if (IsPersonel == true)
                            PersonelO++;
                        if (IsYemek == true)
                            YemeO++;
                        if (IsUlasim == true)
                            UlasimO++;

                    }

                }
            }
            YorumCikti yorumCikti = new YorumCikti();
            OlumlulukYuzde = (double)Olumlu / (double)Cumle;
            Dictionary<string, double> kategoriler = new Dictionary<string,
double>();
            yorumCikti.GenelOlumluluk = OlumlulukYuzde;
            if (AnimC!=0)
                kategoriler.Add("Animasyon", (double)AnimO / (double)AnimC);
            if (CocukC != 0)
                kategoriler.Add("Çocuk", (double)CocukO / (double)CocukC);
            if (TemizC != 0)
                kategoriler.Add("Temizlik", (double)TemizO / (double)TemizC);
            if (KonumC != 0)
                kategoriler.Add("Konum", (double)KonumO / (double)KonumC);
            if (DenizC != 0)
                kategoriler.Add("Deniz", (double)DenizO / (double)DenizC);
            if (FitnessC != 0)
                kategoriler.Add("Fitness", (double)FitnessO /
(double)FitnessC);
            if (HavuzC != 0)
                kategoriler.Add("Havuz", (double)HavuzO / (double)HavuzC);
            if (SpaC != 0)
                kategoriler.Add("Spa", (double)SpaO / (double)SpaC);
            if (ToplantiC != 0)
                kategoriler.Add("Toplantı", (double)ToplantiO /
(double)ToplantiC);
            if (OtoC != 0)
                kategoriler.Add("Otopark", (double)OtoO / (double)OtoC);
            if (PersonelC != 0)
                kategoriler.Add("Personel", (double)PersonelO /
(double)PersonelC);
            if (YemeC != 0)
                kategoriler.Add("Yemek", (double)YemeO / (double)YemeC);
            if (UlasimC != 0)
                kategoriler.Add("Ulaşım", (double)UlasimO / (double)UlasimC);

            yorumCikti.KategoriOlumluluk = kategoriler;
            AnimC = 0;CocukC = 0;TemizC = 0;
            KonumC = 0;
            DenizC = 0;
            FitnessC = 0;
            HavuzC = 0;
            SpaC = 0;
```

```
        ToplantiC = 0;
        OtoC = 0;
        PersonelC = 0;
        YemeC = 0;
        UlasimC = 0;
        AnimO = 0;
        CocukO = 0;
        TemizO = 0;
        KonumO = 0;
        DenizO = 0;
        FitnessO = 0;
        HavuzO = 0;
        SpaO = 0;
        ToplantiO = 0;
        OtoO = 0;
        PersonelO = 0;
        YemeO = 0;
        UlasimO = 0;
        Cumle = 0;
        Olumlu = 0;

