



T.C DOĐUŐ ÜNİVERSİTESİ
LİSANSÜSTÜ EĐİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĐİ YÜKSEK LİSANS PROGRAMI

E-TİCARET ÜRÜNLERİNDEKİ TÜRKÇE KULLANICI YORUMLARININ
OYUNLAŐTIRMA, NLP VE MAKİNE ÖĐRENMESİ TEKNİKLERİ İLE
SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

RİFAT MERT DOLAR

201695003

DANIŐMAN:

Dr. Öğr. Üyesi YASEMİN KARAGÜL

İstanbul, 2021



YÜKSEK LİSANS TEZ SINAV TUTANAĞI

Doküman No	FR.1.26
Yürürlük Tarihi	1.11.2017
Revizyon Tarihi	4.12.2020
Revizyon No	2
Sayfa	1 / 1

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Tarih : 23/02/2021

Anabilim/Anasanat Dalı : Bilgisayar Mühendisliği.....

Öğrencinin Adı Soyadı : Rıfat Mert DOLAR.....

Öğrenci No : 201695003.....

Tez Danışmanının Adı Soyadı : Dr. Öğr. Üyesi Yasemin KARAGÜL.....

İkinci Tez Danışmanının Adı Soyadı :.....

Tezin Başlığı : E-Ticaret Ürünlerindeki Türkçe Kullanıcı Yorumlarının
Oyunlaştırma, NLP ve Makine Öğrenmesi Teknikleri ile
Sınıflandırılması

Doğuş Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin 32.Maddesi uyarınca
yapılan değerlendirmeler sonunda;

☒ tezin kabul
edilmesine

☐ tezde düzeltme
verilmesine

☐ tezin
reddedilmesine

oy birliği / oy çokluğu ile karar verilmiştir. Gereği için arz olunur.

Danışman Üye
Dr. Öğr. Üyesi Ya*** K

Üye
Dr. Öğr. Üyesi M*** Z*** G***

Üye
Dr. Öğr. Üyesi O*** K***

Üye

Üye

Anabilim/Anasanat Dalı Başkanı Onayı: Dr. Öğr. Üyesi Yasemin KARAGÜL

Doğuş Üniversitesi ("Üniversite") olarak kişisel verilerinizin güvenliği hakkında azami hassasiyet göstermekteyiz. Üniversite olarak, Üniversite ile ilişkili tüm şahıslara ait her türlü kişisel verinin 6698 sayılı Kişisel Verilerin Korunması Kanunu ("KVK K.")'na uygun olarak işlenmesine ve gizliliğinin sağlanmasına büyük önem vermekteyiz. Anayasa m.20/3 ile tanımlanan "kişisel verilerinizin korunmasını isteme" hakkınızın bilincinde olarak, KVK K. kapsamında tanımlı "Veri Sorumlusu" sıfatıyla, kişisel verilerinizi mevzuata uygun şekilde işlemekteyiz. Daha geniş açıklama için Doğuş Üniversitesi <https://www.dogus.edu.tr/hakkimizda/kisisel-verilerin-korunmasi>

As Doğuş University ("University"), we show maximum sensitivity about the security of your personal data. As a university, we attach great importance to the processing and confidentiality of all personal data of all individuals associated with the University in accordance with the Law No. 6698 on Protection of Personal Data ("KVK K."). Being aware of your right to "ask for the protection of your personal data" recognized by the constitution article 20/3, as "Data Controller" defined within the scope of KVK K., we process your personal data in accordance with the legislation. For a broader explanation you can visit <https://www.dogus.edu.tr/hakkimizda/kisisel-verilerin-korunmasi>

hakkımızda: about us kişisel-verilerin-korunmasi: protection of personal data

(Form No: FR-001, Revizyon Tarihi:30.09.2016, Revizyon No:00)

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “E-Ticaret Ürünlerindeki Türkçe Kullanıcı Yorumlarının Oyunlaştırma, Nlp ve Makine Öğrenmesi Teknikleri ile Sınıflandırılması” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

08.03.2021

Rıfat Mert DOLAR



ÖNSÖZ

Bu tez çalışmasında, e-ticaret platformlarındaki ürünlere yapılan kullanıcı yorumlarının oyunlaştırma, NLP ve makine öğrenmesi teknikleri ile sınıflandırılması çalışması ve değerlendirmeleri yapılmıştır. Çalışmada bana, yeni düşünme yolları ve araştırma metodolojilerini öğreten, ilerleyemediğim durumlarda yardımını esirgemeyen tez danışmanım Dr. Öğr. Üyesi Yasemin Karagül'e sonsuz teşekkürlerimi sunarım.

Sınıflandırma algoritmalarında kullandığım verileri puanlama ve sınıflandırma konusunda bana yardımcı olan K*** İ** K** ; E** İ** arkadaşlarıma teşekkürü borç bilirim. Yüksek Lisans yapmam konusunda bana destek olan ve her zaman motivasyonumda önemli rol alan eşim A*** 'a sonsuz sevgimi ve teşekkürlerimi sunarım.

Çalışmamın, bu konu üzerine araştırma yapmak isteyenlere yol göstermesini dilerim.

Ocak, 2021

Rıfat Mert DOLAR

ÖZET

Türkiye’de ve dünyada internet alışverişi, günden güne büyüyen, önemi giderek artan, dev bir sektör halini almış durumdadır. Özellikle bu tezin yazıldığı dönem itibari ile pandemi sürecinin getirdiği global kısıtlamalarla beraber altın çağını yaşayan internet alışverişinin önemi bir kez daha görülmüştür. Zaman tasarrufu, evden çıkmadan alışveriş yapmanın rahatlığı, yorulmadan ayağımıza gelen hizmet, ürün çeşitliliği, ödeme kolaylığı, ürünler ile ilgili yorumları görebilme, ürün karşılaştırmaları ve teslimat kolaylığı tüketicileri internet alışverişine çeken önemli unsurların başında gelmektedir. Bu unsurlardan bazıları tüketicilerin ürün tercihlerini ciddi oranda etkilemektedir. Bunlardan biri de ürünlere yapılmış olan tüketici yorumlarıdır. Bir ürünü daha önce almış ve deneyimlemiş tüketicilerin yorumları, ürünü alacak olan tüketicilerin tercihlerini etkilemektedir.

Bu tezin başlıca amacı, Türkiye’deki internet alışverişi sitelerindeki satılan ürünlere yapılan yorumları sınıflandırma algoritmaları, oyunlaştırma ve doğal dil işleme (NLP) teknikleri kullanılarak değerlendirilmesi amacıyla gelecek çalışmalara ışık tutmaktır. Toplamda 27 teknoloji ürünü ve 11236 kullanıcı yorumu üzerinde oyunlaştırma tekniği kullanılmıştır. Mobil uygulama oyunu üzerinden yapılan bir yarışma ile veri tabanında saklanan veriyi, rastgele bir şekilde uzmanlara göstererek, manuel etiketlemeler sonucu kategorize edilmiş eğitim verisi, doğal dil işleme (Zemberek, gürültü azaltma, belirteç oluşturma, normalleştirme), makine öğrenmesi ve sınıflandırma teknikleri (Fasttext, cbow, skipgram) kullanılarak 5 farklı duyguya (Çok olumlu, olumlu, nötr, olumsuz, çok olumsuz) kategorize edilmiştir. Kullanıcıların yazdıkları yorumlar ve verdikleri puanlardaki bazı tutarsızlıklar da grafiksel olarak gösterilmiş olup, gelecek çalışmalara yönelik olarak yorumlanmıştır.

Anahtar Kelimeler: Veri madenciliği, doğal dil işleme, makine öğrenmesi, sınıflandırma algoritmaları, Zemberek, Fasttext, ürün öneri sistemleri, hatalı kullanıcı yorumu tespiti

ABSTRACT

Internet shopping is growing, increasing importance, it has already been a huge sector in Turkey and the world. Especially since the writing of this thesis, the importance of internet shopping, which is experiencing its golden age with the global restrictions brought by the pandemic process, has been once again seen. Saving time, the convenience of shopping from without leaving home, the service that comes to our home without going any place, the variety of products, the convenience of payment, seeing the comments about the products, product comparisons and the ease of delivery are the main reasons that attract consumers to online shopping. Some of these reasons seriously affect the consumers' product preferences. One of these is consumer comments on products. The comments of consumers who have already bought and experienced a product before affect the preferences of the consumers who will buy the product.

The main objective of this thesis is to shed light on future studies while using classification algorithms and techniques, gamification and natural language processing (NLP) to commented products sold in Internet Shopping Sites in Turkey. Using gamification on daily usage of 27 technology products and 11236 users in total, by holding a competition over mobile application game, by showing the data stored in the database randomly to experts and categorized as a result of manual tagging, natural language processing (Zemberek, noise reduction, tokenization, normalization), machine learning and rendering style (Fasttext, cbow, skip-gram) are categorized into 5 different emotions (Very positive, positive, neutral, negative, very negative). Some inconsistencies in the comments written by the users and their scores were also shown graphically and interpreted for future studies.

Keywords: Data mining, natural language processing, machine learning, classification algorithms, Zemberek, Fasttext, product recommendation systems, incorrect user review detection

İÇİNDEKİLER

Sayfa No.

ÖNSÖZ	ii
ÖZET	v
ABSTRACT.....	vi
TABLO LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	x
SİMGE VE KISALTMA LİSTESİ	xi
1. GİRİŞ.....	1
1.1 Çalışmanın Amacı.....	1
1.2 Çalışmanın Kapsamı	2
2. LİTERATÜR TARAMASI.....	3
2.1 Veri Madenciliği	3
2.2 Makine Öğrenmesi	4
2.3 Metin Sınıflandırma	5
2.3.1 Word2Vec modeli	5
2.3.2 FastText modeli	6
2.4 Oyunlaştırma (Gamification)	7
2.4.1 Neden Oyunlaştırma	8
3. ARAÇLAR VE YÖNTEMLER.....	9
3.1 Kullanılan Araç ve Gereçler.....	10
3.1.1 Extensible markup language (XML)	10
3.1.2 JavaScript object notation (JSON)	11
3.1.3 Kotlin programlama dili	12
3.1.4 Java programlama dili	13
3.1.5 IntelliJ yazılım geliştirme ortamı	14
3.1.6 Android Studio yazılım geliştirme ortamı	16
3.1.7 Firestore bulut sistemi	18
3.1.8 SQLite veri tabanı.....	19
3.1.9 Postman	19
3.2 Veri Setinin Elde Edilmesi.....	19
3.3 Etiketleme Öncesi Yapılan Çalışmalar	21

3.3.1 Oyunlaştırma tekniğinin kullanımı	21
3.3.2 Etiketleme ortamının geliştirilmesi	23
3.4 Veri Setinin Etiketlenmesi	26
3.5 Etiketlenen Verinin NLP Teknikleri ile Düzenlenmesi	27
3.5.1 Noise Removal tekniğinin uygulanması	28
3.5.2 Tokenization tekniğinin uygulanması	28
3.5.3 Normalization tekniğinin uygulanması	28
3.5.4 Küçük harf dönüşümü ve noktalama işaretlerinin silinmesi	28
3.6 Eğitim ve Test Verilerinin Hazırlanması	28
4. SONUÇLAR VE BULGULAR.....	31
4.2 Kullanıcı Puanları ve Uzman Etiketlerini Arasındaki İlişki	34
4.3 FastText Test Sonuçları.....	36
5. TARTIŞMA VE GELECEK ÇALIŞMALARI.....	43
KAYNAKÇA.....	46

TABLO LİSTESİ

Sayfa No.

Tablo 3.1 Ürünler ve Yorum Sayıları	20
Tablo 3.2 Ham Veri Bilgileri	20
Tablo 3.3 Etiketlenen Yorum ve Kişi Dağılımı	27
Tablo 3.4 Etiketlenen Yorumların Sınıflara Dağılımı	27
Tablo 4.1 En Çok Kullanılan Kelimeler	31
Tablo 4.2 En Çok Kullanılan İkili Kelime Öbekleri.....	32
Tablo 4.3 En Çok Kullanılan İkili Kelime Öbekleri.....	33
Tablo 4.4 Uzman Etiketleri ve Kullanıcı Puanları.....	35
Tablo 4.5 Başarım Özeti	42

ŞEKİL LİSTESİ

Sayfa No.

Şekil 2.1 Veri Madenciliği Disiplinleri.....	3
Şekil 2.2 Veri Madenciliği İlgili Alanları	4
Şekil 2.3 Makine Öğrenme Türleri	5
Şekil 2.4 CBOW ve SKIPGRAM Çalışma Yöntemi.....	6
Şekil 2.5 CBOW ve SKIPGRAM Çalışma Yöntemi 2.....	7
Şekil 3.1 Araç ve Yöntemler Şeması Faz 1	9
Şekil 3.2 Araç ve Yöntemler Şeması Faz 2	10
Şekil 3.3 Genişletilebilir işaretleme dili (XML) örneği.....	11
Şekil 3.4 JavaScript object notation(Json) örneği.....	12
Şekil 3.5 Kotlin ile geliştirilen metod örneği.....	13
Şekil 3.6 Java ile geliştirilen metod örneği.....	14
Şekil 3.7 IntelliJ yazılım geliştirme aracı	15
Şekil 3.8 Android Studio yazılım geliştirme aracı.....	17
Şekil 3.9 Firestore geliştirici konsolu	18
Şekil 3.10 Kullanıcı tablosu.....	22
Şekil 3.11 Skor tablosu	23
Şekil 3.12 Mobil projede kullanılan kütüphaneler.....	24
Şekil 3.13 Mobil oyun uygulamasının arayüzü	26
Şekil 3.14 Zemberek-full.jar ve Eğitim Verisi Terminal Gösterimi.....	29
Şekil 4.1 En Çok Kullanılan Kelimeler	32
Şekil 4.2 En Çok Kullanılan İkili Kelime Öbeği	33
Şekil 4.3 En Çok Kullanılan Üçlü Kelime Öbeği.....	34
Şekil 4.4 Uzman Etiketleri ve Kullanıcı Puanlarını Karşılaştırma	35
Şekil 4.5 Uzman Etiketleri, Yorum ve Puan örneği	36
Şekil 4.6 Ham Veri Test sonucu	37
Şekil 4.7 NLP ve Veri Sayısı Artırımı ile Elde Edilen Test Sonucu	38
Şekil 4.8 Noktalama İşaretleri Kaldırılmış Test Sonucu	38
Şekil 4.9 Marka ve Reklam Kelimeleri Silinmiş Test Sonucu	39
Şekil 4.10 Tüm Harfler Küçük Harfken Elde Edilen Test Sonucu.....	40
Şekil 4.11 İkili Kelime Öbekleri ile Elde Edilen Test Sonucu	41
Şekil 4.12 Üçlü Kelime Öbekleri ile Elde Edilen Test Sonucu	42

SİMGE VE KISALTMA LİSTESİ

CBOW	: Continuous bag of words
E- Ticaret	: Elektronik Ticaret
JSON	: JavaScript Object Notation
JVM	: Java Virtual Machine
ML	: Machine Learning
NLP	: Natural Language Processing
SDK	: Software Development Kit
XML	: Extensible Markup Language



1. GİRİŞ

Dünyada internet erişimi elektrik, su, doğalgaz gibi temel ihtiyaçlardan biri haline gelmiştir. Bilgisayarlarımızdaki verilerin büyük çoğunluğunun bulut sistemlerinde saklanmaya başlanması ile yeryüzündeki bilgi miktarının her yıl katlanarak büyüdüğü günümüzde, büyük verinin sayısı da buna paralel olarak hızlı bir şekilde artmaktadır. İşlem yapabilme gücünün ve sunucu maliyetlerinin günden güne ucuzlamasıyla birlikte, verinin kolay bir şekilde saklanması da verinin maliyetlerini aşağıya çekmektedir (Vahaplar & İnceoğlu, 2001). Bulut sistemlerinin sağladığı bu avantajlarla günümüzde elektronik ticaret ve online alışveriş platformları da yaygınlaşmaya başlamıştır. Bu platformlarda birçok kullanıcı verileri elde edilmeye başlanmıştır. Bu veriler işlemde geçmemiş ham veriler olarak isimlendirilir. Ham verinin anlamlandırılması sayesinde elde edilen anlamlı bilgilerin değerlendirilmesi ile bu alanda dünyada birçok yeni şirketin ortaya çıkması, öneri sistemleri ve veri madenciliğinin ne denli önemli olduğunu göstermektedir. Bu alanda çalışma yapan Türkiye çıkışlı bazı küresel şirketler: Segmentify, Insider, Visilabs.

