

T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

# YAPAY ZEKA TEKNİKLERİ KULLANILARAK BECERİ ve YETERLİLİK BELİRLEMEYE DAYALI KARİYER EŞLEŞTİRME

Hilal ERİSEV

DANIŞMAN
Prof. Dr. Ahmet SERTBAŞ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Aralık, 2022

T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

# YÜKSEK LİSANS TEZİ

# YAPAY ZEKA TEKNİKLERİ KULLANILARAK BECERİ ve YETERLİLİK BELİRLEMEYE DAYALI KARİYER EŞLEŞTİRME

**Hilal ERİSEV**

**DANIŞMAN**
**Prof. Dr. Ahmet SERTBAŞ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Aralık, 2022**

# TEZ KABUL VE ONAYI

**Hilal ERİSEV** tarafından, **Prof. Dr. Ahmet SERTBAŞ** danışmanlığında hazırlanan **" Yapay Zeka Teknikleri Kullanılarak Beceri ve Yeterlilik Belirlemeye Dayalı Kariyer Eşleştirme "** başlıklı bu çalışma, jürimiz tarafından **26/12/2022** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

**Tez Jürisi**

| | | **İmza** | **Sonuç** |
|---|---|---|---|
| **DANIŞMAN** | Prof. Dr. Ahmet SERTBAŞ<br>İstanbul Üniversitesi-Cerrahpaşa<br>Bilgisayar Donanımı ve Gömülü Sistemler<br>Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Dr. Öğr. Üyesi M. Erdem İSENKUL<br>İstanbul Üniversitesi-Cerrahpaşa<br>Bilgisayar Donanımı ve Gömülü Sistemleri<br>Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Doç. Dr. Can Eyüpoğlu<br>Milli Savunma Üniversitesi/Hava Harp Okulu<br>Bilgisayar Donanımı Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Dr. Öğr. Üyesi Fatih KELEŞ<br>İstanbul Üniversitesi-Cerrahpaşa<br>Bilgisayar Donanımı ve Gömülü Sistemler<br>Anabilim Dalı | | ☒ Kabul<br>☐ Ret |
| **ÜYE** | Dr. Öğr. Üyesi Zeynep TURGUT<br>Medeniyet Üniversitesi<br>Bilgisayar Donanımı Anabilim Dalı | | ☒ Kabul<br>☐ Ret |

Annem/Babam /Hocalarıma/Aileme ithaf ediyorum…

# BÜTÇE DESTEKLERİ

## YAPAY ZEKA TEKNİKLERİ KULLANILARAK BECERİ ve YETERLİLİK BELİRLEMEYE DAYALI KARİYER EŞLEŞTİRME

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

# TEŞEKKÜR

Bu çalışmanın başlamasından, tamamlanmasına kadar geçen süreçte desteğini ve yardımlarını esirgemeyen, bilgi birikimi ve tecrübesiyle her konuda yol gösteren değerli danışman hocam Prof. Dr. Ahmet SERTBAŞ'a çok teşekkür ederim.

Öğrenim hayatım boyunca beni hiç şüphesiz destekleyen, bana yol gösteren, her zaman arkamda duran ve beni cesaretlendiren babam Ali ERİSEV'e, annem Hasibe ERİSEV'e sonsuz sevgi ve teşekkürlerimi sunarım.

Son olarak da çalışma süresince değerli fikirlerini ve bilgilerini paylaşmaktan çekinmeyen saygıdeğer hocalarıma ve çalışma arkadaşlarıma, ihtiyacım olan her anda maddi ve manevi yanımda olan, beni destekleyen, sorunlarımla başa çıkmama yardımcı olan sevgili arkadaşlarım İlhan Kodalak, Özge Çıplak ve Özkan Doğan'a teşekkürlerimi bir borç bilirim.

Aralık 2022 Hilal ERİSEV

# İÇİNDEKİLER

**Sayfa No**

**TEZ KABUL VE ONAYI** ..... **ii**
**BEYAN** ..... **iii**
**BÜTÇE DESTEKLERİ** ..... **iv**
**TEŞEKKÜR** ..... **v**
**İÇİNDEKİLER** ..... **vi**
**ŞEKİL LİSTESİ** ..... **viii**
**TABLO LİSTESİ** ..... **x**
**SİMGE VE KISALTMA LİSTESİ** ..... **xi**
**ÖZET** ..... **xii**
**ABSTRACT** ..... **xiv**
**1. GİRİŞ** ..... **1**
**2. KAVRAMSAL ÇERÇEVE** ..... **5**
2.1. DOĞAL DİL İŞLEME (NLP) ..... 5
2.2. KELİME TEMSİL YÖNTEMLERİ ..... 5
2.2.1. Terim Frekansı - Ters Belge Frekansı (TF-IDF) ..... 6
2.2.2. Word2Vec Kelime Temsil Yöntemi ..... 7
2.2.3 Doc2Vec Kelime Temsil Yöntemi ..... 9
2.3 MAKİNE ÖĞRENİMİ TEKNİKLERİ ..... 9
2.3.1 Destek Vektör Makineleri (SVM) ..... 11
2.3.2 Rastgele Orman (RF) ..... 12
2.3.4 Lojistik Regresyon (LR) ..... 14
2.4 DERİN ÖĞRENME TEKNİKLERİ ..... 14
2.4.1 Yinelenen Sinir Ağları (RNN) ..... 15
2.4.2 Uzun Kısa Süreli Hafıza (LSTM) ..... 16
2.4.3 Kapı Özyinelemeli Geçitler (GRU) ..... 18
2.4.4 Evrişimsel Sinir Ağları (CNN) ..... 19
2.5 SINIFLANDIRMA MODELLERİNDE PERFORMANS ÖLÇÜTLERİ ..... 20
2.5.1 Hata Matrisi (Confusion Matrix) ..... 20

2.6 YAKE ..... 22
**3. YÖNTEM ..... 24**
3.1 VERİ SETİ ..... 25
3.1.1 Veri Ön İşleme ..... 25
3.2 İŞ İLANLARININ SINIFLANDIRILMASI ..... 27
3.2.1 Metinlerin Sayısal Verilere Dönüştürülmesi ..... 27
3.2.2 Sınıflandırma Problemi için Kullanılan Algoritmalar ..... 27
3.3 YETKİNLİKLERİN ÖNERİLMESİ ..... 28
3.3.1 Yake ile Özellik Çıkarımı ..... 28
**4. BULGULAR ..... 28**
4.1 KELİME TEMSİL YÖNTEMLERİ VE SINIFLANDIRMA MODELLERİN BAŞARILARININ KIYASLANMASI ..... 28
4.2 İŞ İLANLARININ SINIFLANDIRMASI ..... 32
4.3 ÖNERİ SİSTEMİNİN GELİŞTİRİLMESİ ..... 36
**5. TARTIŞMA ..... 40**
**6. SONUÇ VE ÖNERİLER ..... 43**
**KAYNAKLAR ..... 44**
**İNTİHAL RAPORU İLK SAYFASI ..... 50**
**KURUM İZNİ YAZILARI ..... 51**
**ÖZGEÇMİŞ ..... 52**