        return yorumCikti;
    }
```

Analiz methodumuzun amacı methoda gelen yorumun genel olumluluk yüzdesi ve cümleleri hangi kategorilere ait ise o kategorileri ve olumluluk yüzdelerini geri döndürmektir.

BÖLÜM 4

Sosyal ağların ortaya çıkmasıyla yaşanan iletişim devriminde bireylerin etkileşimlerini kullanarak bilgi çıkarımı yapmak ve bu bilgilerden sonuçlar elde etmek, son yıllarda üzerinde çalışılan popüler konulardır. Yapılan bu çalışmaların temel konularından birisi de veri analizidir. Veri analizi çalışması metinlerin duygusal açıdan sınıflandırılması amacıyla yapılır. Veri analizi çalışmaları genel olarak sözlük tabanlı ve makine öğrenmesi yöntemleri yaklaşımları olarak ikiye ayrılır. Sözlük tabanlı yöntemlerde önceden oluşturulmuş duygu ağırlığı taşıyan terim sözlükleri kullanılır. Makine öğrenmesi yöntemlerinde de veri madenciliğinde kullanılan sınıflandırma algoritmaları kullanılır.

Bu tez çalışması bir veri analizi çalışmasıdır. Çalışmada kullanılan veri kaynağı TripAdvisor sitesinde paylaşılan otel yorumları metinleridir. Veri analizi çalışması yapılırken her iki yaklaşım da kullanılmıştır. Popüler bir otel rezarvasyon sitesi olan TripAdvisor, toplanan otel yorumları verileri belirli filtreleme ve ön işlemelerden

geçtikten sonra veri analizi yöntemleriyle otel yorumlarının içerdiği düşüncelerin durumu belirlenmiştir.

4.1. SONUÇLAR

Türkiye'deki eğitim kavramına ait verileri analiz ederek kişilerin düşencelerinin olumlu veya olumsuz olduğunu öğrenmek için yapılan bu çalışmada çeşitli veri analizi yöntemleri kullanılmıştır. Kullanılan bu yöntemlerin içinde makine öğrenmesi yöntemlerinden biri olan SMO algoritması ve NaiveBayes algoritması ile elde edilen sonuçlar uygulanan diğer algoritmalardan daha yüksek doğruluk oranına sahip olduğundan dolayı tercih edilmiştir. Daha düşük başarı oranının elde edildiği sınıflandırma algoritmaları içerdikleri işlemsel karmaşıklık nedeniyle ve doğruluk oranının daha düşük olmasından dolayı kullanılmamıştır.

SMO algoritması ve NaiveBayes algoritması ile elde edilen doğruluk yüzdesi, literatürdeki diğer çalışmalarla karşılaştırıldığında başarılı düzeydedir. Bu algoritmanın uygulanması ile elde edilen doğruluk oranı %70 ile %90 arasında değiştiği gözlemlenmiştir. Farklı veriler üzerinde alınan sonuçlar karşılaştırıldığında ortalama olarak %70 ve üzerinde bir doğruluk oranı elde edilmiştir. Otel yorumları üzerinde yapılan bu çalışmada, literatürdeki çalışmalar ile benzer seviyede başarı oranlarının elde edildiği görülmüştür. %70'in üzerinde elde edilen doğruluk oranları veri analizi açısından tatmin edici düzeydedir. Bu çalışma Türkiye bazında eğitim kelimesine bağlı olarak veri analizi çalışması yapılması ve sonuçların yine liste üzerinde gösterilmesi açısından literatürde bulunan diğer arayüz üzerinden yapılan veri analizi çalışmalarından farklılık göstermektedir.

4.2. ÖNERİLER

Bu tez çalışmasında makine öğrenmesi yöntemlerinden duygu analizinde başlıca kullanılan SMO algoritması ve NaiveBayes algoritması kullanılmıştır. Diğer kullanılmayan makine öğrenmesi yöntemlerinin de dâhil edilmesiyle çalışma daha kapsamlı hale getirilebilir. Yapılan çalışmada kişilerin düşüncelerinin olumlu veya olumsuz olduğunun belirlenmesi için anlam ifade eden toplam 300 anahtar kelime kullanılmıştır. Bu kelimelerin sayısının artması yapılan veri analizinin hassasiyetini artıracaktır. Bununla birlikte elde edilen verinin arttırılması, daha ayrıntılı ve dikkatli ön işleme aşamalarına tabi tutularak yapılması yaşanabilecek anlam içeren kelime kayıpların önüne geçerek sonuçların doğruluk oranını arttıracaktır.

Bu çalışmada sadece Türkçe dilinde yazılmış olan otel yorumlarının üzerinde Eğitim anahtar kelimesi ile veri analizi yapılmıştır. Diğer dillere ait anahtar kelime veya kelimelerin belirlenmesi ve o dilde yazılmış olan metinlerin elde edilmesi ve işlenmesiyle diğer dillere de uygulanarak sonuç alınabilir.

KAYNAKLAR

1. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp
2. http://bilgisayarkavramlari.sadievrenseker.com/2009/06/01/weka/
3. http://bilgisayarkavramlari.sadievrenseker.com/2009/06/02/weka/
4. Sadi Evren Seker, "Event Ordering for Turkish Natural Language Texts," in CSW-2010 1 st Computer Science Student Workshop, Istanbul, 2010, pp. 26-29.
5. http://bilgisayarkavramlari.sadievrenseker.com/2011/09/19/weka-ile-svm/
6. http://bilgisayarkavramlari.sadievrenseker.com/2013/02/08/naif-bayes-siniflandiricisi-naive-bayes/
7. Sadi Evren Seker, C Mert, K Al-Naami, N Ozalp, and U Ayan, "Time Series Analysis on Stock Market for Text Mining Correlation of Economy News," International Journal of Social Sciences and Humanity Studies, vol. 6, no. 1, pp. 69 - 91, Mar. 2014.

8. http://dergipark.gov.tr/download/article-file/317689

9. http://web.firat.edu.tr/mbaykara/ubmk3.pdf

10. B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, 2002.

11. http://www.openaccess.hacettepe.edu.tr:8080/xmlui/bitstream/handle/11655/2626/100811e6-ace7-4ded-972e-06577a8fd46f.pdf?sequence=1

12. http://ybsansiklopedi.com/wp-content/uploads/2016/09/duygu_analizi.pdf

13. http://www.openaccess.hacettepe.edu.tr:8080/xmlui/bitstream/handle/11655/2606/947ed694-e80f-4774-8eed-2185410b0c67.pdf?sequence=1&isAllowed=y

14. BOYNUKALIN, Z., "EMOTION ANALYSIS OF TURKISH TEXTS BY USING MACHINE LEARNING METHODS", *MIDDLE EAST TECHNICAL UNIVERSITY*.

15. Y. H. Hu, K. Chen, and P. J. Lee, "The effect of user-controllable filters on the prediction of online hotel reviews," Inf. Manag., vol. 54, no. 6, pp. 728–744, 2017.

16. I. Habernal, T. Ptácek, and J. Steinberger, "Reprint of 'supervised sentiment analysis in Czech social media,'" Inf. Process. Manag., vol. 51, no. 4, pp. 532–546, 2015.

17. A. Tripathy, A. Agrawal, and S. Kumar, "Classification of Sentimental Reviews Using Machine Learning Techniques," vol. 0, no. November, pp. 117–126, 2014.

18. ConnotationWordNet: Learning Connotation over the Word+Sense Network www.aclweb.org/anthology/P/P14/P14-1145.xhtml, 2014.