Dünyada gün geçtikçe değeri artan bir çalışma alanı olan veri madenciliği, depolanmış olan büyük miktardaki verinin veri tabanlarında çeşitli madencilik teknikleri ile aranmasıyla elde edilen sonuçlarla ileriye dönük tahmin yapılması olarak ifade edilebilir (Alpaydın, 2000). Veri madenciliğiyle sonuç elde edilirken, tek başına bir anlam ifade etmeyen veriler arasında anlamlı ilişkiler kurularak değerli olan bilgiye ulaşılmaya çalışılır. Veri madenciliğiyle geçmişte toplanılan veriler işlenerek geleceğe yönelik kullanılabilir veriler elde edilir (Savaş, Topaloğlu & Yılmaz, 2012).

Veri madenciliği tek başına bir çözüm oluşturmayabilir. Ancak çözüme bir adım daha yaklaşmak için verilecek kararlarda etkili olabilir. Problemin çözülmesinde gerekli bilgilerin elde edilmesinde kullanılan bir yöntemler havuzudur. İşlemler analiz edilirken veriler arasındaki ilişkileri bulmada veri madenciliğinden faydalanılır (Baykal, 2006).

1.1 Çalışmanın Amacı

Bu çalışmanın amacını destekleyen faktörlerden biri, veri madenciliği ve öneri sistemlerine odaklanmış birçok start-up ve şirketin, küresel ölçekte büyümeyi tercih etmesiyle beraber, Türkçe veri madenciliği çalışmalarının hızına oranla, internetteki Türkçe verilerin inanılmaz bir hızla büyümesidir.

Türkçe verilerin her geçen gün daha da büyümesiyle ihtiyaca dönüşmeye başlayan öneri sistemleri ve metin madenciliği gibi alanlarda daha önceden benzeri yapılmış olan birçok ürün öneri sistemi çalışmasına destek olmayı hedeflemekle beraber, ileride bu çalışmayı yapacak olan çalışmacılara daha farklı bakış açıları kazandırabilecek bir çalışma olan, e-ticaret ürünlerine yapılan kullanıcı yorumlarını, daha önceden keşfedilmiş olan modellerin harmanlanmasıyla oluşturulacak yeni bir model ile başarı oranını artırmaya yönelik çalışmalar yapılması amaçlanmıştır. Ölçümler sonrası elde edilen sonuçlar hakkında tartışma ve yorumlar yapılacaktır.

Çalışmanın Türkçe dilinde yapılmasının amacı ise, bu dilde oluşan verileri anlamlandırmaya çalışan çalışmaların az olması, çalışma yapmak isteyen çalışmacıların yeterli Türkçe kaynak ve kütüphane bulamaması gibi etmenlerle beraber bu alandaki çalışmaların Türkçe dilinde daha çok yol kat etmesi gerekliliğidir. Literatürde bu çalışmaya benzer çalışmalar yapılmış olup, genel itibarı ile dünyada ortak dil olarak kabul edilen İngilizce dili üzerine yoğunlaşmıştır. Bu çalışma Türkçe diline katkı sağlamak adına Türkçe metin verileri üzerinde yapılan çalışmalarla ilerleyip geleceğe ışık tutmayı amaçlamıştır.

1.2 Çalışmanın Kapsamı

Bu çalışma kapsamında, ham veri kaynağında, ürünlerin web site linkleri, ürünler hakkında yorumlar ve ürün değerlendirme puanları yer almaktadır. Ürün yorumları ve çalışma kapsamında etiketleme görevini üstlenmiş uzmanların etiket puanları üzerinden analizler yapıldıktan sonra işlenmiş veri üzerinde testler yapıp başarı oranlarının ölçülmesi de kapsam dahilindedir. Bu çalışma kapsamı tercih edilmeden önce yapılan araştırmalar sonucu daha önce yapılmış olan benzer çalışmalarda, insandan bağımsız yapılan çalışmalarda sonradan el ve göz yardımından faydalanılmıştır. Erdoğan (2019) çalışmasının bazı adımlarında bu yola başvurulduğu görülmüştür. Bu tez çalışmasında oluşturulacak olan modelin temelini insan sezgilerinden elde edilmiş sayısal veriler oluşturmaktadır.

Bu çalışmada, ilgili konu ile alakalı önceden yapılmış çalışmalar hakkında olan “Literatür Taraması” yapılmıştır. Kullanılan araçların ve uygulanan metotların yer aldığı “Araçlar & Yöntemler” konusu anlatılmıştır. Sonrasında, uygulanan yöntemler neticesinde elde edilen sonuçların yer aldığı “Sonuçlar & Bulgular” bölümüne yer verilmiştir. En son ise “Tartışma & Gelecek Çalışmaları” ele alınmıştır.

2. LİTERATÜR TARAMASI

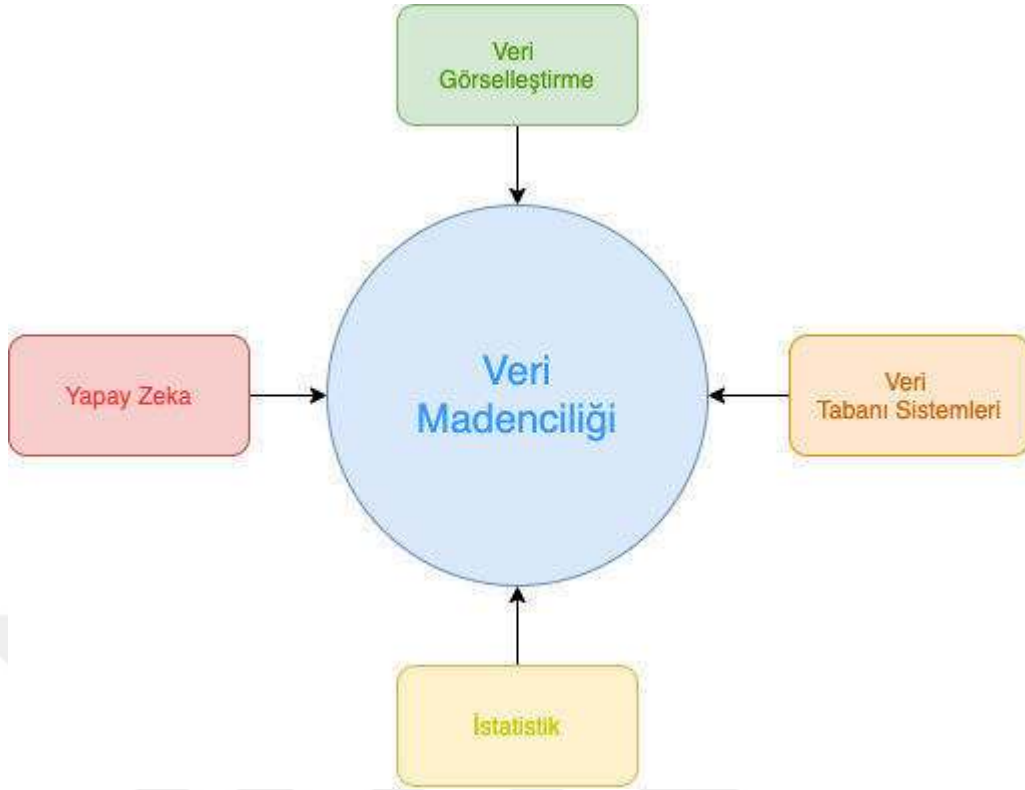
2.1 Veri Madenciliği

Veri Madenciliği, büyük veriler üzerinde uygulanan ve birçok adımdan oluşan işlemler sonucu, anlamlı bilgilere ulaşmak için kullanılan bir yöntemdir. Veri madenciliğinin en önemli amaçlarından biri, veriler içerisinde bulunan bağlantılar üzerinden ilerleyerek yeni bilgiler edinmeyi amaçlamaktadır. Mantığa dayalı bağlantılar elde edebilmek için, çok sayıda verinin üzerinde işlem yapılmalıdır. Veri madenciliğinde birbirinden farklı birçok yöntem bulunmaktadır ve bu yöntemler oldukça büyük verilerin üzerinde karmaşık işlemler yapabilen güçlü yöntemlerdir. Günümüzde birçok veri madenciliği yöntemi bulunmaktadır, bu yöntem ve teknikler karıştırılarak farklı kombinasyonlar meydana getirilerek, yeni yöntemler meydana getirmektedir. Ayrıca çok büyük verilerin kendi aralarındaki bağlantıları inceleyerek, aralarında ne tür bir ilişki olduğu hakkında bilgi elde etmeye yarayan bir yöntemdir (Albayrak, 2017; Kalikov, 2006). Veri Madenciliği disiplinleri şekil 2.1’de gösterilmiştir.



Şekil 2.1 Veri Madenciliği Disiplinleri

İstatistiksel yöntemler dizisi olarak da veri madenciliğini tanımlayabiliriz. Veri madenciliği, istatistikten farklı özellikler de bulundurmaktadır. Pratik bir şekilde mantık kuralları çerçevesinde ölçümlenebilir modellerin elde edilmesidir (Baykasoğlu, 2005). Şekil 2.2’de gösterildiği gibi veri madenciliği kullanım alanları, istatistik, yapay zekâ, veri tabanları ve veri görselleştirme gibi alanlarla bağlantılıdır.

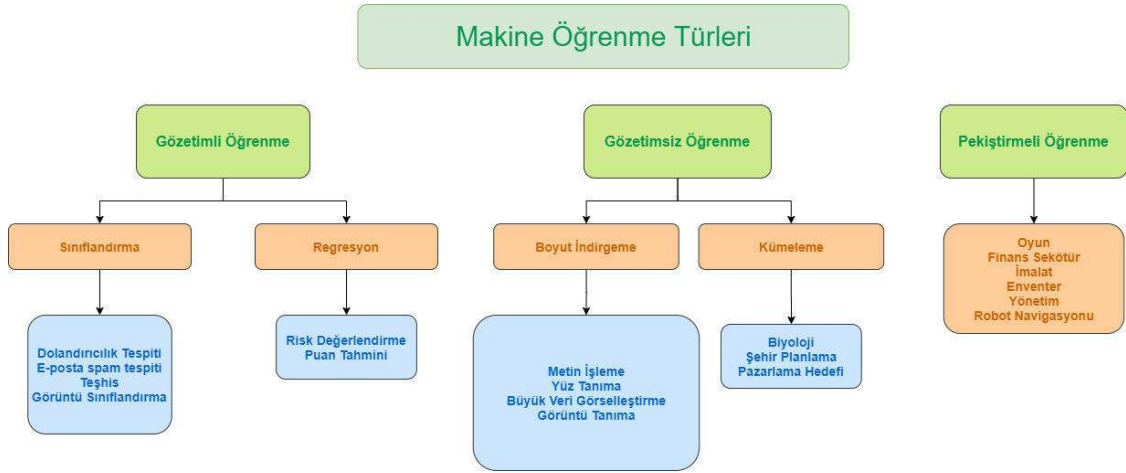


Şekil 2.2 Veri Madenciliği İlgili Alanları

2.2 Makine Öğrenmesi

Büyük miktardaki verilerin işlenmesi ve analizinin yapılmasında Makine öğrenmesi yöntemler kullanılmaktadır. Bu yöntemlerin ortaya çıkma nedeni, çok büyük verilerin insan eli ve gözü yordamıyla yapılmasının çok zor, hatta neredeyse imkânsız olmasıdır. Makine öğrenmesi yöntemleri, ham veriyi çeşitli işlemlerden geçirerek, yeni ve en verimli modeli bulmayı amaçlar (Diri, 2014).

Makine öğrenmesinde kullanılan bazı metotlar şunlardır; sınıflandırma, kümeleme, Regresyon (Eğri Uydurma), Birliktelik Kuralları vb. Şekil 2.3'te bu alanlar görülmektedir.



Şekil 2.3 Makine Öğrenme Türleri

2.3 Metin Sınıflandırma

Metin sınıflandırması, Doğal Dil İşlemede (NLP) önemli bir problemdir. Gerçek dünya kullanımları arasında spam filtreleme veya e-posta kategorizasyonu bulunur. Arama ve sıralama gibi daha karmaşık sistemlerde temel bir bileşendir. Son zamanlarda, sinir ağlarına dayalı derin öğrenme teknikleri çeşitli NLP uygulamalarında son derece başarılı sonuçlar elde etmiştir (Joulin vd., 2017). Dünya üzerindeki en ünlü modellerden bazıları Word2Vec, FastText, Glove.

2.3.1 Word2Vec modeli

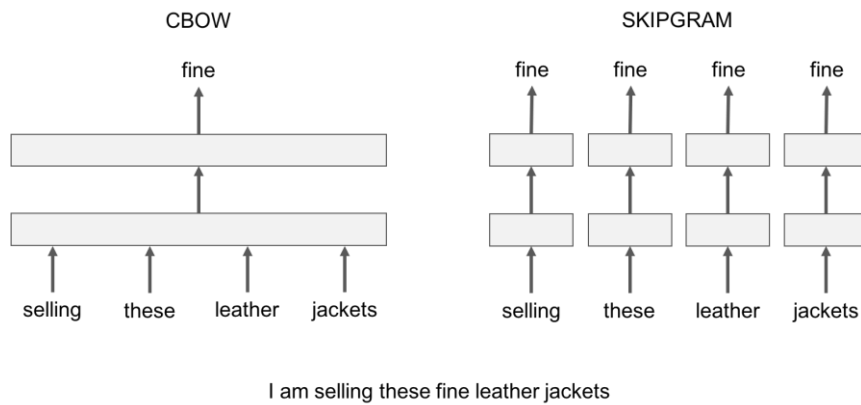
Word2vec, temeli yapay sinir ağlarına dayanan doğal dil işleme için kullanılan tekniklerden biridir. Temelde 3 farklı katmandan oluşur. Bunlar sırasıyla girdi, çıktı ve gizli katman olarak isimlendirilir. Bir kelimenin diğer kelimeler ile olan ilişkilerinden yaptığı çıkarımlar ile kelime öbeklerini öğrenmeyi amaçlar. Skipgram ve continuous-bag-of words mimarilerinden gücünü almaktadır. CBOW adından da anlaşılacağı üzere kelime çantaları oluşturarak hedefteki kelimeyi tanımlama ile görevlendirilmiştir. Skipgram’da, hedeflenen kelimeyi girdi olarak kullanır ve hedef kelimenin etrafındaki kelimeleri tahmin etmeye çalışır. CBOW tekniğinin en büyük özelliklerinden biri, verinin az olmasına karşın çalışmasını büyük oranda başarıyla tamamlamasıdır. Bunun aksine Skipgram’ın doğru sonuçlar verebilmesi için büyük veri setlerine ihtiyacı vardır (Onan, 2020). Bu tekniğin, verileri eğitme hızı, FastText’ten daha iyi olsa dahi, nadir kelimeleri ayırt etme konusunda daha zayıf olmasından dolayı bu tezde FastText tercih edilmiştir.

2.3.2 FastText modeli

FastText, Facebook AI Research ekibi tarafından geliştirilen ve metin sınıflandırma işlemlerinde kullanılan açık kaynak bir kütüphanedir. Daha hafif (lightweight) olması ile ön plana çıkmaktadır. Cümle ve sözcükleri vektörlere dönüştürerek gerekli yerlerde kullanılmasını sağlamaktadır. Model boyutu küçültüldüğü zaman mobil cihazlara da uygun hale gelmektedir (FASTTEXT, 2020). FastText ile 1 milyardan fazla kelimeyi çok çekirdekli işlemciler (CPU) ile 10 dakikadan daha kısa bir sürede eğitebilir ve yarım milyon cümleyi 312 bin sınıfa sınıflandırmak bir dakikadan daha kısa vakit almaktadır (Joulin vd., 2016).