# ŞEKİL LİSTESİ

**Sayfa No**

**Şekil 2.1:** CBOW ve SG Modeli ((Mikolov ve diğ., 2013). ……..……..……7

**Şekil 2.2:** Skip-gram örneği (Naskar, 2020). …….……..……..8

**Şekil 2.3:** CBOW örneği (Naskar, 2020). …….…..……...8

**Şekil 2.4:** PV-DM Model (Liu ve diğ., 2019). ……..……..9

**Şekil 2.5:** PV-DBOW Model (Liu ve diğ., 2019). ……..……9

**Şekil 2.6:** Makine Öğrenimi (MathWorks, 2022). ……..10

**Şekil 2.7:** Makine öğrenmesi uygulama aşamaları (Security Team, 2018). ……..11

**Şekil 2.8:** SVM yöntemi (Luo, 2021). …….…..……..11

**Şekil 2.9:** Rastgele Orman Modeli (Khan ve diğ., 2021). …….…..……12

**Şekil 2.10:** Belirli bir kullanıcının bilgisayar oyunlarını sevip sevmediğini tahmin etmek için oluşturulan ağaç topluğu modeli (Shaw, 2017). …….……..….13

**Şekil 2.11:** Ağaç algoritmalarının gelişimi (Talun ve diğ., 2020). ……13

**Şekil 2.12:** Lojistik Regresyon (Tutor Team, 2021). ……..14

**Şekil 2.13:** AI, ML ve DL arasındaki ilişki (Security Team, 2018). ……..15

**Şekil 2.14:** Derin öğrenme uygulama süreçleri (Security Team, 2018). ……15

**Şekil 2.15:** RNN mimarisi (Zhang ve diğ. 2018). …….……..16

**Şekil 2.16:** LSTM Modeli (Phi, 2018). …….……..17

**Şekil 2.17:** GRU Modeli (Phi, 2018). …….……19

**Şekil 2.18:** TextCNN Mimarisi (Song ve diğ., 2019). …….…….……..20

**Şekil 2.19:** Hata Matrisi (Demir, 2022). …….……..……...20

**Şekil 3.1:** Sistem mimarisi diyagramı. …….…..……..24

**Şekil 3.2:** Çalışmada kullanılan veri seti örneği. ……..……..25

**Şekil 3.3:** Kategorilere göre verilerin dağılımı. ……..…26

**Şekil 3.4:** Durdurma kelimeleri. ……..……26

**Şekil 4.1:** Model performanslarının doğruluk değerleri. ……..…..….28

**Şekil 4.2:** Model performanslarının f1-skorları. ……...…..29

**Şekil 4.3:** K-fold. …….…...….29

**Şekil 4.4:** K-fold kullanarak eğitilen modellerin performans değerleri. ……..…..….30

**Şekil 4.5:** SVM modelinin hata matrisi. ……32

**Şekil 4.6:** Çevrimiçi platformdan alınan iş ilanı metninden kategori tahmin örneği. …….……..……..33

**Şekil 4.7:** ChatGPT tarafından üretilen Data Engineer özgeçmiş örneğinden kategori tahmini. .……...33

**Şekil 4.8:** Benzer kategorilere ait özelliklerin venn şeması ile gösterimi. ……34

**Şekil 4.9:** Backend Developer alanında çalışan bir kişinin özgeçmiş örneği ve kategori tahmini. ….....34

**Şekil 4.10:** Frontend Developer alanında çalışan bir kişinin özgeçmiş örneği ve kategori tahmini. ....34

**Şekil 4.11:** 3 farklı kategoriye ait belirgin yetkinliklerin venn şeması gösterimi. ……35

**Şekil 4.12:** Özgeçmiş örneği ve tahmin edilen kategorileri. ……...…….…….36

**Şekil 4.13:** Özgeçmişten çıkarılan öznitelikler. ……..……37

**Şekil 4.14:** Data Engineer ile ilgili yetkinlikler…….…...…37

**Şekil 4.15:** Önerilen yetkinlikler. ……...38

**Şekil 4.16:** Özgeçmiş ve tahmin edilen kategorisi. ……38

**Şekil 4.17:** Android kategorisine ait yetkinlikler. ……. 39

**Şekil 4.18:** Özgeçmişten çıkarılan öznitelikler. ..…….……..39

**Şekil 4.19:** Özgeçmişte eksik olan yetkinliklerin önerilmesi. …….…….……..39

# TABLO LİSTESİ

**Sayfa No**

**Tablo 4.1:** TF-IDF ile elde edilen skorlar. ……30

**Tablo 4.2:** Word2Vec ile elde edilen skorlar. ……..……..31

**Tablo 4.3:** Doc2Vec ile elde edilen skorlar. ……..….31

**Tablo 5.1:** İlgili çalışmalar. ……42

# SİMGE VE KISALTMA LİSTESİ

| Kısaltmalar | Açıklama |
|---|---|
| **NLP** | : Doğal Dil İşleme (Natural Language Process) |
| **NLTK** | : Doğal Dil Araç Takımı (Natural Language Toolkit) |
| **SVM** | : Destek Vektör Makineleri (Support Vector Machine) |
| **RF** | : Rastgele Orman (Random Forest) |
| **LR** | : Lojistik Regresyon (Logistic Regression) |
| **XGBoost** | : Aşırı Gradyan Arttırma (Extreme Gradient Boosting) |
| **RNN** | : Yinelemeli Sinir Ağı (Recurrent Neural Network) |
| **LSTM** | : Uzun Kısa Süreli Hafıza (Long-Short Term Memory) |
| **GRU** | : Kapı Özyinelemeli Geçitler (Gated Recurrent Unit) |
| **CNN** | : Evrişimsel Sinir Ağı (Evrişimsel Sinir Ağı) |
| **ML** | : Makine Öğrenimi (Machine Learning) |
| **AI** | : Yapay Zeka (Artificial Intelligence) |
| **DL** | : Derin Öğrenme (Deep Learning) |
| **CBOW** | : Sürekli Kelime Torbası (Continuous Bag of Word) |
| **SG** | : Gram Atlama (Skip-Gram) |
| **PV-DM** | **:** Paragraf Vektörlerinin Dağıtılmış Bellek Modeli (Distributed Memory Model of Paragraph Vectors) |
| **TF-IDF** | **:** Terim Frekansı-Ters Belge Frekansı (Term Frequency-Inverse Document Frequency) |
| **CV** | : Özgeçmiş (Curriculum Vitae) |