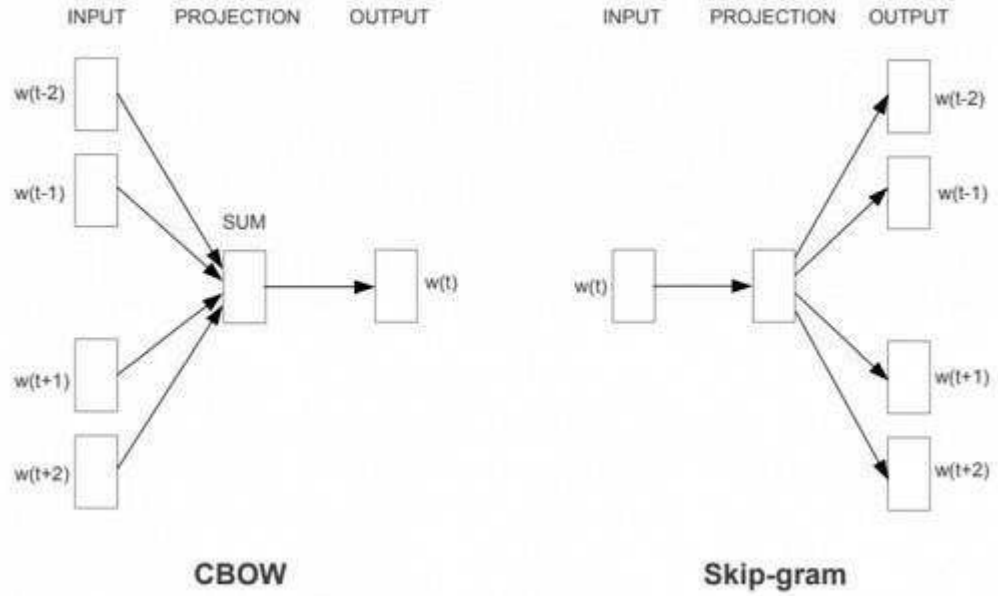
FastText modeli, yapı bakımından zengin olan diller ve ender kelimeler için fazlasıyla verimli kelime öbeği yapısı sağlar. Bir kelime metin belgelerinden kelime öbeklerini elde etmek için kullanılan etkili bir modeldir. Kelime öbeklerinin belirlenmesinde sıklıkla kullanılan n-gram yöntemini de içerisinde barındırmaktadır (Onan, 2020).

FastText kelime temsilleri için iki model kullanır: SKIPGRAM ve CBOW (Continuous-Bag-Of-Words). Skipgram modeli, tahminleri, hedef kelimenin yakınındaki kelimelere bakarak öğrenir. Diğer yandan, Cbow modeli ise tahminlerini hedef kelimenin bulunduğu içeriğe bakar öğrenir. Pratikte Skipgram modellerinin alt kelime bilgileriyle Cbow'dan daha iyi çalıştığını gözlemlenmiştir (FASTTEXT, 2020) Şekil 2.4'te çalışma yöntemi gösterilmiştir.



Şekil 2.4 CBOW ve SKIPGRAM Çalışma Yöntemi

FastText kelime temsilleri için iki model kullanır: SKIPGRAM ve CBOW (continuous-bag-of-words). Skipgram modeli, tahminleri, hedef kelimenin yakınındaki kelimelere bakarak öğrenir. Şekil 2.5'te çalışma yönteminin ikinci bir gösterimi yer almaktadır.



Şekil 2.5 CBOW ve SKIPGRAM Çalışma Yöntemi 2

2.4 Oyunlaştırma (Gamification)

Oyunlaştırma, günümüzde eğitim alanından, pazarlama ve ticarete kadar oldukça geniş bir alanda kullanılan bir yöntemdir. Temel dayanağı, insanların bu yaklaşım ile bir işe veya uğraşa daha motive, ilgili ve odaklı olmalarını sağlamayı amaçlamaktadır. Bu yöntemin günümüzde sık tercih edilmesindeki en önemli neden, oyun ve oyun kültürünün günden güne giderek yaygınlaşmasıdır. Yapılan çalışmalara göre yeni jenerasyonun %95'inden fazlası oyunlarla oldukça fazla vakit geçirmektedir. Bu da oyunlaştırmanın önemini her geçen gün artırmaktadır (Bozkurt & Kumtepe, 2014).

Özellikle mobil dünyadaki birçok mobil uygulama oyunlaştırma tekniğinden faydalanmaktadır. Örneğin, mobil operatörlerin uygulamalarında belirli zaman aralıklarında insanlara ufak hediyeler veren mini oyunlar ve oyunlaştırmalar yer almaktadır. Özellikle telefonlarda bulunan hareket sensörleri kullanılarak yapılan, telefonu sallayarak ödül kazandırma ya da çeşitli bulmacalar ile kampanyalardan indirimler kazandırma gibi uygulamalar oldukça yaygınlaşmaktadır. Bunlardan bazıları;

Turkcell, Dijital Operatör uygulamasında yer alan salla kazan oyunu. N11 e-ticaret sitesinde her yıl yapılan 11 Kasım etkinliklerindeki, kupon kapmaca ve çarkıfelek oyunları örnek olarak gösterilebilir.

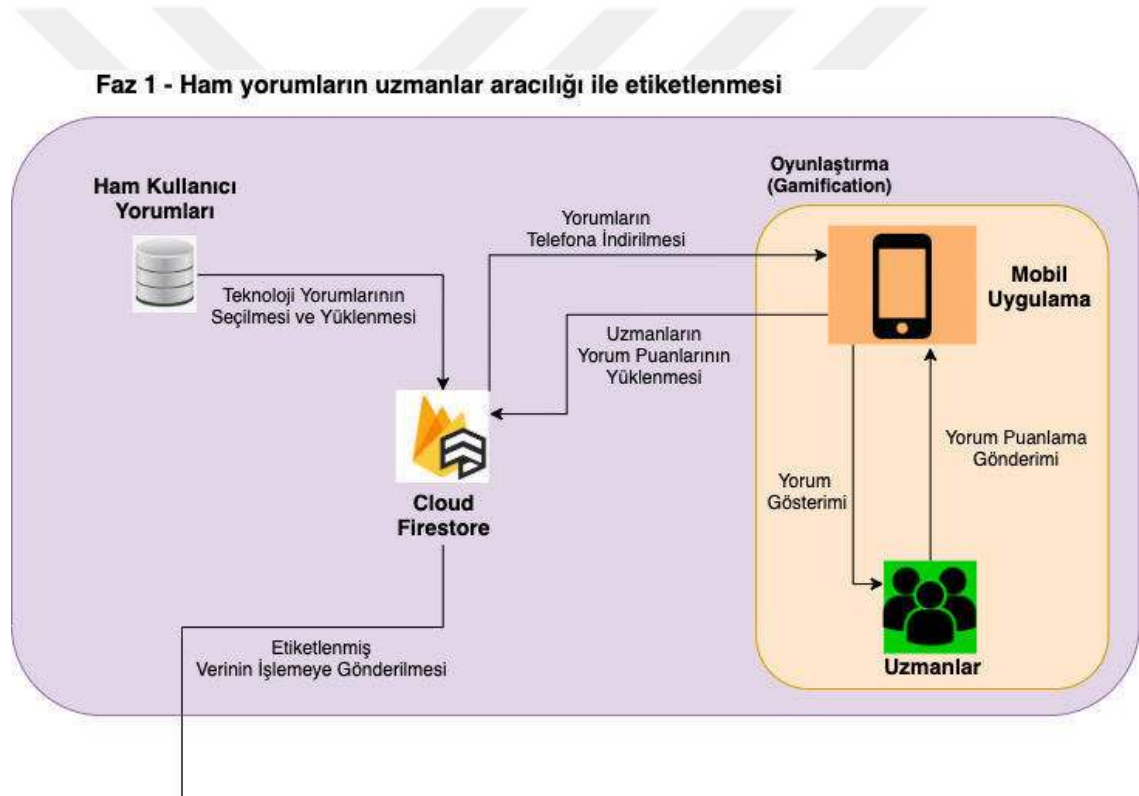
2.4.1 Neden Oyunlaştırma

Oyunlar, insan hayatında uzun yıllardır var olan ama önemi son yıllarda çok hızlı bir şekilde artan bir sektör halini almıştır. Bunun en etkili nedenlerinden biri dijital oyun dünyasının son yıllarda yaptığı hızlı ataklar ve sektörün devasa bir hal almasıdır. Bunu gören ticari kuruluşların oyunlaştırmayı kendi alanlarındaki ilgiyi arttırmak için kullanması sonucu, ticari gelirlerinde ciddi kar elde ettikleri görülmüştür. Ticari kullanım dışında eğitim, öğretim ve akademik çalışmalarda da oyunlaştırma son derece önem kazanmıştır (Bozkurt & Kumtepe, 2014).

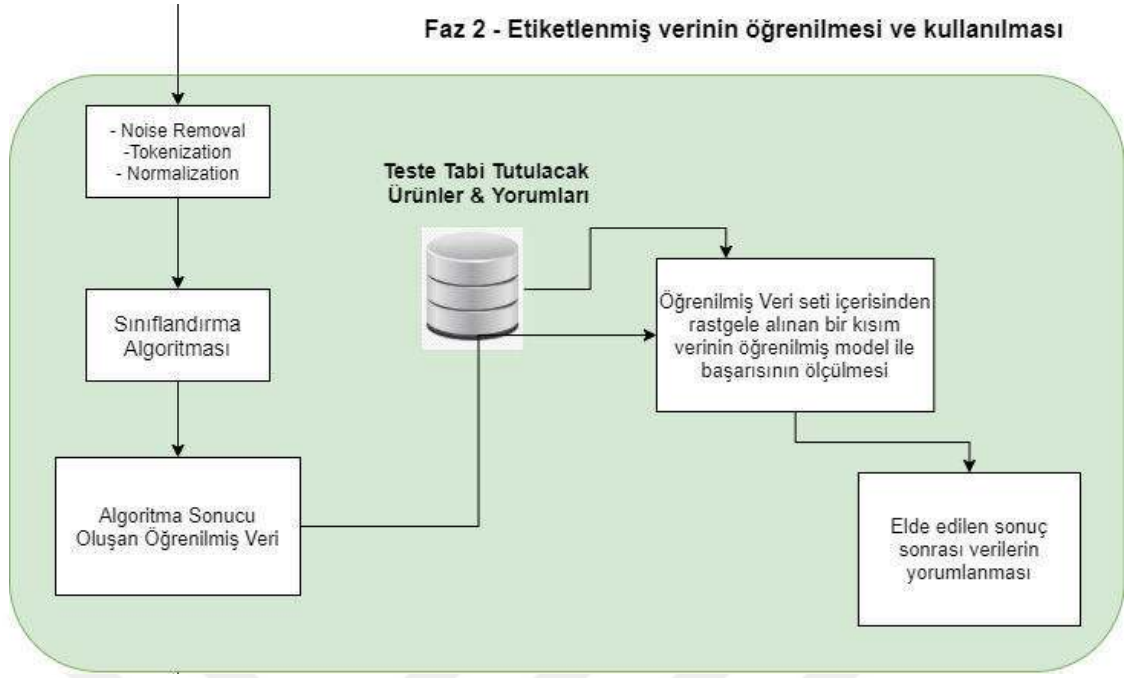
Bu tezin yazıldığı dönemde, yaşanan salgın nedeniyle, dünyadaki birçok ülke sokağa çıkma yasakları uygulanmıştır. Bu yasaklardan dolayı evinden çıkamayan insanların birçoğu zaman geçirmek için kendini oyunlara vermiştir. Özellikle satranç, tavla, yapboz gibi masaüstü oyunları satan firmaların talebe yetişemedikleri birçok haber kanalında yer bulmuştur. Dijital oyun firmaları arka arkaya rekor satışlar elde etmiştir. Oyunlaştırmanın ne denli güçlü bir ilgi aracı olduğu bu dönemde bir kez daha görülmüştür.

3. ARAÇLAR VE YÖNTEMLER

Bu çalışmada ilk olarak verilerin toplanması aşaması gerçekleştirilmiştir. Elde edilen veriler Json formatına döndürülmüştür. Formatlanan veriler manuel olarak etiketlenmek üzere bir mobil uygulamaya uygun hale getirilmiş ve online bir veri tabanı platformu olan Firestore'a aktarılmıştır. Firestore'daki verilerle etkileşime girip, etiketlemenin sağlanması için Android Studio aracı ile bir mobil oyun uygulaması yazılmıştır. Mobil oyun üzerinden etiketlemeyi yapacak olan kullanıcılar arasında ödüllü bir yarışma düzenlenmiş olup oyunlaştırma tekniği kullanılmıştır. Bu etiketlemeler sonucu elde edilen veriler Makine öğrenmesi teknikleri kullanılarak sınıflandırılmış olup sonrasında kullanıcı yorumları ve puanları arasındaki ilişki hakkında bilgiler edinilmiştir. Şekil 3.1 ve Şekil 3.2 de çalışmada kullanılan araç ve yöntemlerin şeması yer almaktadır.



Şekil 3.1 Araç ve Yöntemler Şeması Faz 1



Şekil 3.2 Araç ve Yöntemler Şeması Faz 2

3.1 Kullanılan Araç ve Gereçler

Ham veriyi işlemekten geçirmek için ve sonuç elde etmek için birçok araç, gereç ve programlama dili kullanıldı. Veri setini etiketlenebilir, modellenilebilir şekle getirmek için kullanılan araç ve programlama dilleri: XML, JSON, Kotlin, Java, IntelliJ, Android Studio, Firestore, SQLite, Postman.

3.1.1 Extensible markup language (XML)

XML (Genişletilebilir İşaretleme Dili), XML verilerin transferi, depolanması, sorgulanması, mobil uygulama arayüz şemaları olarak kullanılan bir veri standardıdır. XML android uygulamalardaki arayüz tasarımlarında kullanılır. Bu tez çalışmasında XML, Android oyun uygulamasındaki arayüz tasarımı için kullanılmıştır. XML yapısı Şekil 3.3'te gösterilmiştir. Örnekteki XML'de, etiketleştirme yapacak kullanıcıya sunulan seçeneklerden "Çok Olumlu" seçeneği için tasarlanmış buton tasarımıdır. LinearLayout denilen katman içinde, dikey olarak yerleştirilen ImageView adlı resim göstermeye yarayan katman, TextView adlı metin göstermeye yarayan katman, alt alta gösterilerek seçenek görseli sunulmaktadır.

```

<LinearLayout
    android:id="@+id/llVeryPositive"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="5dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="5dp"
    android:background="@drawable/bg_shape"
    android:orientation="vertical"
    android:padding="10dp">

    <ImageView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_gravity="center"
        android:src="@drawable/mood_3_white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Çok Olumlu"
        android:textSize="16sp"
        android:textColor="@android:color/white"
        android:textStyle="bold" />
</LinearLayout>

```

Şekil 3.3 Genişletilebilir işaretleme dili (XML) örneği

3.1.2 JavaScript object notation (JSON)

JSON, farklı programlama dilleriyle yazılmış platformlar arasında veri aktarımı ve dopalamayı sağlayabilen yöntemlerden biridir. XML'den çok daha pratik bir kullanıma sahiptir. Hem hız hem de kapladığı alan bakımından kullanımı son yıllarda çok yaygınlaşmıştır (Albayrak, Koçer & Uslu, 2013). JSON formatının doğrudan sunucu kodu olarak kullanılması, sunucuların ve kullanıcıların gelişmesine büyük kolaylık sağlamaktadır.

Kullanıcı verilerinin formatlanıp mobil uygulama üzerinde saklanması ve bulut sistemi (Firestore) ile haberleşirken JSON yapısı aktif bir şekilde kullanılmıştır. Şekil 3.4'te kullanıcı verilerinin de görüldüğü örnek bir JSON parçası gösterilmiştir.

```
{
  "message": "Hello Mert",
  "surname": "Dolar",
  "listOfReview": [
    {
      "rating": "5",
      "review": "3 yıldır tık demedi. :)",
    },
    {
      "rating": "5",
      "review": "3 yıldır kullanıyorum müthiş ",
    },
    {
      "rating": "4",
      "review": "Almaya karar verdim. Hemencecik geldi. Keyifle kullanıyorum",
    },
    {
      "rating": "5",
      "review": "Günlük kullanımınızı çok çok iyi karşılıyor kısaca mükemmel",
    },
    {
      "rating": "2",
      "review": "Çok beğenmedim. Tavsiye etmiyorum",
    }
  ]
}
```

Şekil 3.4 JavaScript object notation(Json) örneği

3.1.3 Kotlin programlama dili

Kotlin programlama dili, JetBrains adlı Çek Cumhuriyeti menşeli bir yazılım geliştirme firması tarafından geliştirilmiş JVM tabanlı bir programlama dilidir. İlk kararlı versiyonunu 22 Temmuz 2011 yılında yayınlamıştır (Jetbrains, 2020).