# ÖZET

## YÜKSEK LİSANS TEZİ

### YAPAY ZEKA TEKNİKLERİ KULLANILARAK BECERİ ve YETERLİLİK BELİRLEMEYE DAYALI KARİYER EŞLEŞTİRME

**Hilal ERİSEV**

**İstanbul Üniversitesi-Cerrahpaşa**

**Lisansüstü Eğitim Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Danışman : Prof. Dr. Ahmet SERTBAŞ**

İnternetin kullanımı gün geçtikçe yaygınlaşmaktadır. Bu sayede gelişen teknoloji ile beraber bilgiye ulaşma hızı, isteklerimiz ve beklentilerimiz de artmaktadır. Yapay Zeka'nın hayatımıza girmesiyle birlikte de birçok alanda işimizi kolaylaştıran ve otomatikleşen makineler ortaya çıkmaya başlamıştır. Çeşitli alanlarda kullanılan farklı Yapay Zeka yöntemleri bulunmaktadır. Nasıl ki robotlar insanlara fiziksel olarak yardımcı olan birer makine ise karar vermemize de yardımcı olacak sistemler bulunmaktadır. Bu çalışmada bilişim sektöründeki ilanlar dikkate alınarak bir karar destek sisteminin geliştirilmesi amaçlanmıştır. Yapay Zeka'nın alt yöntemlerinden biri olan Doğal Dil İşleme teknikleri kullanılarak internet ortamından toplanan iş ilanlarının sınıflandırılması ve ilanların başlıca yetkinliklerinin önerilmesi sağlanmıştır. İlanların sınıflandırılması için Makine Öğrenimi ve Derin Öğrenme algoritmalarından yararlanılmıştır. En yüksek performans ölçütüne sahip algoritma seçilmiştir. Veri seti ve problem tanımına en uygun algoritma seçilirken literatür taraması sonucunda metin sınıflandırmada kullanılan başlıca algoritmalar belirlenmiştir. Makine Öğrenimi algoritmalarından; Destek Vektör Makineleri, Rastgele Orman, Lojistik Regresyon, Aşırı Gradyan Arttırma: Derin Öğrenme algoritmalarından; Uzun Kısa Süreli Hafıza, Kapı Özyinelemeli Geçitler ve Evrişimsel Sinir Ağı kullanılmıştır. Bu algoritmalar, 3 farklı kelime vektörleştirme tekniği ile beraber kullanılarak performansları değerlendirilmiştir. Bunun sonucunda en iyi skoru üreten algoritmanın Destek Vektör Makineleri olduğu gözlemlenmiştir. Bu işlemler sonucunda herhangi bir iş tanımının hangi alana ait olduğunu ya da yüklenen özgeçmişe sahip kişinin en yakın hangi

alanlarda çalışabileceği listelenmiştir. İstenilen çalışma alanına göre de ilgili alanda sahip olunması gereken yetkilikler önerilmiştir.

Aralık 2022 , 70 sayfa.

**Anahtar kelimeler:** Makine Öğrenimi, Derin Öğrenme, Doğal Dil İşleme, Yetkinlik Önerisi

# ABSTRACT

## M.Sc. THESIS

**CAREER MATCHING BASED on DETERMINING SKILLS and COMPETENCIES USING ARTIFICIAL INTELLIGENCE TECHNIQUES**

**Hilal ERİSEV**

**İstanbul University-Cerrahpaşa**

**Institute of Graduate Studies**

**Department of Computer Engineering**

**Computer Engineering Program**

**Supervisor : Prof. Dr. Ahmet SERTBAŞ**

The use of the Internet is becoming more and more widespread day by day. In this way, with the developing technology, the speed of accessing information, our requests and expectations are also increasing. With the introduction of Artificial Intelligence into our lives, machines that facilitate and automate our work in many areas have begun to emerge. There are different Artificial Intelligence methods used in various fields. Just as robots are machines that help people physically, there are systems to help us make decisions. In this study, it is aimed to develop a decision support system by considering the advertisements in the informatics sector. By using Natural Language Processing techniques, one of the sub-methods of Artificial Intelligence, job postings collected from the internet are classified and the main competencies of the postings are suggested. Machine Learning and Deep Learning algorithms were used to classify the advertisements. The algorithm with the highest performance criterion was selected. While choosing the most suitable algorithm for the data set and problem definition, the main algorithms used in text classification were determined as a result of the literature review.

From Machine Learning Algorithms; SVM, Random Forest, Logistic Regression, XGBoost: Deep Learning algorithms; LSTM, GRU and CNN are used. These algorithms were used together with 3 different word vectorization techniques to evaluate their performance. As a result, it was observed that the algorithm producing the best score was SVM. As a result of these processes, it is listed to which field any job description belongs to or in which fields the person with the uploaded CV can work closest. According to the desired field of study, the competencies to be possessed in the relevant field are suggested.

December 2022, 70 pages.

**Keywords:** Machine Learning, Deep Learning, Natural Language Processing, Competence Proposal

# 1. GİRİŞ

Son yıllarda internete bağlı olarak gelişen teknoloji ve insanların hayal gücü ile birlikte makinelerin yapabilecekleri neredeyse sınırsız hale gelmiştir. Günümüzde insanlığın ve makinelerin birbiriyle olan bu yakın ilişkisinin temelleri ilk olarak Allan Turing'in "Makineler düşünebilir mi?" sorusuyla atılmış oldu. 1956'da da John McCarty'nin "Yapay Zeka" terimini ortaya çıkarmasıyla her geçen gün gelişen ve birçok alanda insanlara yardımcı olan makineler geliştirilmeye başlandı. Aklımıza ilk gelen robotların yapımı dışında da kullanılan Yapay Zeka birçok veriyi işleyebilmeyi ve onlardan bir değer yaratabilmeyi de sağlıyor. Yapay Zeka teriminin altında yapılmak istenene bağlı olarak kullanılabilen birçok yöntem bulunmaktadır. Bunlardan bir tanesi NLP tekniğidir. İnsanlar kendilerini ifade etmek için kullandıkları dili bilgisayarların anlayabileceği makine dillerine dönüştürülmesini sağlayan bu yöntem ses ve metin verileri bilgisayarlar için anlamlı hale getirilebilmektedir. Bunun sonucunda da dil çevirileri, metin sınıflandırmaları, yazım denetimi ve chatbotlar gibi uygulamalar gerçekleştirilmektedir. Bilgin ve Şentürk (2017) Twitter üzerinden kullanıcılardan alınan geri bildirim metinlerini kullanarak şirketlerin ürün ve hizmetlerini iyileştirmelerine yardımcı olmak amacıyla duygu analizi yapmışlardır. Çalışmada Doc2Vec kullanılarak Türkçe ve İngilizce Twitter mesajlarını işlemişler ve Türkçe metinlerde en yüksek başarıyı %46 doğruluk değeri ile PV-DM kullanarak elde etmişlerdir. İngilizce metinlerde ise en yüksek başarıyı %64 doğruluk değerini yine PV-DM kullanarak elde etmişlerdir. Çelik ve Koç (2021) çalışmalarında farklı vektörleştirme tekniklerini ve sınıflandırma algoritmalarını kullanarak haber metinlerini sınıflandırmışlardır. Çalışmada 6 kategoride haber metinlerinden yararlanılmış, en yüksek başarı oranı %95 ile FastText vektörleştirme tekniği ve SVM algoritmasının kullanılmasıyla elde etmişlerdir. Tran ve diğ. (2021) iş tanımı metinlerinden ilgili iş unvanlarını tahmin etmek için çok etiketli sınıflandırma yaklaşımı önermişlerdir. Bi-GRULSTM-CNN algoritmalarını iş unvanları tahmin problemine uygulamak için önceden eğitilmiş farklı dil modelleri kullanmışlardır. Kelime gömme katmanı önceden eğitilmiş BERT modeline dayalı olarak oluşturulmuştur. Çok dilli önceden eğitilmiş modele sahip BERT, geliştirme setinde %62,20 ve test setinde %47,44 olan hem geliştirme hem de test setlerinde Fl-skor ile en yüksek sonucu almıştır.

Son yıllarda neredeyse her alanda kullanılmaya başlanan Yapay Zeka teknikleri çevrimiçi uygulamalar aracılığıyla iş verenlerin en uygun adayları bulabilmesine, iş arayanların da niteliklerine göre başvurabilecekleri iş ilanlarının önerilmesine yardımcı olması amacıyla sıklıkla başvurulan bir yöntem haline gelmiştir. Horat ve diğ. (2019) adayla işvereni en hızlı ve en verimli şekilde bir araya getirilebilmesi amacıyla mülakat görüşme sonuçlarının raporlanmasını sağlayan Yapay Zeka görüşme sistemini sunmuşlardır. Kariyer.net' te yayınlanan ilanları kullanarak NLP ve metin madenciliği yöntemleri ile daha farklı pozisyonlar için gerekli olan görev tanımı ve yetkinlik listelerini çıkarmışlardır. Çıkarılan öznitelikleri kullanarak eğitilen makine öğrenmesi modellerinde en yüksek başarı oranını %84 ile Lojistik Regresyon algoritmasını kullanarak elde etmişlerdir. Kolli ve diğ. (2021) internet üzerindeki sahte ilanlarının tespiti için de NLP tekniklerinden ve Makine Öğrenimi algoritmalarından yararlanmışlardır. Kişisel bilgilerin bulunduğu bir ortamda öneriler genellikle profilden alınan bilgilerle birlikte özelleştirilebilmektedir. Birçok görüşe göre bu öneriler sadece teknik yeteneklere bakılmaksızın yapıldığı için daha doğru olduğu düşünülmektedir. Yi ve diğ. (2016) kullanıcı benzerliklerini ve iş pozisyonlarının benzerliklerini ölçen optimize edilmiş işbirlikçi filtreleme yöntemini sunmuşlardır. İş tanımını, iş pozisyonlarına başvuru yapan kullanıcıların benzerliklerini ve kullanıcıların geçmişte yaptıkları pozisyonların benzerliklerini de dikkate alan model ile temel öneri algoritmasına göre daha iyi sonuçlar elde etmişlerdir. 9 ilan önerisi yapan sistemin doğruluğunu değerlendirmek için Kesinlik (Precision), Geri Çağırma (Recall) ve F ölçüm yöntemlerinden faydalanmışlardır. İşbirlikçi filtreleme yönteminin kullanıldığı diğer bir çalışmada, Patel ve diğ. (2017) kullanıcı profilini dikkate alan, içerik tabanlı filtreleme ve öğe tabanlı filtreleme yöntemlerini kullanarak bir yetkinlik öneri sistemi geliştirmişlerdir. Diğer çalışmadan farkı kullanıcıların geçmişte başvurdukları ilanların ya da iş ilanlarına daha önce başvuru yapan kişilerin benzerlikleri dikkate alınmamıştır. Çalışmada San Jose Eyalet Üniversitesi'nin kariyer merkezinden alınan veriler ile iş ilanlarındaki yetkinlikler çıkarılmış ve kullanıcı profilleri ile ikili karşılaştırma sonucunda eksik yetkinlikleri önerilmiştir. Ayrıca SJSU Bilgisayar Mühendisliği bölümü öğrencilerinden toplanan özgeçmişlerden elde edilen veriler de kullanılmıştır. Kullanıcıların profillerinin benzerlikleri K-en yakın komşu algoritması ile belirlenmiş, bu benzerlik doğrultusunda ortak olmayan becerileri diğer kullanıcılara önermişlerdir. İçerik tabanlı ve öğe tabanlı filtreleme olmak üzere iki yöntemin başarısı ayrı ayrı değerlendirilmiş ve içerik tabanlı filtrelemede Öklid Mesafesi ile %73, öğe tabanlı filtrelemelerde ise Pearson Korelasyonu ile %60 başarı oranı elde etmişlerdir. Herhangi bir

profil oluşturmadan özgeçmişlerden ve iş ilanlarından özellik çıkarımı yapan birçok çalışma da mevcuttur. Lin ve Lei (2016) özgeçmişleri ve iş ilanlarını eşleştirmeyi amaçlayan çalışmalarında, iş tavsiye sistemi oyunundan alınan açık kaynak bir veri setinde bulunan özgeçmişleri kullanmışlardır. Farklı makine öğrenmesi ve derin öğrenme algoritmaları kullanarak maaş, yaş cinsiyet, pozisyon gibi birçok özelliği en kısa sürede XGBoost algoritması ile elde etmişlerdir. Değerlendirme ölçeği olarak Kesinlik (Precision) kullanmışlar ve bu ölçeğe göre en yakın çıkarımları, geliştirdikleri IBagging yöntemi ile elde etmişlerdir. Geliştirdikleri yöntem ile özellikleri göre pozisyon önerisi yapmayı amaçlayan bu çalışmada, en doğru önerileri 4 pozisyon önerisinde sağlamışlardır. Pozisyon önerisinde değerlendirme ölçeği olarak Geri Çağırma (Recall) ölçüm yöntemi kullanmışlar ve geliştirdikleri IBagging yöntem ile pozisyon önerisinde %78 oranında başarı sağlamışlardır. Bu çalışmada kişisel verilere başvurulmadan Bilişim sektöründeki alanlara yönelik sahip olunması gereken teknik özelliklerin önerilmesi amaçlanmıştır. Böylece çalışmak istediği alana karar veren bir kişi herhangi bir profil oluşturmadan, kişisel bilgilerini girmek mecburiyetinde kalmadan istediği alan ile ilgili hangi yetkinliklere sahip olması gerektiğini öğrenebilmektedir. Bunun haricinde özgeçmiş üzerinden öznitelik çıkarımı yapılarak, özgeçmişe göre çalışabileceği alanların da belirlenmesi sonucunda o alanla ilgili bilmesi gereken ama özgeçmişte bulunmayan yetkinler önerilmiştir. Çalışmada açık kaynak veri setlerinin bulunduğu Kaggle üzerinden toplanan ve çevrimiçi iş ilanı sitesinden alınan gerçek verilerin birleşiminden elde edilen veri seti kullanılmaktadır. İş tanımları bir makine öğrenimi modeli olan SVM algoritması ile sınıflandırılmaktadır. Model 17 kategorilik bir veri seti üzerinde %91 oranında başarı sağlamıştır. Çalışmanın ikinci bölümü olarak düşünebileceğimiz öneri kısmında da öncelikle özgeçmiş SVM algoritması ile sınıflandırılmaktadır. Sınıflandırılan özgeçmişten ayrı bir derin öğrenme modeli olan LSTM modeli ile belirleyici olabilecek yetkinlikler çıkarılmaktadır. Önceden belirlenen özgeçmişin kategorisi dikkate alınarak da bu kategoriye ait sahip olunması gereken yetkinlikler özgeçmiş sahibine önerilmektedir. Bilişim sektörünü kapsayan bu çalışma daha sonra istenildiği takdirde farklı alanlar için de kullanılabilir. Çalışma, bu alana yeni başlayan kişileri hedeflese de birçok yere entegre edilebilir. İş ilanlarının bulunduğu çevrimiçi bir sitede her gün birçok ilan paylaşılmaktadır. Bu ilanların hangi kategorilerde yayınlanacağı da otomatikleştirilebilir. Elde bulunan iş tanımı metnine göre hangi kategori altında yayınlanacağı belirlenebilir, önemli özellikler öne çıkarılabilir. 2. bölümde çalışmada kullanılan yöntemlerle alakalı geniş kapsamlı bilgiler verilecek ve bu yöntemlerle ilgili literatür