Kotlin tipik olarak Android geliştirme ile ilişkilidir ve bununla ilgili çoğu tartışma bunun etrafında dönmektedir. Ancak dilin sunabileceği birçok şey mevcut ve modern sunucu tarafı geliştiricileri için de ideal bir dildir (Samuel & Bocutiu, 2017).

Kotlin programlama dili, etiketlemede kullanılacak mobil uygulamanın geliştirilmesinde kullanılmıştır. Birçok veri dönüştürme ve mantık operasyonları mobil uygulama tarafında kotlin ile yazılmıştır.

```
private fun getAllDataFromFirestore(){
    val pd = ProgressDialog( context: this)
    pd.setMessage("Yükleniyor... İlk Açılış 1,2 dakika sürebilir. (KAPATMAYINIZ)")
    pd.show()
    firestoreDB.collection( collectionPath: "sentences")
        .get()
        .addOnSuccessListener { result ->
            var count = 0
            for (document : QueryDocumentSnapshot! in result) {
                if(!document.data["sentence"].toString().isNullOrEmpty() && !document.data["meaningPoint"].toString().isNullOrEmpty()
                    &&!document.data["userCount"].toString().isNullOrEmpty()){
                    try{
                        val sentenceStr : String = document.data["sentence"].toString()
                        val meaningPoint : Double = document.data["meaningPoint"].toString().toDouble()
                        val userCount : Int = document.data["userCount"].toString().toInt()
                        val sentence = Sentence(sentenceStr,meaningPoint, isApproved: false,document.id, rate: 0.0, url: "",userCount)
                        roomDB?.sentenceDao()?.insertAll(sentence)
                        Log.d(TAG, msg: "${document.id} => ${document.data}")
                    }
                    catch (ex : Exception){
                        Log.d( tag: "Exception",ex.message.toString())
                    }
                }
            }
            fillViewWithSentenceData()
            pd.hide()
        }
        .addOnFailureListener { exception ->
            Log.w(TAG, msg: "Error getting documents.", exception)
            pd.hide()
        }
}
```

Şekil 3.5 Kotlin ile geliştirilen metod örneği

Şekil 3.5'teki Kotlin ile yazılan metodun amacı, Bulut sistemi üzerinde bulunan Kullanıcı yorumları verisini telefona alıp lokal veri tabanı üzerine kaydeder.

3.1.4 Java programlama dili

Java ilk olarak Sun Microsystems tarafından 1995 yılında piyasaya sürülen bir programlama dili ve bilgi işlem platformudur. Dizüstü bilgisayarlardan veri depolama merkezlerine, oyun konsollarından bilimsel süper bilgisayarlara, cep telefonlarından Internet'e kadar birçok platformda Java kullanılmaktadır (Oracle, 2020). JVM tabanlı bir programlama dilidir.

Etiketlenen yorum verilerinin işlenmesinde, makine öğrenmesi ve sınıflandırma tekniklerinin uygulanmasında Java dili kullanılmıştır. Şekil 3.6'daki Java kodunun amacı, Verilen parametrelerle birlikte bir metin parçasının tahmin modelini besleyip sonrasında çıktı olarak modelde bulunan kategorilerden en yakından en uzağa doğru sıralamalı bir veri seti dönüşü yapar.


```

List<ScoredItem<String>> predict(String line, int k, float threshold) {
    IntVector words = new IntVector();
    IntVector labels = new IntVector();
    this.dict_.getLine(line, words, labels);
    if (words.isEmpty()) {
        return Collections.emptyList();
    } else {
        List<FloatIntPair> modelPredictions = this.model_.predict(words.copyOf(), threshold, k);
        List<ScoredItem<String>> result = new ArrayList(modelPredictions.size());
        Iterator var8 = modelPredictions.iterator();

        while(var8.hasNext()) {
            FloatIntPair pair = (FloatIntPair)var8.next();
            result.add(new ScoredItem(this.dict_.getLabel(pair.second), pair.first));
        }

        return result;
    }
}

```

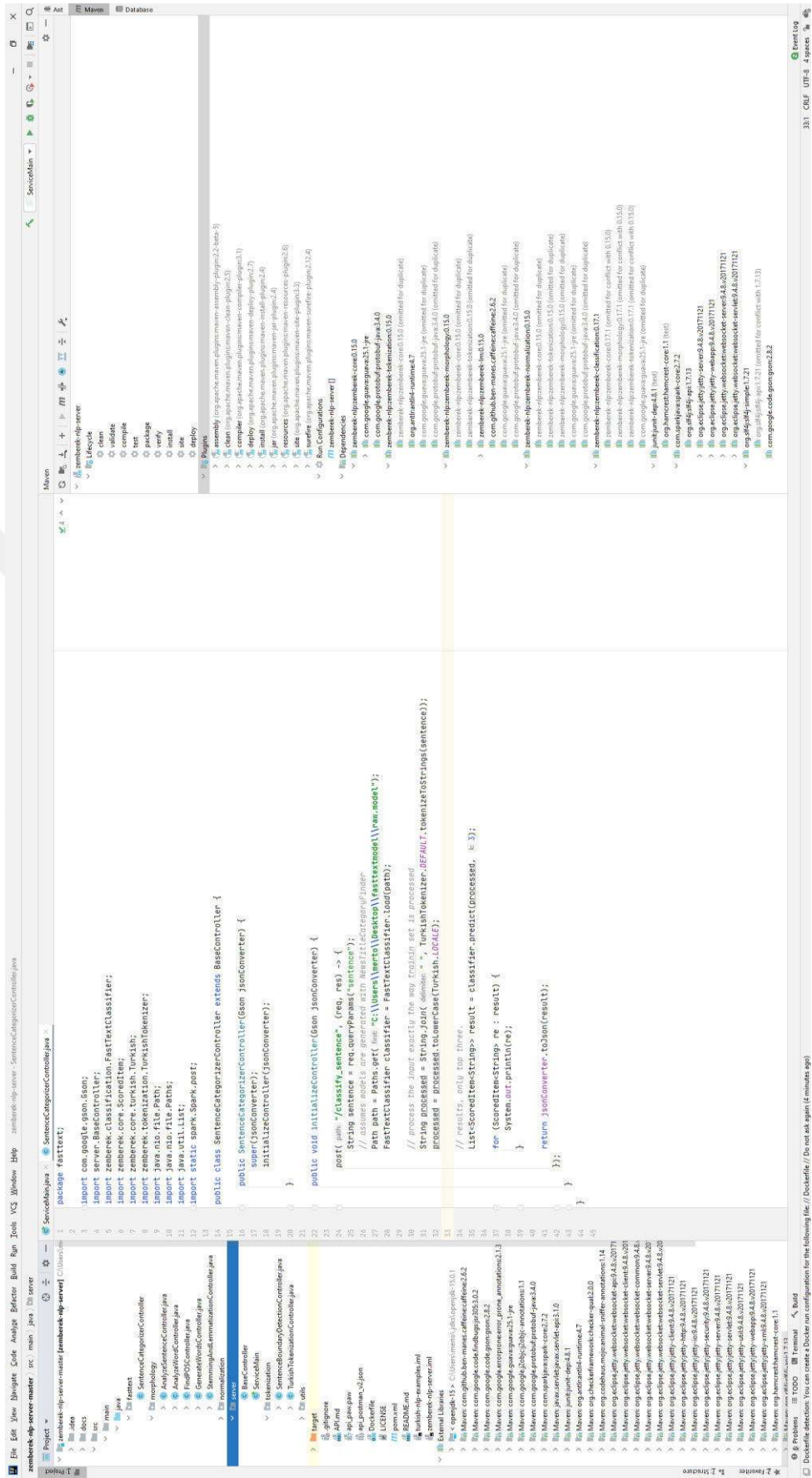
Şekil 3.6 Java ile geliştirilen metot örneği

3.1.5 IntelliJ yazılım geliştirme ortamı

IntelliJ, Java'da bilgisayar yazılımı geliştirmek için JetBrains tarafından geliştirilen kullanılan bir tümleşik geliştirme ortamıdır. Hem Apache 2 lisansı altında yayınlanan topluluk sürümü hem de ticari sürümü bulunmaktadır. Her ikisi de ticari amaçlı geliştirmelerde kullanılabilir.

IntelliJ yazılım geliştirme aracı ile Zemberek ve FastText kütüphanelerini birlikte kullanan Dockerized bir proje yazılmıştır. Bu proje REST altyapısı ile yazılmıştır. Bu rest altyapısının içinde çeşitli Zemberek tokenization, normalization, fastText tahmin metodları bulunmaktadır. Bu metodlar aracılığı ile yorum verileri işlenmiştir. Tez çalışmasında yöntem aşamasında detaylandırılarak anlatılmıştır.

Şekil 3.7'de IntelliJ geliştirme aracı görülmektedir. Sol panelde dosya, klasör ve sınıf isimleri listelenmektedir. Orta kısımda SentenceCategorizerController sınıfının kodları görülmektedir. Bu kod FastText kütüphanesinin sahip olduğu modelleme metodlarını kontrol eder, aynı zamanda /classify_sentence uzantılı bağlantıya bir cümle ile gidildiği zaman cümlelerin hangi kategoriye ait olduğuna dair sonuçlar döner.



Şekil 3.7 IntelliJ yazılım geliştirme aracı

3.1.6 Android Studio yazılım geliştirme ortamı

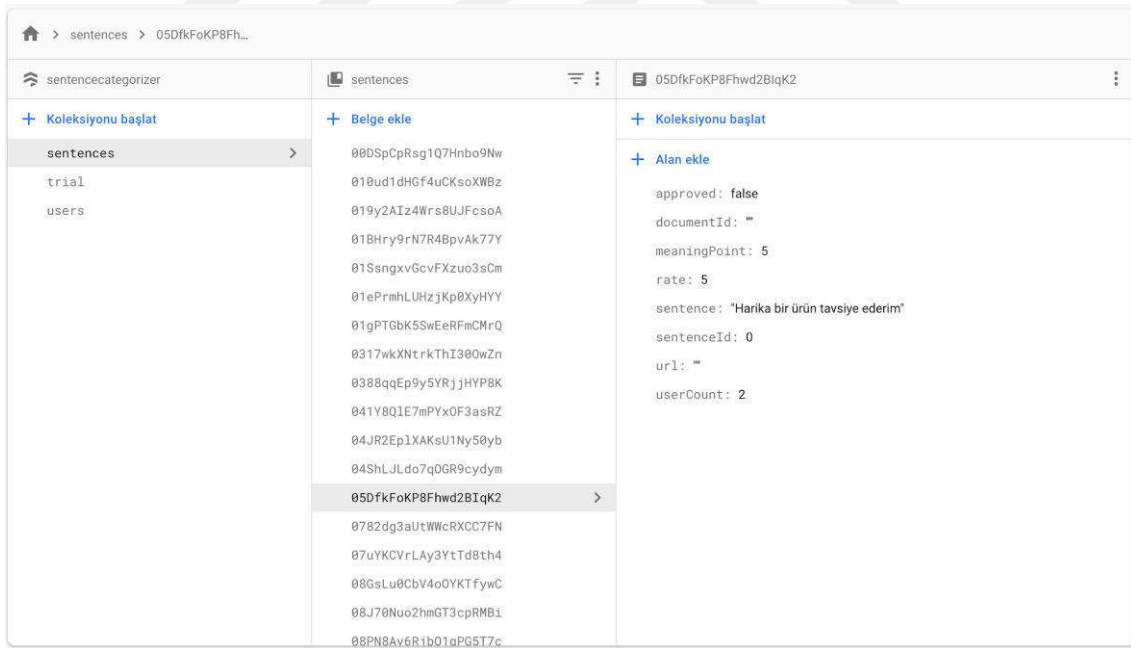
Android Studio, Java, Kotlin ve C++ dillerini destekleyen, Android işletim sistemine yazılım geliştirmek için JetBrains tarafından geliştirilen ve kullanılan bir geliştirme ortamıdır.

Android Studio, ürünlere yapılan kullanıcı verilerini etiketlemeyi oyunlaştırmak için geliştirilen mobil oyunda önemli bir rol almıştır. Bulut datasındaki verilerin işlenip yeniden bulut sistemine gönderilmesinde görev alan uygulamanın geliştirildiği araçtır. Şekil 3.8’de örnekte, android projesindeki DataCategorizer kotlin sınıfını görülmektedir. Sol panelde yer alan kısımda mevcut sınıf ve activity adı verilen android uygulama sayfalarını temsil eden sınıfları görülmektedir. Activity, android uygulamaların sayfalarının yaşam döngülerine sahip olan ve görsel ve mantıksal işlemlerin yapıldığı sınıflardır. DataCategorizer sınıfı normal bir sınıftır. Görevi ise etiketlenmiş yorum verilerinin FastText’in öğrenebileceği model formatına dönüştürme işlemini yapmaktır. Sağ panelde ise projeyi derleme, temizleme, tekrar çalıştırma gibi fonksiyonların olduğu kısa yollar görülmektedir.

3.1.7 Firestore bulut sistemi

Cloud Firestore, mobil ve web uygulamalar için verileri küresel ölçekte depolamayı, senkronize etmeyi ve sorgulamayı sağlayan bir NoSQL belge veri tabanıdır (Google, 2020). Verilerin, koleksiyonlar ve belgelerle kolayca yapılandırılmasını sağlar. İlgili verileri depolamak ve anlamlı sorgular kullanarak ihtiyaç duyulan verileri kolayca almak için hiyerarşiler oluşturulabilir. Tüm sorgular, sonuç kümesinin boyutuna göre ölçeklenir, böylece uygulama ilk günden itibaren ölçeklenmeye hazırdır.

Firestore etiketlenecek olan verinin bulut tarafında tutulmasını ve etiketlenmesinin sağlandığı süreçte önemli bir role sahip olmuştur. Etiketleyecek uzmanların oyun üzerinden yaptıkları manuel yorum etiketleme sonuçları bu platform üzerinde tutulmuş ve güncellenmiştir. Aynı zamanda oyunlaştırma yapılan mobil uygulamadaki yarışma için, kullanıcı skorlarının tutulması ve kazananın belirlenmesi için yine kullanıcı bilgileri ve etiketledikleri yorum sayısının bulunduğu bilgiler burada saklanmıştır. Şekil 3.9’da Firestore geliştirici konsol ekranı görülmektedir.



Şekil 3.9 Firestore geliştirici konsolu

3.1.8 SQLite veri tabanı

SQLite, küçük, hızlı, bağımsız, yüksek güvenli, tam özellikli bir SQL veri tabanı motoru uygulayan bir C dili kütüphanesidir. SQLite, dünyada en çok kullanılan veri tabanı motorudur. SQLite, tüm cep telefonlarında ve çoğu bilgisayarda kullanılır (SQLite Consortium, 2020).

SQLite, etiketlenen verilerin mobil uygulama üzerinde yedekleme olarak tutulmasını ve bulut sistemine sıklıkla sorgu atılmasını azaltarak, sunucu ve kullanıcı tarafında oluşacak sorgu trafiğini azaltmada katkı sağlamıştır. Android mobil uygulama üzerinde Google tarafından geliştirilen Room adlı kütüphane ile bütünleşmiş bir şekilde çalışmaktadır. Kullanıcı yorumlarının mobil cihazda tutulması sağlanmıştır.

3.1.9 Postman

Postman, API'ları paylaşmak, test etmek, belgelemek, monitör etmek için kullanılan bir rest client uygulamasıdır. En öne çıkan özelliklerinden biri tüm bu özellikler için çok hızlı ve kullanıcı dostu bir arayüze sahip olmasıdır (Postman, 2020).