incelemesi sonucunda yapılan çalışmalardan bahsedilecektir. 3. bölümde çalışmada kullanıla, veri seti, araçlar ve yöntemler açıklanacaktır. 4. bölümde çalışmada edinilen bulgulardan bahsedilecektir. 5. bölümde ise literatürdeki benzer çalışmalar ile bu çalışmada edinilen sonuçlar tartışmalı bir şekilde karşılaştırılacaktır. 6. bölümde ise çalışmanın sonucu ve gelecek çalışmalarda nasıl geliştirilebileceğinden bahsedilecektir.

# 2. KAVRAMSAL ÇERÇEVE

İş tanımları metinlerden oluşmaktadır ve metin sınıflandırmada kullanılan farklı Makine Öğrenimi ve Derin Öğrenme teknikleri bulunmaktadır. Bu bölümde, çalışmada kullanılan yöntemler açıklanmış ve literatürdeki benzer yöntemlerle yapılan çalışmalar incelenmiştir.

## 2.1. DOĞAL DİL İŞLEME (NLP)

Bir problemi çözümlemek için öğrenme algoritmaları kullanılmaktadır. Problemin girdileri bilgisayarın anlayabileceği dilde olması gerekmektedir. Sayısal verilerden oluşan bir veri setini dönüştürmek için bir şey yapılmasına gerek yoktur. Ancak veriler insanların birbirini anlayabilmek için kullandığı dillerde, metin olarak temsil ediliyorsa bu verilerin bilgisayarın anlayabileceği hale dönüştürülmesi gerekmektedir. NLP burada devreye girmektedir. Birçok dil için NLP yöntemleri, kütüphaneleri bulunmaktadır. Uygulama alanları, yararlı bilgilerin çıkarılması, diller arasında metin çevirisi yapılması, yazılı eserlerin özetlenmesi, soruların otomatik olarak yanıtlanması, belgelerin sınıflandırılması ve kümelenmesi gibi konuları içerir (Otter ve diğ. 2021). Bu yöntemler istenmeyen kelimelerin kaldırılması, metinlerin noktalama işaretleri ve sembollerden arındırılması ve bilgisayarların anlayabileceği şekilde sayılara dönüştürülmesi gibi uygulamalar da olabilmektedir. Bu çalışmada ingilizce metinler kullanıldığı için Python'da bulunan NLTK kütüphane ile metin ön işleme işlemleri gerçekleştirilmiştir. Doğal dil araç takımı anlamına gelen NLTK ile metinler kelimelere bölünebilir, kelimenin ekleri kaldırılıp kökleri bulunabilir, metinler noktalama işaretlerinden arındırılabilir. Ayrıca NLTK kütüphanesinin Konuşma Parçası Etkileme (Part-of-Speech Tagging) modülü, kelimeleri cümlelerde geçtiği konuma göre sıfat, isim,zarf ve fiil olarak olarak etiketleyebilmektedir.

## 2.2. KELİME TEMSİL YÖNTEMLERİ

Kelime gömme (word embedding) diye anılan kelime temsili metin verilerinin makineler tarafından anlaşılabilecek hale getirilmesidir. Kelime temsil yöntemleri, kelimeleri veya cümleleri sayısal vektörler haline getiren yöntemlerdir.

### 2.2.1. Terim Frekansı - Ters Belge Frekansı (TF-IDF)

TF-IDF, belirli bir belgede kelimelerin göreceli sıklığını, bu kelimenin tüm veri seti üzerindeki tersine oranına göre belirleyerek çalışır. Sezgisel olarak, bu hesaplama ile hedef kelimenin belirli bir belge ile ne kadar alakalı olduğunu tespit eder (Çelik ve Koç, 2021). TF yani terim frekansı, Hans Peter Luhn (1957), IDF yani ters belge frekansı ise Karen Spärck Jones (1972)'un çalışmalarıyla ortaya çıkan metriklerdir. TF, ilgili kelimenin belgede kaç kez geçtiğini belirtir. Kelimenin belgede geçme sayısı, belgedeki toplam kelime sayısına bölünerek bulunur. IDF ise toplam belge sayısının, ilgili kelimenin geçtiği belge sayısına bölünmesiyle elde edilir. IDF, ilgili kelimenin o belge için ne kadar önemli olduğunu belirten metriktir. Değer 0'a ne kadar yakınsa o kelimenin birçok belgede geçtiği anlamına gelmektedir. 1'e yaklaştıkça belgelerde daha nadir görülen bir kelime olduğu anlamını taşımaktadır. TF-IDF ile ilgili denklemler sırasıyla 2.1, 2.2 ve 2.3'te verilmiştir (Kınık ve Güran 2021).