3.2 Veri Setinin Elde Edilmesi

Günümüzde, dijital veri setlerine erişim önemli bir yere sahiptir. Özellikle e-ticaret ve bankacılık alanlarında kullanıcılara yönelik geliştirilecek ya da geliştirilmiş yeni teknolojiler üzerinde yapılan çalışmalarda sıklıkla kullanılan ve ihtiyaç duyulan bir araç haline almıştır. Bunun farkına varan birçok girişimci ve firma, bu verilerin satılması ve paylaşılması için internet siteleri ve platformlar oluşturmuşlardır. Bunlardan kaggle.com, netdata.com en çok bilinenlerinden bazılarıdır. Bu platformlar, web sitelerinden, dokümanlardan ya da uygulama çıktılarından elde edilen verilerin depolanabileceği ve satarak para kazanılabilen bir pazar yeridir. Depoladığınız bu verilere farklı birçok formatta anlık erişim imkânı sağlayarak uygulama geliştiriciler için bir veri kaynağı görevi görür.

Bu kaynaklar üzerinden yapılan araştırmalar sonucu e-ticaret sitelerindeki ürünlere yapılan yorum ve verilen puanların ham veri setine erişilmiştir. Bu veri seti üzerindeki en çok yoruma sahip kategori olan teknoloji kategorisi ele alınmıştır. 27 popüler teknoloji ürünü ve toplam 15601 tane yorum çalışmada kullanılmak üzere seçilmiştir. Yorumların içinden çok uzun olan yorumlar elenmiştir. Bunun için Twitter karakter limitine benzer bir yol izlenerek, 200 karakterden daha uzun olan yorumlar elenmiştir. Twitter'daki karakter limiti ve oluşan jargon sayesinde insanlar

anlatacaklarını çok daha kısa ve öz şekilde ifade etmişlerdir. (Meral, 2014). Bu eleme sonrası 11236 kullanıcı yorumu verisi etiketlenmek için hazır hale gelmiştir. Tablo 3.1’de Ürünler ve ürünlere ve yorum sayıları görülmektedir. Tablo 3.2’de ise Ham veri bilgileri yer almaktadır.

Tablo 3.1 Ürünler ve Yorum Sayıları

Ürünler	Yorum Sayısı
A4 Tech G3-280A 2.4 GHz Kablosuz Usb Mouse Siyah	262
Apple Iphone 7 32 GB	852
Asus wt425 kablosuz mouse	198
Everest sgm x7 kablolu oyuncu mouse ve mouse pad	433
Hp deskjet ink advantage 1115-yazıcı	230
Kingston 16gb mini metal usb bellek	494
Logitech m175 kablosuz nano mouse	675
Philips she1350-00 kulakiçi siyah kulaklık	219
Philips shm1900 mikrofonlu kulakustu kulaklık	144
Samsung 32gb microsd evo class10 48mb-sn hafıza kartı	225
Samsung galaxy c9 pro	264
Samsung galaxy j7 prime	575
Samsung galaxy note fan edition	391
Sandisk dual drive 32gb usb 3-0 otg usb bellek	237
Sandisk 32gb microsd 48mb/s class10 hafıza kartı	302
Sandisk cruzer fit 16gb usb bellek	543
Sandisk ssd plus 120gb 530mb 400mb/s sata 3 ssd	213
Sandisk ultra II 240gb 550mb 500mb/s sata 3 ssd	248
Sandisk cruzer blade 16gb usb bellek	314
Seagate expansion 1tb 2.5 usb 3.0-taşınabilir disk	287
Targus cn31-15-4-16 notebook çantası	293
Toshiba canvio basic 1tb 2.5 siyah taşınabilir disk	363
Tp link td-w8961n 300mbps adsl2 modem router anten	196
Tp link tl-wa850re 300mbps kablosuz kompakt access point	282
Xiaomi 10000 mah taşınabilir şarj cihazı	699
Xiaomi mi-band 2 akıllı bileklik	2167
Xiaomi mi tv-box 3 4k android tv box	130

Tablo 3.2 Ham Veri Bilgileri

Cümle Sayısı	22473
Kelime Sayısı	160851
Yorum Sayısı	11236
Karakter Sayısı	1093795

3.3 Etiketleme Öncesi Yapılan Çalışmalar

Verinin, etiketlemeye hazırlanması sürecinde en hızlı ve verimli şekilde nasıl sonuç elde edileceği konusunda araştırmalar yapılmıştır. Özellikle etiketlenecek verinin depolanacağı güvenli bir veri saklama merkezinin bulunması ve sonrasında bu verilere güvenli bir şekilde erişimin sağlanıp, etiketlenme işleminin yapılması için araştırmalar yapılmıştır. Araştırmalar neticesinde Google yazılım ekipleri tarafından geliştirilen Firebase çatısı altında yer alan özelliklerden biri olan Cloud Firestore Database tercih edilmiştir.

Cloud Firestore'un tercih edilme nedenlerini sıralayacak olursak, NoSql veri tabanına sahip olması ve aktarılan veriyi dikey düzlemde sorguladığı için büyük verilerde hızlı sonuç elde edilmesi, Veri güvenliği konusunda sağladığı sayısız önlem ve kontrol, android uygulamalara uygun bir SDK'ya sahip olması, uygulamaya yapılacak entegrasyonda harcanacak eforu ciddi oranda azaltması... Rakiplerinden ayıran önemli özelliklerinden biri de, günlük 20 bin yazma, 50 bin okuma ve 20 bin silme işlemine kadar ücretsiz olması.

3.3.1 Oyunlaştırma tekniğinin kullanımı

Günümüzde zaman, insanların hayatının merkezi haline gelmiştir. Planlamalar, programlar, toplantılar ve benzeri birçok etkinlikte zaman kavramı geçmektedir. Artık insanlar için dakikaların, hatta saniyelerin bile önem kazandığı bir çağda yaşamaktayız. Bilgisayarlar günden güne hızlanmakta ve insanların tahammül süreleride doğru orantılı bir şekilde kısalmaktadır. Bilgisayarlardaki ve mobil telefonlardaki uygulamalar öncelikli olarak, kullanıcıların hızlı bir şekilde ihtiyaca erişim sağlaması üzerine kurulmaktadır. Bunların hepsi bir araya geldiğinde insanlara programları dışında bir şeyler yaptırmak daha da zorlaşmış durumdadır. Artık yazılım uygulamaları insanların günlük programına uymak için yarışmaktadır. Bu tez çalışmasında oyunlaştırma tekniği, etiketleme işlemini, insanların programlarına dahil olacak bir forma dönüştürmeyi amaçlamıştır.

Bu tez çalışmasının yapıldığı dönem içerisinde yaşanmış olan virüs salgını ve karantina sürecinin de etkisiyle, oyunlaştırma tekniğine geçilmiştir. Pandemi öncesinde oyunlaştırma yerine anket sistemi ile etiketleme düşünülmüştür ancak yaşanan küresel olağan üstü durum nedeniyle insanları sosyal mesafe kuralları çerçevesinde motive etmek adına oyunlaştırma önemli bir rol oynamıştır. Etiketleme sürecine yardımcı olan kişiler arasında pandemi sürecini evinde geçirmiş olan ve okulların kapanmasından dolayı iş

yoğunluğu normalden daha az olan ilkokul ve lise öğretmenleri de katılım sağlamıştır. Bu sayede etiketleme işleminde daha çok veri elde edilmiştir. Etiketleme işlemi yapacak kullanıcılar rastgele seçilmemiş olup, deneye katkı sağlayacağına güvenilen ve verileri doğru girmesi yönünde ciddiye ve sorumluluk sahibi kişiler seçilmiştir. Katılım sağlayan kişi sayısı 6 olup, 1 sosyal bilimler uzmanı, 2 edebiyat ve dil anlatım öğretmeni ve 3 mühendisten oluşmaktadır.

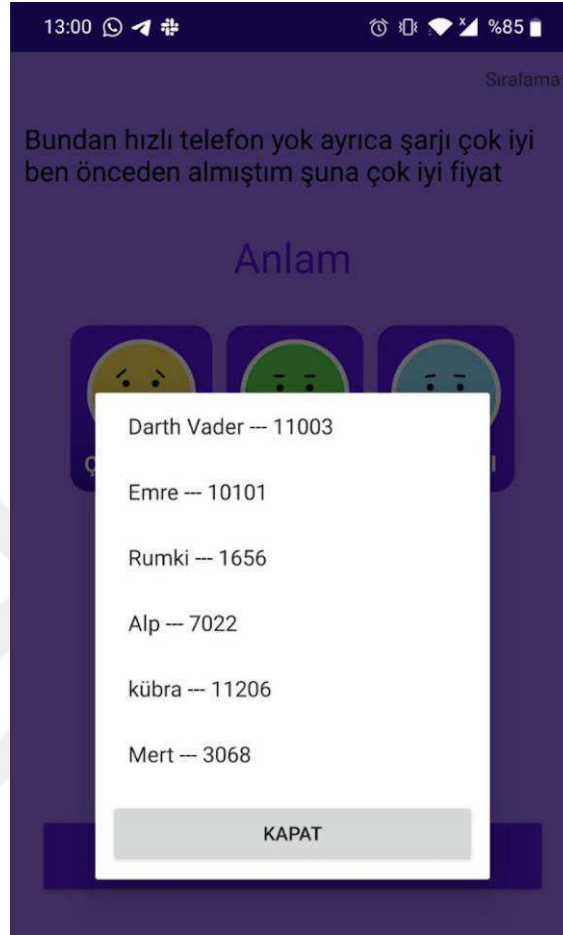
Bu projede, oyunlaştırma için bir mobil oyun uygulaması yazılmıştır. Mobil oyun aracılığı ile bir yarışma düzenlenmiştir. Hem bulut sistemi tarafında hem de mobil uygulama tarafında kullanıcı tabloları oluşturulup, skor tabloları yapılmıştır. Şekil 3.10'da bulut sisteminde tutulan kullanıcılar tablosunda bir kullanıcı bilgisi örneği görülmektedir. Sol bölümde "users" tablosundaki birincil anahtarlar (Primary Key) görülmektedir. Bu anahtarlar her kullanıcı için benzersiz olarak üretilir. Bu üretimi yeni kullanıcı eklendiğinde Firestore otomatik olarak sağlar. Etiketleme yapmış olan kullanıcıların listesine bu tablodan erişilebilir. Ayrıca bu liste üzerinden tek tek de gezinilebilir. Sağ taraftaki bölüm ise kullanıcı bilgilerinin detaylandırılmış hali görülmektedir.

users	l6ceZMTckcDP2ia3J3j5
+ Belge ekle	+ Koleksiyonu başlat
NrM2PGwDjkH1ITkS151d	+ Alan ekle
djXG1U3zHXV0IadLSSr3	documentId: ""
e4KWDOG1mWu71k5UMzva	nickName: "kübra"
ecAMC9Em0eJYRs0Ibzmİ	point: 11206
l6ceZMTckcDP2ia3J3j5 >	userId: 12
sBrBXoBhb521BWBQohb3	

Şekil 3.10 Kullanıcı tablosu

Kullanıcıların birbirlerinin skorlarını anlık takip edebildiği bir müsabaka ortamı yaratılmıştır. Kullanıcılar oyuna ilk girişlerinde, rumuzlarını girdikten sonra yarışmaya katılmışlardır ve Cloud Firestore üzerindeki users tablosuna rumuz ve skor bilgileri kaydolmaya başlamıştır. Müsabakada birinci olana ve belirli yorum etiketleme sayısını geçenlere, kitap ödülleri verilmiştir. Ödül sistemi rekabeti arttırmış olup, daha fazla

etiketleme verisinin elde edilmesi sağlamıştır. Şekil 3.11’de sağ üst kısmında görülen “Sıralama” yazısına basıldığında, kullanıcıların anlık olarak kendilerinin ve rakiplerinin skorlarını takip edebildikleri skor tablosu ekranı açılmaktadır.



Şekil 3.11 Skor tablosu

3.3.2 Etiketleme ortamının geliştirilmesi

Projedeki etiketleme işlemleri dışında, veri okuma, yazma, güncelleme ve silme işlemleri için de android uygulama kullanılmıştır. Android uygulama sadece etiketleme yapılacak kullanıcılara bir etiketleme aracı olmakla kalmayıp aynı zamanda ham verinin Cloud Firestore tarafına yüklenmesinde de görev almıştır.

Mobil uygulama projesi Android Studio üzerinden, geliştirme dili kotlin ve minimum SDK versiyonu 23 Android 6.0 Marshmallow olarak belirlenmiştir. Proje oluşturulduktan sonra projede kullanılacak olan kütüphaneler eklenmiştir. Şekil 3.12’de Kullanılan kütüphaneler ve versiyon numaraları görülmektedir. İçerisinde Firebase

kelimesi geçen kütüphanelerden, bulut sistemi, analitik veri toplama, uygulama çökme verilerini toplama gibi özelliklere sahip kütüphaneler eklenmiştir. İçeriğinde room kelimesi geçen kütüphaneler ise Google mühendisleri tarafından geliştirilen, androide kolay bir şekilde sqllite entegrasyonu sağlamaktadır. İçeriğinde Gson kelimesi geçen kütüphane ise json dosyalarını modellemeye ya da tam tersi çevirme işlemlerinde kullanılan yaygın bir kütüphanedir.

```
implementation 'com.google.firebase:firebase-analytics:17.2.2'
implementation 'com.google.firebase:firebase-firestore-ktx:21.5.0'

def room_version = "2.2.5"

implementation "androidx.room:room-runtime:$room_version"
kapt "androidx.room:room-compiler:$room_version"

implementation 'com.google.firebase:firebase-analytics-ktx:17.5.0'
implementation 'com.google.firebase:firebase-crashlytics-ktx:17.2.1'

implementation 'com.google.code.gson:gson:2.8.6'
```

Şekil 3.12 Mobil projede kullanılan kütüphaneler

Kütüphane entegrasyonları sonrası ilk olarak verilerin bulut sistemine taşınması için, UploaderActivity adında bir sınıf oluşturulmuştur. Bu sınıf içerisinde yer alan “jsonToObject” ve “loadJsonFromAsset” metodları kullanılarak, eldeki ham veri projede bulunan “Assets” klasöründen alınıp, önce byteArray tipinden json string datasına, ardından da json datasından ise kotlin data class model objesine dönüştürülmüştür. Dönüştürülen model “uploadData” metodu ile bulut sistemine kaydedilmiştir. Tüm yorum verileri yüklendikten sonra bulut sisteminde verilerin eksiksiz eklenip eklenmediği kontrol edilmiştir.

Verilerin bulut sistemine yüklenmesinin ardından mobil uygulamanın arayüzü ve işleyiş yapısı için çalışmalar başlamıştır. Öncelikle uygulama arayüzünde kullanılmak üzere çeşitli ikon ve resimler bulunmuştur. İkon ve resimler, android proje yapısında her zaman yer alan ve resimlere erişimin sağlanmasında görevli olan “drawable” klasörüne eklenmiştir. Ardından kullanıcıların basit bir şekilde kullanacakları, ürüne yapılan müşteri yorumunun bulunduğu ve altında çoktan seçmeli olarak yer alan etiketlerin bulunduğu arayüz tasarlanmıştır. Etiketleme yapacak uzmanların karşılaştıkları kullanıcı

arayüzü Şekil 3.13’te de görüldüğü gibidir. Daha önceden de anlatıldığı gibi “Sıralama” butonu, uzmanlar arasındaki skoru takip etmek amacıyla skor tablosu ekranını açmaktadır. Sıralama butonunun altında ürüne yapılan müşteri yorumu görülmektedir. Hemen altında etiketleme yapacak uzmana hatırlatma yapmak için cümlelerin anlam olarak olumluluk durumuna bakması için eklenmiş bir metin alanı bulunmaktadır. Sonrasında ise çoktan seçmeli alan ve seçilen şıkkın etrafında çıkan yeşil çerçeve görülmektedir. En altta ise “Onayla” butonu yer almaktadır. Kullanıcı bu butona basmadan, etiketlenen veri bulut sistemine gönderilmemektedir. “Onayla” butonu, işaretlemeye yapılacak herhangi bir hatalı tıklamada kullanıcının etiket bilgisinin bulut sistemine hemen gönderilmemesi adına yapılmış bir emniyet butonu olarak görev almaktadır. Etiketleme yapacak uzmanın cümleyi, anlam olarak kategorize ettikten sonra bir etiket seçmesi ve ardından “Onayla” butonuna basması ile bulut sisteminde aynı id bilgisine sahip cümle güncellenir.