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, D) \tag{2.1}$$

$$TF(t, d) = \log(1 + frekans(t, d)) \tag{2.2}$$

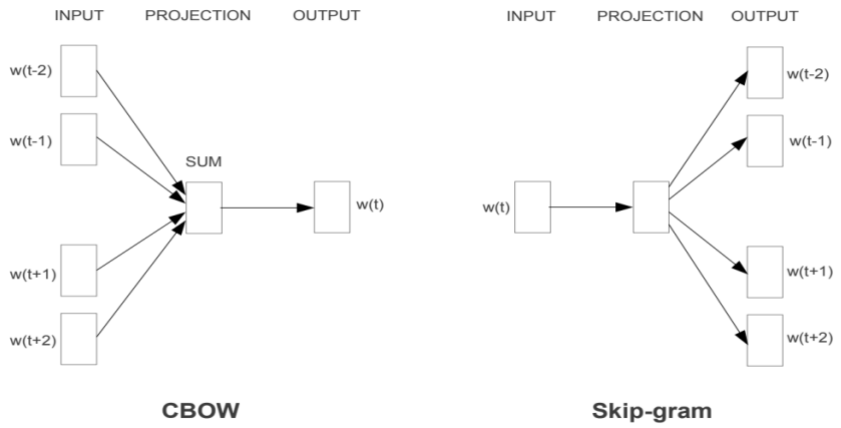
$$IDF(t, D) = \log\left(\frac{N}{Say(d \in D : t \in d)}\right) \tag{2.3}$$

Burada frekans(t,d) hedef kelime t'nin d.belgede kaç kez geçtiğini belirtir; say(d∈D:t∈d) ifadesi ise hedef kelime t'nin D derlemini oluşturan belgelerden kaç tanesinde bulunduğu bilgisidir (Kınık ve Güran 2021). Dadgar ve diğ. (2016) 2 adet veri setinde bulunan haber metinlerini sınıflandırmışlardır, BBC ve 20Newsgroup veri setlerinde kullanıldığı çalışmada ön işlemeden geçirilen metinler TF-IDF yöntemi ile vektörleştirilmiş ve SVM ile sınıflandırmışlardır. Her iki veri seti için de 5 konu başlığı bulunmaktadır ve BBC veri seti için performans değeri %97,84 başarı oranı olarak ölçülmüş, 20Newsgroup veri seti için ise %94,93 başarı oranı elde edilmiştir. Kelimelerin vektör uzayında temsil edilmesini sağlayan TF-IDF, en yaygın kullanılan geleneksel yöntemlerdendir. Literatür incelemesi sonucunda yüksek başarı oranlarının sağlandığı çalışmalarda kullanıldığı görülmüş, bu çalışma için sinir ağı tabanlı Word2Vec ve Doc2Vec yöntemleri ile karşılaştırıldığında da daha iyi sonuçlar elde edilmesi sebebiyle metinlerin vektör uzayında temsili için TF-IDF yöntemi kullanılmıştır.

### 2.2.2. Word2Vec Kelime Temsil Yöntemi

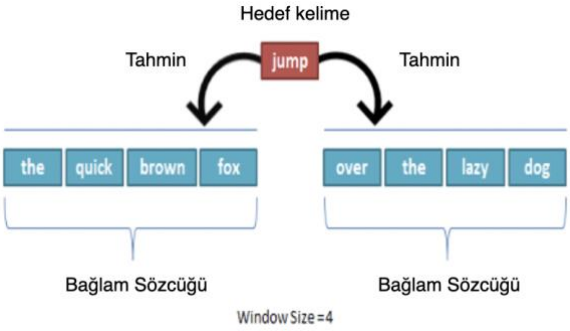
Word2Vec, tahmin tabanlı kelime temsil yöntemidir. Bir girdi, bir çıktı ve gizli katmandan oluşan gözetimsiz sinir ağı modelidir. Mikolov ve diğ., (2013) iki yeni model mimarisi önermişlerdir. Word2Vec, Skip-Gram ve CBOW olarak iki farklı yöntem ile gerçekleştirilir. Bu iki yaklaşım girdi ve çıktı değişkenlerinin farklı uygulanmasına dayanır ancak temel olarak aynı sinir ağını kullanırlar. (Di Gennaro ve diğ., 2021).

- CBOW modeli, merkez kelimenin etrafındaki kelimelerden merkezi kelimeyi tahmin etmeye çalışır. CBOW modeli küçük veri setlerinde daha iyi tahmin yapmaktadır. Öte yandan birden fazla anlamı olan kelimeleri anlamakta zorlanır. Model mimarisi Şekil 2.1'de gösterilmektedir (Mikolov ve diğ., 2013).
- Skip-gram modeli, bir kelimenin etrafında bulunan kelimeleri tahmin etmeye çalışır. Skip-gram modeli daha büyük verilerde daha iyi çalışmaktadır. Aynı zamanda Skip-gram modeli CBOW'a göre 2 veya daha fazla anlamı olan kelimeleri anlamakta daha başarılıdır. Model mimarisi Şekil 2.1'de gösterilmektedir.

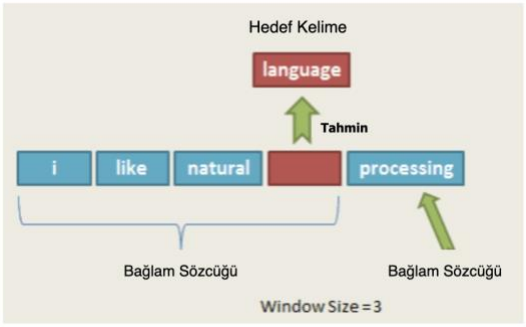



**Şekil 2.1:** CBOW ve SG Modeli (Mikolov ve diğ., 2013).

Skip-Gram ve CBOW modellerinin nasıl çalıştığı, sırasıyla Şekil 2.2 ve Şekil 2.3'te İngilizce cümle örneği üzerinde gösterilmektedir (Naskar, 2020).




**Şekil 2.2:** Skip-gram örneği.




**Şekil 2.3:** CBOW örneği.

Word2Vec eğitim aşamasında belirtilmesi gereken bazı parametreleri vardır. Bu parametreler sırasıyla aşağıda verilmiştir:

**Sentences**: Model girdiyi liste olarak alır,

**Size**: Kelime vektörünün boyutu,

**Window**: Cümle mevcut olan ve tahmin edilen kelimelerin arasındaki maksimum mesafe değeri,

**Min_count**: Belirli bir sayıdaki kelimelerin elenmesi,

**Workers**: Çalışacak Thread sayısı,

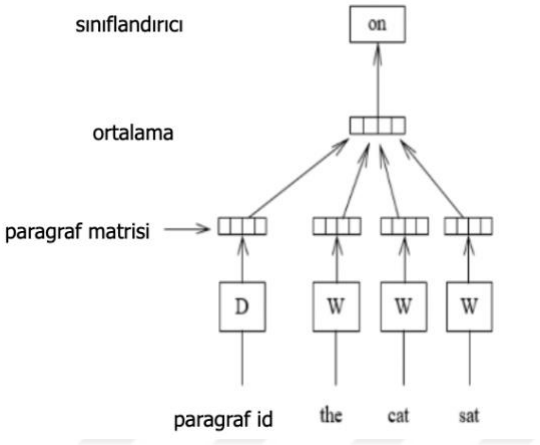
**Sg**: Skip-gram yöntemi için 1, CBOW yöntemi için 0 değeri kullanılır,

**Alpha**: Başlangıçta modelin öğrenme oranıdır (Opuz, 2019).

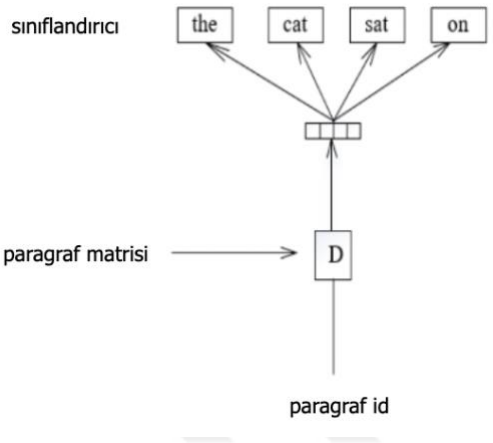
Long ve Yanqing (2015) Word2Vec kelime temsil yöntemini kullanarak büyük veri üzerinde kelime vektörleri oluşturmuşlardır. Çalışmada Word2Vec modeli ile eğitim veri seti hazırlanmış ve eğitilen model sonucunda kelimelerin benzerlikleri değerlendirilmiştir. Benzer sözcükleri bir araya toplayarak oluşturulan kümeleri yeni bir veri boyutuna sığdırılması ile veri boyutunu küçültmeyi hedeflemişlerdir. Wang ve diğ. (2018) 3 ayrı veri setinde bulunan kullanıcı yorumlarından duygu analizi gerçekleştirmişlerdir. 3 ayrı veri seti de Word2Vec kelime gömme yöntemi ile temsil edilmişlerdir. Her veri seti için 3 ayrı sınıflandırıcı modeli kullanmışlar ve en iyi performans gösteren modeli LSTM olarak belirlemişlerdir. Bu çalışmada da Word2Vec kelime temsil yönteminden faydalanılmıştır. İş ilanlarının tanım metinleri Word2Vec modeli ile eğitilmiş, vektör uzayında temsil edilmiştir. Diğer kelime temsil yöntemleriyle aynı sınıflandırma algoritmaları kullanılarak modellerin performanslarıyla karşılaştırılmış, çalışmada kullanılan veri seti için TF-IDF yöntemine göre daha az başarı gösterdiği için son aşamada tercih edilmemiştir.

### 2.2.3 Doc2Vec Kelime Temsil Yöntemi

Doc2Vec modeli, Word2Vec mimarisini kullanarak her bir belge için ayrı bir belge vektörünün eklenmesi ile yaratılmıştır. Bu belge vektörleri ile derlemdeki belgeler uzunluklarından bağımsız bir şekilde vektör uzayında sayısal değerlerle temsil edilmektedirler. (Kınık ve Güran 2021). Doc2Vec modeli iki yaklaşımla eğitilebilir: Paragraf Vektörünün Dağıtılmış Bellek sürümü (PV-DM) ve Paragraf Vektörünün Dağıtılmış Kelime Çantası sürümü (PV-DBOW). Şekil 2.4 ve Şekil 2.5, sırasıyla PV-DM ve PV-DBOW çerçevesini göstermektedir (Liu ve diğ. 2019).




**Şekil 2.4:** PV-DM Model.




**Şekil 2.5:** PV-DBOW Model.

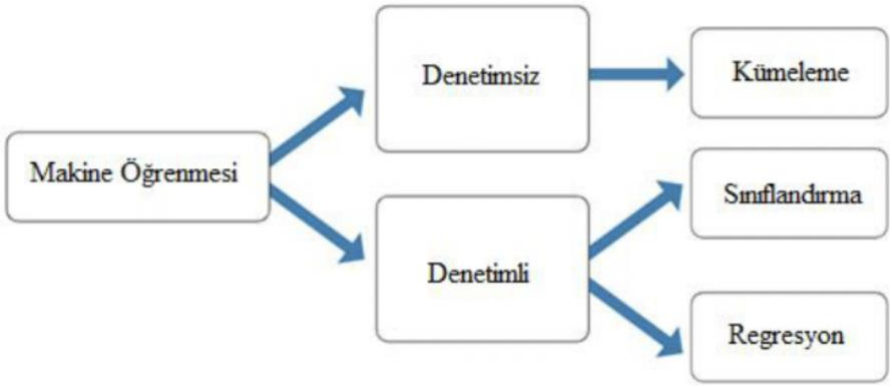
Doc2Vec modelindeki PV-DM, Word2Vec modelindeki CBOW'a karşılık gelmektedir, PV-DBOW yöntemi ise Skip-gram yöntemi olarak uygulanmaktadır.

Chang ve diğ. (2018) anormal yorumları belirlemek için, kümeleme algoritması kullanarak duygu analizi yöntemlerini uygulamışlardır. Kelimeleri vektör uzayında temsil etmek için de Doc2Vec modelinden yararlanmışlardır.

## 2.3 MAKİNE ÖĞRENİMİ TEKNİKLERİ

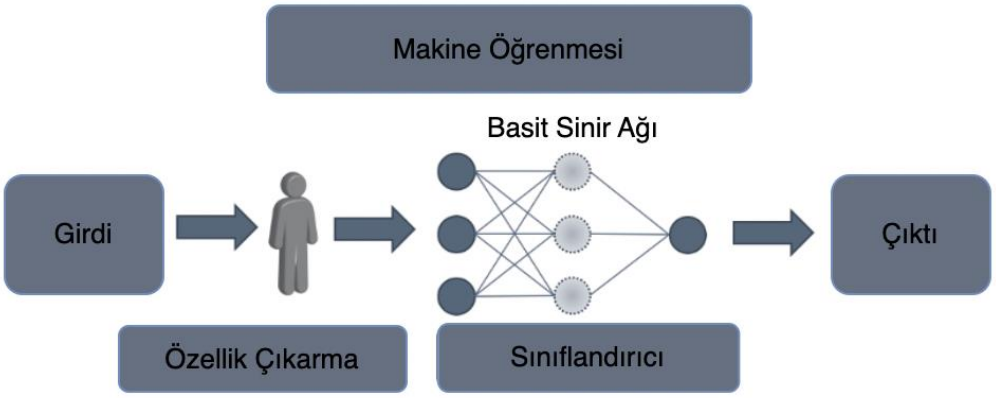
Makine öğrenimi ilk olarak 1943'te Walter Pitts ve Warren McCulloch tarafından tanıtıldı. McCulloch, insan beyninde ve sinir ağlarında karar verme sürecini haritalandıran matematiksel bir makale yayınlayan sinirbilimcidir. Makalede, beyindeki her nöron basit bir dijital işlemciye, beyin ise bir bilgi işlem makinesine benzetildi. 1950'de ise matematikçi ve bilgisayar bilimcisi Alan Turing, Turing testini tanıttı. Yapay Zeka'nın bir alt dalı olan makine öğreniminin ortaya

çıkışı ise 1980'lere dayanmaktadır (Gopani, 2022). Makine öğrenimi modellerinde öğrenme yöntemleri denetimli öğrenme ve denetimsiz öğrenme olarak ikiye ayrılmaktadır. Denetimsiz öğrenmede veriler etiketlenmemiştir. Model verilerini keşfederek bir sonuca varır. Denetimli öğrenmede ise veriler etiketlenmiştir. Hem girdi hem de çıktı verileri belirtilmiştir. Makine öğrenimi modellerinin kullanım alanları çok geniştir. Regresyon, sınıflandırma gibi problemlerde sıklıkla kullanılmaktadır. Şekil 2.6'da makine öğreniminde kullanılan öğrenme yöntemleri ve kullanılabilecekleri problem kapsamları verilmiştir.