Şekil 3.13 Mobil oyun uygulamasının arayüzü

3.4 Veri Setinin Etiketlenmesi

Veri setindeki verilerden 15.09.2020 ile 15.10.2020 tarihleri arasında 11236 kullanıcı yorumu, içlerinde dil ve anlatım bilgisinde de uzman farklı kişiler tarafından manuel etiketleme yapılarak eğitim verisi elde edilmiştir. Etiketleme yapılırken çok olumlu, olumlu, nötr, olumsuz ve çok olumsuz olmak üzere 5 sınıf seçilmiştir. Cümle bütünlüğüne göre sezgisel yaklaşımla yorumların sınıflarının belirlenmesi sağlanmıştır. Etiketleme işlemi esnasında etiketleme sınıfları veri tabanında ondalık sistemde tutulmuştur. Bu sayede etiketleme tamamlandıktan sonra tam sayı veriler ile ondalık veriler ayrıştırılmıştır. Bu işlemin sonucunda; kullanıcıların %87.08 oranında ortak görüşe sahip olduğu görülmüş olup etiketleme güvenilirliği test edilmiştir.

Tablo 3.3 Etiketlenen Yorum ve Kişi Dağılımı

	1 Kişi	2 Kişi	3 Kişi	4 Kişi	5 Kişi	Toplam
Etiketlenen Yorum Sayısı	0	3142	8059	34	1	11236

Tablo 3.3'te yorumlara, kaç kişinin etiketleme yaptığı görülmektedir. Örneğin, 8059 tane yoruma 3 kişinin etiketleme yapmış olduğu görülmektedir. Bir cümleye yapılan etiketleme sayısı ne kadar fazla ise, o cümlemin doğru sınıfa yerleşme ihtimali doğru orantılı bir şekilde artmaktadır. Sınıflandırmada yapılacak hataları azaltmak için, etiketleme işlemlerinde, her cümlede en az iki kişinin etiketlemesinin yer alması konusuna dikkat edilmiştir.

3.5 Etiketlenen Verinin NLP Teknikleri ile Düzenlenmesi

Etiketleme işlemleri tamamlandıktan sonra oluşan etiketleme datalarının arasında yer alan ondalıklı puanlar tam sayılara yuvarlanmıştır. Yuvarlama işleminden sonraki durum tablo 3.4'te gösterilmiştir.

Tablo 3.4 Etiketlenen Yorumların Sınıflara Dağılımı

Çok Olumlu	Olumlu	Ortalama	Olumsuz	Çok Olumsuz
2627	6265	584	240	66

Etiketlenen verinin bulut sisteminden alınıp dönüştürülmesi için Android uygulama tarafında DownloaderActivity ve DataCategorizer sınıfları oluşturulmuştur. DownloaderActivity Cloud Firestore üzerindeki verileri indirip cihaza json formatında kaydetmekle görevlidir. DataCategorizer ise Json formatında kaydedilen veriyi okuyup, fastText'in işleyebildiği model formatına dönüştürüp metin dosyası oluşturmaktadır. Oluşan metin dosyasındaki yorum verileri NLP teknikleri kullanılmak üzere hazırdır. NLP tekniklerinin kullanılması için Türkçe dil işlemedeki en çok bilinen ve en başarılı dil işleme kaynaklarından biri olan Zemberek kütüphanesi kullanılmıştır. Zemberek kütüphanesini docker mikroservis(Konteyner halinde taşınabilir) yapısında çalışmasını sağlayan ve temel özelliklerini webservis halinde sunan açık kaynaklı bir java proje bulunmuştur. Bu projeye, tez çalışması için gerekli olan fastText ile ilgili gerekli sınıflar ve metodlar yazılmıştır.

3.5.1 Noise Removal tekniğinin uygulanması

Bu teknik ile kullanıcı yorumlarında bulunan html, xml ya da wysiwyg yapısındaki modeli direkt olarak etkileyecek tagların veriden ayıklanması sağlanmıştır. Yorum datalarında bu tip yapılara çok fazla rastlanmamış olsa dahi oluşturulacak eğitim setinin temizlenmesinde kullanılmıştır.

3.5.2 Tokenization tekniğinin uygulanması

Bu tekniğin uygulanmasının nedeni, bir sonraki adım olan Normalization tekniğini kullanmadan önce ön hazırlık olarak cümleleri kelime parçacıklarına bölmektir.

3.5.3 Normalization tekniğinin uygulanması

Bu teknikte ise, tokenization tekniği ile ayrıştırılan kelimelerin temizlenip en basit haline getirilmesini amaçlanmıştır. Normalization tekniğinin alt dalı olan stemming yöntemiyle kelimelerin kökleri elde edilmeye çalışılmıştır.

3.5.4 Küçük harf dönüşümü ve noktalama işaretlerinin silinmesi

Veriyi mümkün olduğunca temiz ve kelime öbeklerinden oluşan bir yapıya büründürmek için noktalama işaretleri kaldırılmıştır. Ayrıca metindeki benzer kelimelerdeki harf farklılıklarını en aza indirmek için tüm metin dosyasındaki harfler küçük harfe dönüştürülmüştür.

3.6 Eğitim ve Test Verilerinin Hazırlanması

NLP teknikleri uygulanmış veri, fastText üzerinden model oluşturmaya hazır hale gelmiştir. Zemberek ekibi, c++ ile yazılmış olan fastText kütüphanesini, java ile yazılmış olan Zemberek kütüphanesine, sınıflandırma modülü olarak eklemiştir. Tez çalışmasında da fastText tekniklerine Zemberek kütüphanesi üzerinden erişilmiştir.

Öncelikle bilgisayarda bulunan terminal ekranı açılmıştır. Terminal ekranında zemberek-full.jar dosyasının bulunduğu dizine gidilmiştir. Bu dizine oluşturulan eğitim datası eklenmiştir. Şekil 3.14'te bu dizine "cd" ve "ls" komutları yardımıyla gidilmesi gösterilmiştir.

```
Last login: Sat Dec 12 15:04:58 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Rfat-MacBook-Pro:~ rifatmertdolar$ cd Desktop
Rfat-MacBook-Pro:Desktop rifatmertdolar$ cd Zemberek
Rfat-MacBook-Pro:Zemberek rifatmertdolar$ ls
raw.train                zemberek-full.jar
Rfat-MacBook-Pro:Zemberek rifatmertdolar$
```

Şekil 3.14 Zemberek-full.jar ve Eğitim Verisi Terminal Gösterimi

Sonrasında fastText için kullanılacak eğitim ve test verilerini oluşturmak üzere aşağıdaki komut girilmiştir;

```
$ head -n 1152 comment-category-set >raw.test
$ tail -n 11236 comment-category-set >raw.train
```

Bu komut çalıştırıldıktan sonra 11236 yorum verisi içerisinde ilk 1152 tanesini alınır ve test verisi olarak kullanmak üzere ayrılıp raw.test adında yaratılan metin dosyasına yazılır. Kalan eğitim verisi ise raw.train metin dosyası olarak kayıt edilir.

Elde edilen raw.train dosyası ile model oluşturmak için ise aşağıdaki komut girilmiştir;

```
$ java -jar zemberek-full.jar TrainClassifier -i raw.train -o
raw.model -ec 50 -lr 0.1
```

Bu komut çalıştırıldıktan sonra artık öğrenilmiş verinin bulunduğu raw.model dosyası oluşturulmuştur. Oluşan bu model ile artık raw.test dosyası ile test yapıp başarı oranlar öğrenilebilir.

Veri setindeki en çok geçen keimeler ve kelime öbeklerine bakıldığında birçok 2’li söz öbeğine rastlanmıştır ve bundan dolayı fastText kütüphanesinin sahip olduğu parametrelerden biri olan wordNGrams parametresi ile n-gram algoritması kullanılmıştır. Bu algoritmayı kullanmak için aşağıdaki komut girilmiştir;

```
$ java -jar zemberek-full.jar TrainClassifier -i raw.train -o
raw.model -ec 50 -lr 0.1 --wordNGrams 2
```


Eldeki model verisi en çok 2’li söz öbeği barındırdığı için yukarıdaki komutta wordNGrams değeri 2 olarak girilmiştir. Artık veri setinin başarısını ölçmek için tüm işlemler yapılmıştır. Başarı oranlarını öğrenebilmek için aşağıdaki komut girilmiştir:

```
$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i  
raw.test
```

Bu komut sonrasında başarı oranları elde edilmiştir. Başarı oranları ve başarının artırılması ile ilgili elde edilen sonuçlar ve bulgular bölüm 4’te paylaşılmıştır.



4. SONUÇLAR VE BULGULAR

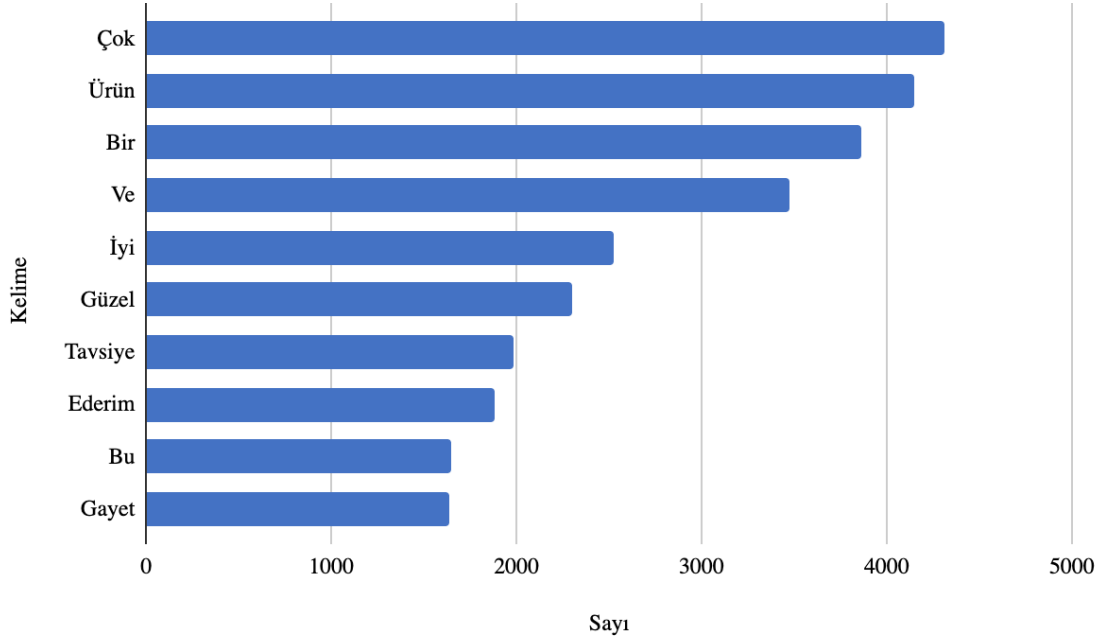
4.1 Eğitim Setinde En Çok Kullanılan Kelimeler ve Kelime Öbekleri

Eğitim veri setindeki en çok kullanılan kelimeler, ikili ve üçlü kelime öbekleri başarımlarında önemli bir rol oynamıştır. Kullanılan fastText kütüphanesinde yer alan wordNGrams parametresi sayesinde n kadar kelime öbeğini eğitim setinden çıkaran modeli teste tabi tuttuğumuzda, başarı oranı artmıştır. N-gram metodu, metin içerisindeki kelimelerin veya kelime içindeki karakterlerin n sayısı kadar ayrılarak vektörlere dönüştürülmesi işlemidir. Aşağıda en çok karşılaşılan örnekler Tablo ve Şekillerle gösterilmiştir.

Veri setindeki en çok kullanılan kelimeler Tablo 4.1 ve Şekil 4.1’de görülmektedir.

Tablo 4.1 En Çok Kullanılan Kelimeler

Kelime	Sayı	Oran
Çok	4314	5%
Ürün	4146	5%
Bir	3866	5%
Ve	3478	4%
İyi	2521	3%
Güzel	2301	3%
Tavsiye	1983	2%
Ederim	1884	2%
Bu	1644	2%
Gayet	1641	2%

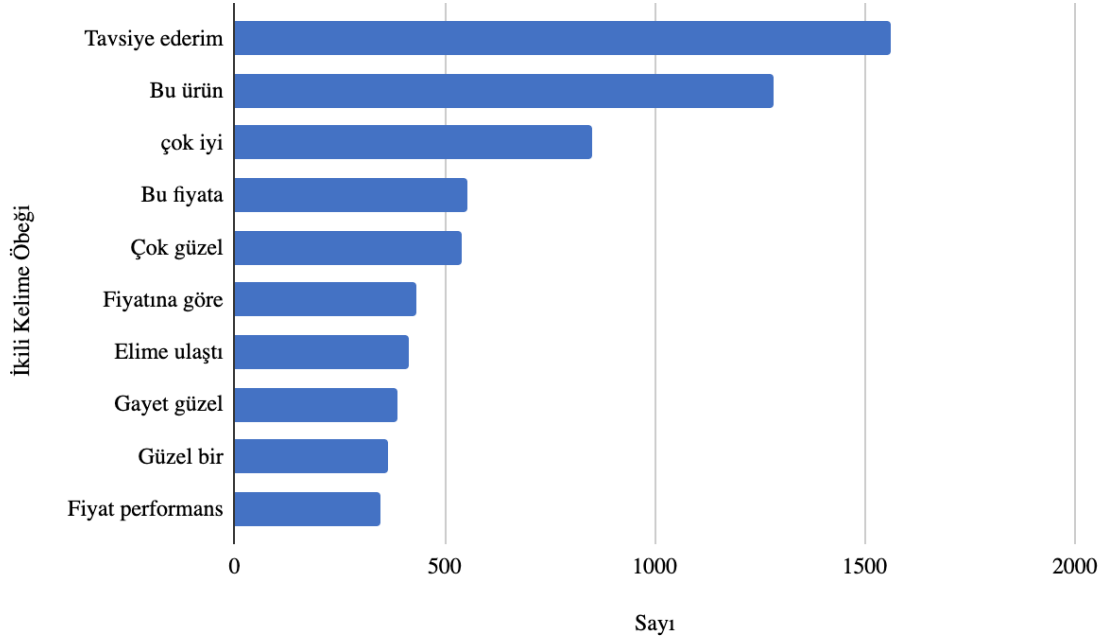


Şekil 4.1 En Çok Kullanılan Kelimeler

Veri setindeki en çok kullanılan ikili kelime öbekleri Tablo 4.2 ve Şekil 4.2’de görülmektedir.

Tablo 4.2 En Çok Kullanılan İkili Kelime Öbekleri

2 Kelime	Sayı	Oran
Tavsiye ederim	1560	7%
Bu ürün	1281	5%
Çok iyi	853	4%
Bu fiyata	556	2%
Çok güzel	539	2%
Fiyatına göre	432	2%
Elime ulaştı	416	2%
Gayet güzel	389	2%
Güzel bir	366	2%
Fiyat performans	345	1%

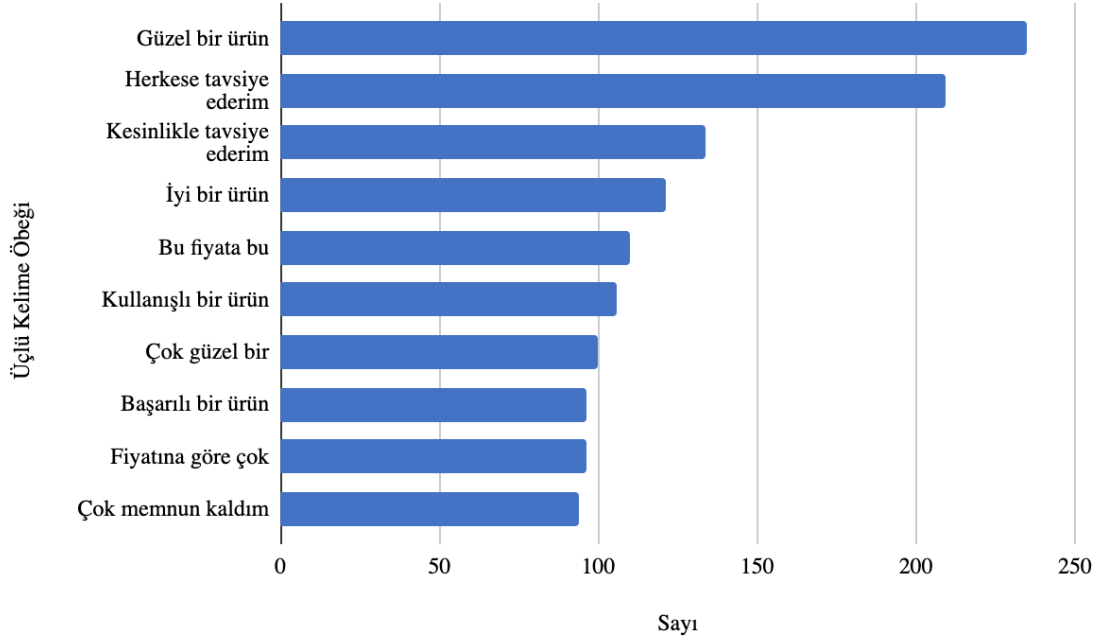


Şekil 4.2 En Çok Kullanılan İkili Kelime Öbeği

Veri setindeki en çok kullanılan üçlü kelime öbekleri Tablo 4.3 ve Şekil 4.3'te görülmektedir.

Tablo 4.3 En Çok Kullanılan İkili Kelime Öbekleri

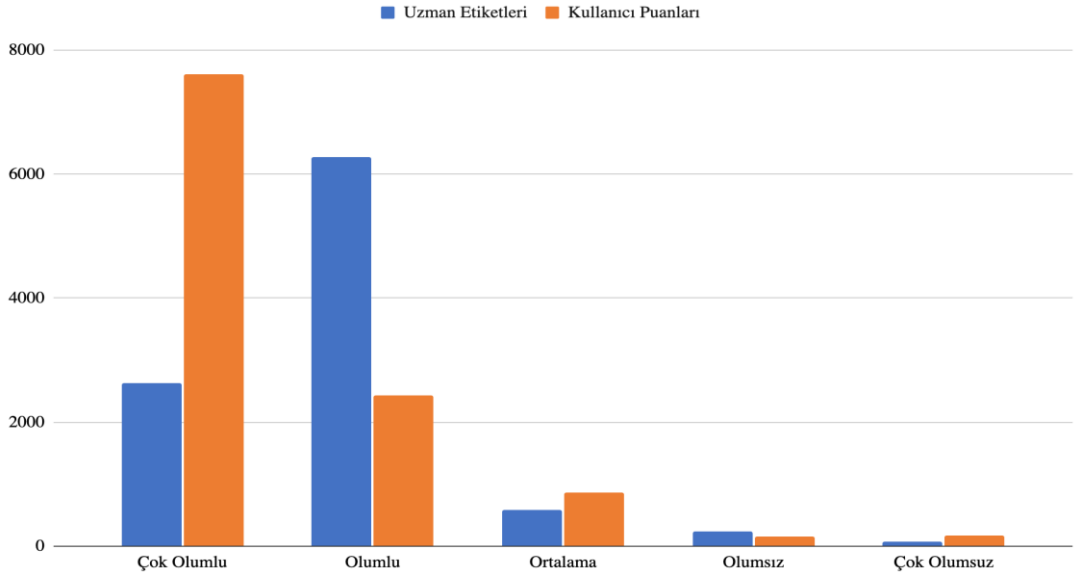
3 Kelime	Sayı	Oran
Güzel bir ürün	235	3%
Herkese tavsiye ederim	209	3%
Kesinlikle tavsiye ederim	134	2%
İyi bir ürün	121	2%
Bu fiyata bu	110	2%
Kullanışlı bir ürün	106	2%
Çok güzel bir	100	1%
Başarılı bir ürün	96	1%
Fiyatına göre çok	96	1%
Çok memnun kaldım	94	1%



Şekil 4.3 En Çok Kullanılan Üçlü Kelime Öbeği

4.2 Kullanıcı Puanları ve Uzman Etiketlerini Arasındaki İlişki

Uzman etiketlemeleri sonrası bulut sisteminde bulunan veri seti, eğitim veri seti formatına dönüştürülmeden önce, ürün adı, kullanıcı yorumu, kullanıcı puanı ve uzman etiketlemeleri parametrelerine sahip bir json formatında bulunmaktadır. Bu verilerden kullanıcı puanları ve uzman etiketleri arasındaki ilişki Şekil 4.4'te görülmektedir.



Şekil 4. 4 Uzman Etiketleri ve Kullanıcı Puanlarını Karşılaştırma

Tablo 4.4: Uzman Etiketleri ve Kullanıcı Puanları

	Çok Olumlu	Olumlu	Ortalama	Olumsuz	Çok Olumsuz
Uzman Etiketleri	2627	6265	584	240	66
Kullanıcı Puanları	7611	2435	869	146	175

Şekil 4.4 ve Tablo 4.4’te görüldüğü gibi özellikle çok olumlu ve olumlu kategorileri arasında ciddi geçişler olduğu görünmektedir. Bu fark ile ilgili yapılan manuel örnekleme çalışmalarında, ürünleri kullanan kullanıcıların, ürünlerle yaşadıkları ufak problemleri puanlarına yansıtmadıkları, ürünün genel olarak başarılı olması tam puan alması için yeterli olmuş. Etiketleme yapan uzmanlar ise ürünlere yapılan bu ufak sorun ve problemleri dikkate alarak etiketlemelerini yapmışlardır.

```
[
  {
    "userScore": "5",
    "meaningPoint": 4,
    "sentence": "Kablo su yaklaşık 60 cm.Biraz kısa geldi standardımı böyledir bilmiyorum ama güzel ürün.Tavsiye ederim.Kargo da bedava daha ne olsun."
  },
  {
    "userScore": "5",
    "meaningPoint": 4,
    "sentence": "kamerasi biraz düşük bunun disinda gercekten çok iyi telefon"
  },
  {
    "userScore": "5",
    "meaningPoint": 4,
    "sentence": "hafif ve kullanışlı çok rahatsız etmiyor ama uzun süre kullanımda bilekte geçici iz yapıyor."
  }
]
```

Şekil 4.5 Uzman Etiketleri, Yorum ve Puan örneği

Şekil 4.5'te görülen json metninde userScore, kullanıcı puanlarını, meaningPoint, uzman etiketlerini ve sentence, kullanıcı yorumlarını temsil etmektedir. Kullanıcılar ufak problemler yaşamalarına rağmen puan kırmayıp tam puan vermeyi tercih etmektedirler. Etiketleme yapan uzmanlar ise üründe ufakta olsa bir sorun olduğunu fark edip cümledeki negatif yorumuda hesaba katarak puan kırdıkları gözlemlenmektedir. Aynı durum negatif yorumlar içinde gözlemlenmiştir. Kullanıcıların, ürünler ile ilgili yaşadıkları problemler üzerine yoğunlaşıp, ürünün diğer olumlu özelliklerini değerlendirmeye katmadan en düşük puanı verdikleri görülmüştür. Kullanıcılar ürünleri kullanabildikleri sürece puan kırmaktan kaçınmışlar ve puanlara göre ürünün kusursuz, hatasız gibi görünmesine neden olmuşlardır.

4.3 FastText Test Sonuçları

FastText ile elde edilen sınıflandırma modeli ile, başarı oranı adım adım arttırılmaya çalışılmıştır. Başarının arttırılması için, eğitim veri setine çeşitli doğal dil işleme tekniklerinin yanı sıra n-gram ile kelime öbeklerini de yakalayarak, başarıda artış sağlanmaya çalışılmıştır. Başarının testi için başlık 3.6'daki test için hazırlık işlemleri ve model oluşturma adımları tekrarlanmıştır. Uygulanan NLP tekniklerinin sıralamasında "Zemberek ile NLP Sınıflandırma" kaynağından faydalanılmıştır (Zemberek ile NLP Sınıflandırma, 2020).

Öncelikle ham veri ile yapılan testte, eğitim verisi içerisinde alınan 350 yorum üzerinde yapılmıştır. Ham veri ile oluşturulan modelin başarı oranı %48 olarak Şekil 4.5'te görülmektedir.

```
I|16:06:05.129|Word + Label count = 21791
| Dictionary#readFromFile
I|16:06:05.130|Removing word and labels with small counts. Min word = 1, Min Label = 0
| Dictionary#readFromFile
I|16:06:05.162|Word count = 21786 , Label count = 5
| Dictionary#readFromFile
I|16:06:07.658|Training started.
| FastTextTrainer#train
100% [=====] 8374425/8374425 (0:00:01 / 0:00:00) lr: 0.000000
I|16:06:08.886|Saving classification model to raw.model
| TrainClassifier#run
Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.480 R@1: 0.480 Number of examples = 350
java.lang.NullPointerException
    at zemberek.apps.fasttext.EvaluateClassifier.run(EvaluateClassifier.java:53)
    at zemberek.apps.ConsoleApp.execute(ConsoleApp.java:18)
    at zemberek.apps.ApplicationRunner.main(ApplicationRunner.java:27)
Usage: <main class> [options]
Options:
* --input, -i
Input test set with correct labels.
```

Şekil 4.6 Ham Veri Test sonucu

Başarı oranını artırmak için noise removal, tokenization ve normalization işlemleri uygulanmıştır. Bu işlemler ile test edilecek yorum sayısı 750'ye çıkarılmıştır. Şekil 4.6'da görüldüğü gibi başarı oranı %53,2'ye çıkmıştır.


```

Rfat-MacBook-Pro:Desktop rifatmertdolar$ cd Zemberek
Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.532 R@1: 0.532 Number of examples = 750
java.lang.NullPointerException
    at zemberek.apps.fasttext.EvaluateClassifier.run(EvaluateClassifier.java:53)
    at zemberek.apps.ConsoleApp.execute(ConsoleApp.java:18)
    at zemberek.apps.ApplicationRunner.main(ApplicationRunner.java:27)
Usage: <main class> [options]
Options:
* --input, -i
    Input test set with correct labels.
* --model, -m
    Model file.
--output, -o
    Output file where prediction results will be written. If not provided,
    [input].predictions will be generated.
--predictionCount, -k
    Amount of top predictions.
    Default: 1

Rfat-MacBook-Pro:Zemberek rifatmertdolar$

```

Şekil 4.7 NLP ve Veri Sayısı Artırımı ile Elde Edilen Test Sonucu

Başarım oranını artırmak için NLP işlemlerine ek olarak eğitim veri seti içerisindeki noktalama işaretleri kaldırılmıştır. Test edilecek veri sayısı 750 olarak devam etmektedir. Şekil 4.7’de görüldüğü gibi Sonuç %54’e yükselmiştir. Noktalama işaretlerinin kaldırılmasının, başarı oranına fazla katkı sağlamadığı görülmektedir.

```

| TrainClassifier#run
Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.540 R@1: 0.540 Number of examples = 750
java.lang.NullPointerException
    at zemberek.apps.fasttext.EvaluateClassifier.run(EvaluateClassifier.java:53)
    at zemberek.apps.ConsoleApp.execute(ConsoleApp.java:18)
    at zemberek.apps.ApplicationRunner.main(ApplicationRunner.java:27)
Usage: <main class> [options]
Options:
* --input, -i
    Input test set with correct labels.
* --model, -m
    Model file.
--output, -o
    Output file where prediction results will be written. If not provided,
    [input].predictions will be generated.
--predictionCount, -k
    Amount of top predictions.
    Default: 1

Rfat-MacBook-Pro:Zemberek rifatmertdolar$

```

Şekil 4.8 Noktalama İşaretleri Kaldırılmış Test Sonucu

Başarım oranını daha çok artırmak için veri seti içerisindeki marka ve reklamlar kaldırılmıştır. Ayrıca test veri sayısı 1152'ye çıkarılmıştır. Başarı oranı %58,5 olarak gözlemlenmiştir.

```
| TrainClassifier#run
Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.585 R@1: 0.585 Number of examples = 1152
java.lang.NullPointerException
    at zemberek.apps.fasttext.EvaluateClassifier.run(EvaluateClassifier.java:53)
    at zemberek.apps.ConsoleApp.execute(ConsoleApp.java:18)
    at zemberek.apps.ApplicationRunner.main(ApplicationRunner.java:27)
Usage: <main class> [options]
Options:
* --input, -i
    Input test set with correct labels.
* --model, -m
    Model file.
--output, -o
    Output file where prediction results will be written. If not provided,
    [input].predictions will be generated.
--predictionCount, -k
    Amount of top predictions.
    Default: 1
Rfat-MacBook-Pro:Zemberek rifatmertdolar$
```

Şekil 4.9 Marka ve Reklam Kelimeleri Silinmiş Test Sonucu

Başarım oranını daha çok artırmak için veri setindeki tüm harfler küçük harfe dönüştürülmüştür. Küçük harfe dönüştürülen veri setinden oluşturulan model bir önceki test edilen veri sayısı ile aynı sayıda olan örnek veriler teste tabi tutulduğunda %60,3 başarı elde edildiği görülmüştür. Başarı oranı %60 barajını geçmiştir. Veri seti üzerinde yapılacak fiziksel değişikliklerin hepsi tamamlanmıştır. Bu aşamadan sonra sonuç başarımını arttırmak için fastText'in sahip olduğu özellik ve algoritmalarından faydalanılmıştır. Şekil 4.9'daki başarı oranı veri setinde yapılmış olan son değişiklikler ile elde edilmiş başarı oranıdır.

```

I|17:13:33.904|Training Started.
| TrainClassifier#run
I|17:13:33.939|Initialize dictionary and histograms.
| Dictionary#readFromFile
I|17:13:33.962|Loading text.
| Dictionary#readFromFile
I|17:13:34.065|Lines read: 100 (thousands)
| Dictionary#readFromFile
I|17:13:34.065|Word + Label count = 17544
| Dictionary#readFromFile
I|17:13:34.065|Removing word and labels with small counts. Min word = 1, Min Label = 0
| Dictionary#readFromFile
I|17:13:34.094|Word count = 17539 , Label count = 5
| Dictionary#readFromFile
I|17:13:36.533|Training started.
| FastTextTrainer#train
100% [=====] 7642126/7642126 (0:00:01 / 0:00:00) lr: 0.000000
I|17:13:37.751|Saving classification model to raw.model
| TrainClassifier#run
[Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test

Loading classification model...
Result = P@1: 0.603 R@1: 0.603 Number of examples = 1152

```

Şekil 4.10 Tüm Harfler Küçük Harfken Elde Edilen Test Sonucu

Artık başarı oranını artırmak için fastText'in sahip olduğu wordNGrams parametresine kelime öbeği sayısı girerek testlere devam edilmiştir. İlk olarak ikili kelime öbekleriyle olan başarı oranı ölçülmüştür. Başarı oranı %63,5 olarak Şekil 4.10'da görülmektedir. Model algoritmasına ikili Kelime öbeklerinin de dahil edilmesi başarı oranına olumlu katkıda bulunmuştur.

```

I|17:12:53.560|Training Started. | TrainClassifier#run
I|17:12:53.617|Initialize dictionary and histograms. | TrainClassifier#run
I|17:12:53.641|Loading text. | Dictionary#readFromFile
I|17:12:53.746|Lines read: 100 (thousands) | Dictionary#readFromFile
I|17:12:53.746|Word + Label count = 17538 | Dictionary#readFromFile
I|17:12:53.747|Removing word and labels with small counts. Min word = 1, Min Label = 0 | Dictionary#readFromFile
I|17:12:53.780|Word count = 17533 , Label count = 5 | Dictionary#readFromFile
I|17:12:56.227|Training started. | FastTextTrainer#train
100% [=====] 7640207/7640207 (0:00:01 / 0:00:00) lr: 0.000000
I|17:12:57.444|Saving classification model to raw.model | TrainClassifier#run
[Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.635 R@1: 0.635 Number of examples = 1152

```

Şekil 4.11 İkili Kelime Öbekleri ile Elde Edilen Test Sonucu

Son olarak üçlü kelime öbeklerinin de dahil edildiği bir test daha yapılmıştır. Üçlü kelime öbeklerinin de dahil edildiği testteki başarı oranı %65,3'e çıkarılmıştır. Böylece ham veri ile elde edilen %48 olan başarı, NLP yöntemleri ve n-gram kullanarak %65,3'e çıkmıştır. Şekil 4.11'de ikili, Şekil 4.12'de üçlü kelime öbekleri ile elde edilen test sonuçları görülmektedir.

```

I|17:22:36.142|Training Started. | TrainClassifier#run
I|17:22:36.176|Initialize dictionary and histograms. | TrainClassifier#run
I|17:22:36.199|Loading text. | Dictionary#readFromFile
I|17:22:36.304|Lines read: 100 (thousands) | Dictionary#readFromFile
I|17:22:36.304|Word + Label count = 17538 | Dictionary#readFromFile
I|17:22:36.304|Removing word and labels with small counts. Min word = 1, Min Label = 0 | Dictionary#readFromFile
I|17:22:36.336|Word count = 17533 , Label count = 5 | Dictionary#readFromFile
I|17:22:38.689|Training started. | FastTextTrainer#train
100% [=====] 7640321/7640321 (0:00:02 / 0:00:00) lr: 0.000000
I|17:22:40.903|Saving classification model to raw.model | TrainClassifier#run
[Rfat-MacBook-Pro:Zemberek rifatmertdolar$ java -jar zemberek-full.jar EvaluateClassifier -m raw.model -i raw.test
Loading classification model...
Result = P@1: 0.653 R@1: 0.653 Number of examples = 1152

```

Şekil 4.12 Üçlü Kelime Öbekleri ile Elde Edilen Test Sonucu

Tablo 4. 5 Başarım Özeti

Veri	%Başarım
Ham	48
Noise Removal + Tokenization + Normalization	53,2
Noktalama İşaretleri Kaldırılmış	54
Marka ve Reklamlar Kaldırılmış	58,5
Küçük Harf	60,3
2-gram	63,5
3-gram	65,3

Tablo 4.5'te ham veriden, işlenmiş veriye tüm başarı oranlarının özeti görülmektedir. %48 başarı oranında %65,3 başarı oranına çıkılmıştır. Yapılan çalışmalar sonucu başarı oranı %17,3 artırılmıştır.

5. TARTIŞMA VE GELECEK ÇALIŞMALARI

Günümüzde e-ticaret platformları, insanların hayatında önemli bir yere sahiptir. Özellikle son yıllarda vazgeçilmez bir hale gelmiştir. Bu tezin yazıldığı yıl itibari ile dünya çapında salgının getirdiği sosyal mesafe, evde kalmaya yönelik teşvik ve sokağa çıkma yasaklarıyla birlikte e-ticaret, altın çağlarından birini yaşamaktadır.

E-ticaret sitelerinde kullanıcıların alışveriş yaptıkları sırada alacakları ürünleri seçme konusunda başvurdukları ilk yollardan biri, daha önce ürünleri kullanıp deneyimleyen, sonrasında ürün hakkındaki görüşlerini paylaşan kullanıcıların yorumlarıdır. Bu tez çalışmasında, kullanıcı yorumlarının sınıflandırılması sağlanarak elde edilen sonuçlar ile gelecek çalışmalarına ışık tutacak çalışmalar yapılmış ve sonuçlar elde edilmiştir. Kullanıcı yorumlarının sınıflandırılması sağlandıktan sonra en çok dikkat çeken yerlerden bir tanesi, kullanıcıların yorumları ile puanları arasındaki ilişkiydi. Kullanıcılar ürünlerle ilgili yaşadıkları bazı problemleri, yorumlarında belirtirken, verdikleri puanlar, genellikle en iyi skor olurken, bazılarında ise yaşanan en ufak problemde, ürün hakkında en düşük puanı vermeleri, yorumlar ve puanlar arasında orantılı bir ilişki olmadığı gözlemlenmiştir. Bu tez çalışmasından sonraki çalışmalarda, kullanıcı puanları direkt kullanıcılardan alınmak yerine kullanıcı yorumları istenerek, makine öğrenmesi ile elde edilecek puanlar ile e-ticaret sitelerinde satılan ürünler için daha gerçekçi puanlar elde edilebilir. Bu sayede ürün listelerinde, kullanıcıları en çok memnun eden ürünler ön sıralarda yer alabilir. Ayrıca başka bir çalışma olarak, ürün ile alakasız yorumlarında elenmesi gibi özellikler üzerine çalışılarak, ürünler hakkında verilen puanlamalarda düzeltilmeler sağlanabilir.

Son zamanlarda yaygın olarak kullanılan cep telefonlarımızda bulunan uygulama mağazalarındaki uygulamaların bir kısmı siyasi, sosyal ya da farklı nedenlerden dolayı, o günün koşulları dahilinde birçok kullanıcının uygulama ya da verilen hizmetin dışında yapılan olumsuz kullanıcı yorumlarına maruz kalmaktadır. Bu yorumların filtre edilmesinde bu çalışmadaki sınıflandırma tekniklerinden fayda sağlanarak farklı çalışmalar yapılabilir.

Çalışmada elde edilen başarı oranında, e-ticaret ürünlerindeki yorumları yazan kullanıcıların, farklı sosyokültürel altyapıya sahip insanlardan oluşmasının etkisi hissedilmiştir. Bu tez çalışması için yapılan altyapı araştırmalarında karşılaşılan

Zemberek kütüphanesi içerisinde yer alan FastText modülünün kullanımı için yazılmış dokumanda test verisi olarak haber sitelerindeki haber başlıklarının kullanıldığı gözlemlenmiştir. Bu tip yorum verilerindeki imla, noktalama ve yazım yanlışları, e-ticaret sitelerindeki yorumlara oranla daha ender olmaktadır. Özellikle e-ticaret yorumlarına uygulanan NLP tekniklerinde, Türkçe için çok yeterli olmayan NLP tekniklerinin yanı sıra, yazım yanlışlarının da çok fazla olması, algoritmaların doğru çalışmasını zorlaştırmıştır.

Bu tez çalışmasına benzer bir tez çalışmasında insan gücü, yani etiketleme yapan insanlar olmadan 2 milyondan fazla yorumun sınıflandırıldığı gözlemlenmiştir. Kullanılan terim ağırlıklandırma yönteminde duygu skoru ölçülemeyeceği, sadece istatistiksel olarak bir metinde en değerli kelime nedir sorusunun cevabının bulunabileceği, en değerli kelimelerin duygu değerleri el ve göz yardımıyla tekrar değerlendirilip sisteme girilmesi gerektiğinden bahsedilmiştir. Bu yöntemin istenilen başarıyı tam olarak elde edemediği gösterilmiştir (Erdoğan, 2019).

Aynı tez çalışmasında kullanılan bir başka yöntem olan analitik hiyerarşi sürecinde de olumsuz söz öbeklerinin belirlenmesi sonrasında bu söz öbeklerinin metin içerisindeki geçme oranı hesaplanırken insan eline ve gözlemine ihtiyaç duyulmuştur (Erdoğan, 2019).

Bir başka benzer çalışmada da şikâyet mektupları verilerinden faydalanılmıştır. Şikâyet mektuplarında yaşanan yazım yanlışlarından da söz edilmiş ve normalizasyon teknikleri ile giderilmeye çalışılmıştır. Random Forrest Classifier, Naïve Bayes, Logistic Regression, Support Vector Machine, gibi teknikler kullanılarak yapılan 220 binden fazla şikâyet mektubu verisinin işlenmesi sonucu en son elde edilen F1 skorları sırasıyla her bir algoritma için %7, %66, %75 ve %77 olarak elde edilmiştir (Erkaya, 2019). Bu tez çalışması ile işleme sokulan veri sayıları karşılaştırıldığında yaklaşık 20 kat daha fazla verinin makine tarafından işlendiği görülmektedir.

Yukarıda anlatılan çalışmalarda kullanılan veri sayıları ve başarı oranlarına bakıldığında büyük miktarda verinin işleminden geçirildiği görülmektedir. Özellikle ikinci örnekteki 220 bin adet şikâyet mektupları verisinin makine ile sınıflandırılması sonucu elde edilen başarı oranları bu tez çalışması ile kıyaslandığında, yaklaşık 11.236 veri ile elde edilen %65,3'lük başarı skoru, Random Forrest Classifier tekniğinden çok daha başarılı olurken Naïve Bayes tekniği ile elde edilen başarı oranıyla neredeyse aynı başarı

oranına sahip olmuştur. Logistic Regression, Support Vector Machine tekniklerini ise sırasıyla %9,7 ve %11,7 farkla geriden takip etmektedir. Oyunlaştırma tekniği ile sınıflandırılmış veri ile elde edilen başarı ve veri sayısına bakıldığında insan etiketlemesinin başarı oranına ne kadar büyük bir etkisinin olduğu bu karşılaştırmada da görülmektedir.

Bunu destekleyen bir başka örnek ise Google tarafından geliştirilen ve birçok web sitesi ve mobil uygulamanın kullanmış olduğu, Google reCAPTCHA olarak isimlendirilen, sahte kullanıcıları ayırt etmeye yarayan bir sistemdir. Bu sistemde kullanıcılar bir siteye giriş yapacakları zaman kullanıcılara bir obje ile ilgili soru yöneltilir ve soru altında bulunan fotoğraflarda, soruda istenen objenin hangi resimlerde olduğu işaretlenir. Bu işaretlemeler sonucunda kullanıcının sahte olup olmadığı anlaşılır. Aynı zamanda reCAPTCHA aldığı bu geri bildirimler sonucu kendini de geliştirir. Kısaca başarı oranlarının artmasında insan etiketlemelerinin önemi devam etmektedir.

Yukarıda öneminden bahsedilen insan etiketlemelerine yardımcı olan en önemli araçlardan biri de oyunlaştırma tekniğidir. Bu tez çalışmasında verilerin etiketlenmesinde oyunlaştırmanın büyük bir rolü vardır. Etiketleme yapacak kullanıcıları ödül sistemli bir oyunun içine dahil etmek ve oyunu oynamaları konusunda motivasyon kazandırmak, verilerin çok daha hızlı ve doğru toplanmasına yardımcı olmuştur. İleride yapılacak olan çalışmalarda bu tekniğin yanına eklenebilecek başka teknikler ile harmanlanmış daha güçlü veri toplama sistemleri ile devasa veriler çok hızlı bir şekilde işlenebilir.

Sonuç olarak, doğal dil işleme ve makine öğrenme teknikleri günümüzde ve gelecekte, günlük hayatımıza entegre olacak birçok aracın beslendiği başlıca bilimsel alanlar olmayı sürdüreceklerdir. Yapılan literatür taramaları sonucu, dünyada bu alanlara çok büyük bir ilginin olduğu görülmüştür. Bu çalışma Türkçe dili için yapılan çalışmalardan destekle, yeni yapılacak olan çalışmalara ışık tutma amacıyla Türkçe kullanıcı yorumları üzerine çalışılmıştır.

KAYNAKÇA

Albayrak, M. (2017). Bilimsel arařtırmalarda veri madencilięi kullanımı. *International Journal of Social Sciences and Education Research*, 3, 3. Eriřim adresi: <https://dergipark.org.tr/en/download/article-file/261599>

Albayrak, Y., Koer, A., & Uslu, S. (2013). Web servis aracılıęıyla android cihazlardan sıcaklık kontrolü. *Akademik Biliřim Konferansı Bildirileri*, Akdeniz Üniversitesi, Antalya. Eriřim adresi: <https://ab.org.tr/ab13/bildiri/69.pdf>

Alpaydın, E. (2000). Zeki veri madencilięi; Ham veriden altın bilgiye ulaşma yöntemleri. *Biliřim 2000 Eğitim Semineri*, Boęazii Üniversitesi, İstanbul. Eriřim adresi: https://www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2k-notlar.doc

Baykal, A. (2006). Veri madencilięi uygulama alanları. *Dicle Üniversitesi Ziya Gökalp Eğitim Fakóltesi Dergisi*, 7:95-107. Eriřim adresi: <http://www.acarindex.com/dosyalar/makale/acarindex-1423940013.pdf>

Baykasoglu, A. (2005). Veri madencilięi ve imento sektöründe bir uygulama. *Akademik Biliřim '05-VII. Akademik Biliřim Konferansı*, Gaziantep Üniversitesi, Gaziantep. Eriřim adresi: <https://ab.org.tr/ab05/tammetin/171.pdf>

Bozkurt, A., & Kumtepe, E. G. (2014). Oyunlařtırma, oyun felsefesi ve eğitim. *Akademik Biliřim '14- XVI. Akademik Biliřim Konferansı Bildirileri*, 14:147-156. Eriřim adresi: https://www.academia.edu/download/37378201/Oyunlastirma__Oyun_Felsefesi_ve_Egitim-Gamification.pdf

Diri, B. (2014). Makine öğrenmesine giriş (Machine Learning (Machine Learning – ML)). Eriřim adresi: https://www.siskon.com.tr/dosya/PDF/Makale/Makina_Ogrenmesi.pdf

Erdoęmuş, M. (2019). *İnternet ortamındaki müşteri yorumlarının fikir madencilięi ile analiz edilmesine yönelik bir alıřma: Duygu analizi ve sınıflandırma üzerine bir alıřma.* (Yüksek Lisans Tezi). Eriřim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Erkaya, A. E. (2019). *Text classification based on organizational data using machine learning: Makine öğrenmesi ile sınıflandırma çalışması*. (Yüksek Lisans Tezi).

<https://fasttext.cc/docs/en/unsupervised-tutorial.html>. Erişim tarihi: 13 Kasım 2020.

<https://firebase.google.com/products/firestore>. Erişim tarihi: 14 Kasım 2020.

https://java.com/tr/download/help/whatis_java.html. Erişim tarihi: 13 Kasım 2020.

<https://www.jetbrains.com/opensource/kotlin/>. Erişim tarihi: 12 Kasım 2020.

<https://www.postman.com/product/api-client/>. Erişim tarihi: 20 Kasım 2020.

<https://www.sqlite.org/index.html>. Erişim tarihi: 14 Kasım 2020.

<https://github.com/ahmetaa/zemberek-nlp/wiki/Zemberek-NLP-ile-Metin-Sınıflandırma>. Erişim tarihi: 12 Kasım 2020

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valensiya/İspanya. Erişim adresi: <https://arxiv.org/pdf/1607.01759.pdf>

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2017). FastText.zip: compressing text classification models. *International Conference on Learning Representations*, Toulon/Fransa. Erişim adresi: <https://openreview.net/pdf?id=SJc1hL5ee>

Kalikov, A. (2006). *Veri madenciliği ve bir e-ticaret uygulaması: Veri madenciliği üzerine bir uygulama*. (Yüksek Lisans Tezi). Erişim adresi: <https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp>

Meral, M. (2014). Twitter üzerine duygu analizi. *IEEE 22nd Signal Processing and Communications Applications Conference*. Erişim adresi: <https://ieeexplore.ieee.org/abstract/document/6830323>

Onan, A. (2020). Evrişimli sinir ağı mimarilerine dayalı Türkçe duygu analizi. *Avrupa Bilim ve Teknoloji Dergisi*, Özel Sayı:374-380. Erişim adresi: <https://dergipark.org.tr/tr/download/article-file/1240865>

Samuel, S., & Bocutiu, S. (2017). *Programming Kotlin*, Birmingham/UK

Savaş, S., Topaloğlu, N., & Yılmaz, M. (2012). Veri madenciliği ve Türkiye’deki uygulama örnekleri, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 21:1-23. Erişim adresi: <http://acikerisim.ticaret.edu.tr/xmlui/bitstream/handle/11467/685/M00510.pdf>

Vahaplar, A., & İnceoğlu, M. M. (2001). Veri madenciliği ve elektronik ticaret. *Türkiye’de İnternet Konferansları VII’de sunulan bildiri*, Ege Üniversitesi, İzmir. Erişim adresi: <http://inet-tr.org.tr/inetconf7/eposter/inceoglu.doc>