**Şekil 2.6:** Makine Öğrenimi (Sakal, M., 2020).

Regresyon, denetimli öğrenme yaklaşımlarındandır. Sürekli değişkenleri modelleme ve tahminlime için kullanabilir. Doğrusal regresyon algoritmasının uygulama örnekleri, gayrimenkul fiyatının tahmini, satış tahmini, borsadaki hisse senedi fiyat hareketlerinin tahmini olabilir. Regresyonda, etiketli veri kümeleri var ve çıktı değişkeni, girdi değişkenleri tarafından belirlenir. Bu denetimli öğrenme yaklaşımıdır. En basit regresyon şekli, veri kümesine düz bir çizgi (düz hiper düzlem) sığdırılmaya çalışılan ve veri kümesinin değişkenleri arasındaki ilişkinin doğrusal olduğu zaman kullanılabilen bir yöntemdir (Ray, 2019). Sınıflandırma teknikleri, 2 ya da daha fazla kategoriye sahip verilerin çıktısını tahmin eden bir yöntemdir. Örneğin bir e-postanın gerçek mi yoksa sahte mi olduğunu veya bir tümörün kanserli mi yoksa iyi huylu mu olduğunu bulmada kullanılabilmektedir. Sınıflandırma modelleri, girdi verilerini kategori olarak sınıflandırır (MathWorks, 2022). Şekil 2.7'de Makine Öğreniminin uygulanma süreci verilmiştir.

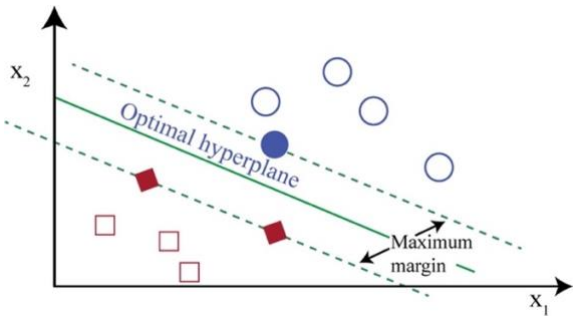



**Şekil 2.7:** Makine Öğrenimi uygulama aşamaları (Security Team, 2018).

Bu çalışmada metinlerin sınıflandırılması için farklı Makine Öğrenme algoritmaları kullanılmıştır. Çalışmada kullanılan veriler etiketlenmiş olduğu için öğrenme yöntemi olarak denetimli öğrenme kullanılmaktadır.

### 2.3.1 Destek Vektör Makineleri (SVM)

Destek vektör makineleri, ilk olarak Joachims tarafından metin sınıflandırması için kullanılan bir sınıflandırma tekniğidir. SVM, en düşük yapısal risk ilkesine dayanan güçlü ve denetimli bir öğrenme örneğidir. Algoritma eğitim sırasında, pozitif ve negatif örnekleri ayırmak için bir hiper düzlem oluşturur. Oluşturulan hiper düzlemde her örneğin nereye yerleştirilmesi gerektiğini belirterek yeni örnekleri sınıflandırır (Dadgar ve diğ. 2016). Şekil 2.8'de SVM mimarisi işleyişi gösterilmiştir.




**Şekil 2.8:** SVM yöntemi (Luo, 2021).

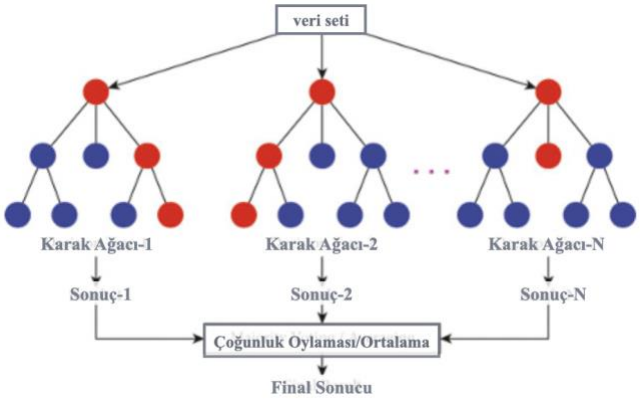
SVM, hem doğrusal sınıflandırma hem de doğrusal olmayan sınıflandırma gerçekleştirir. Doğrusal olmayan sınıflandırma, çekirdek işlevi kullanılarak uygulanır. Doğrusal olmayan

sınıflandırmada, çekirdekler homojen polinom, karmaşık polinom, Gauss radyal temel fonksiyonu ve hiperbolik tanjant fonksiyonu olarak tanımlanır (Vani, 2021).

Hajibabaee ve diğ. (2022), Twitter'da atılan tweetlerde saldırgan bir dil kullanılıp kullanılmadığının tespit etmek için SVM algoritmasını kullanmışlardır. TF-IDF, Word2Vec ve FastText olmak üzere 3 farklı kelime temsil yöntemini 8 farklı Makine öğrenme algoritması ile kullanmışlardır. 2 sınıflı bu problem için en yüksek başarı oranını %95 ile TF-IDF ve SVM algoritmasının birlikte kullanılmasıyla elde etmişlerdir.

### 2.3.2 Rastgele Orman (RF)

2001 yılında Breiman tarafından tanıtılan ([Breiman, 2001) Rastgele Orman modeli birçok sınıflandırma ve regresyon problemleri için kullanılmaya başlanmış ve başarılı sonuçlar verdiği gözlemlenmiştir. Yapısal olarak Karar Ağaçları modelini kullanan Rastgele Orman, Karar Ağaçlarının aşırı öğrenme (overfitting) problemine bir çözüm getirmiştir. Veri setinden rastgele gruplar seçerek yüzlerce karar ağacı oluşturuluyor ve her bir karar ağacı kendi tahminini üretiyor. Eğer problem regresyonsa, Karar Ağaçları tahminlerin ortalamasını alır, problem sınıflandırmaysa tahminler arasında en çok oy alanı seçilir. Şekil 2.9'da Rastgele Orman modeli gösterilmiştir.

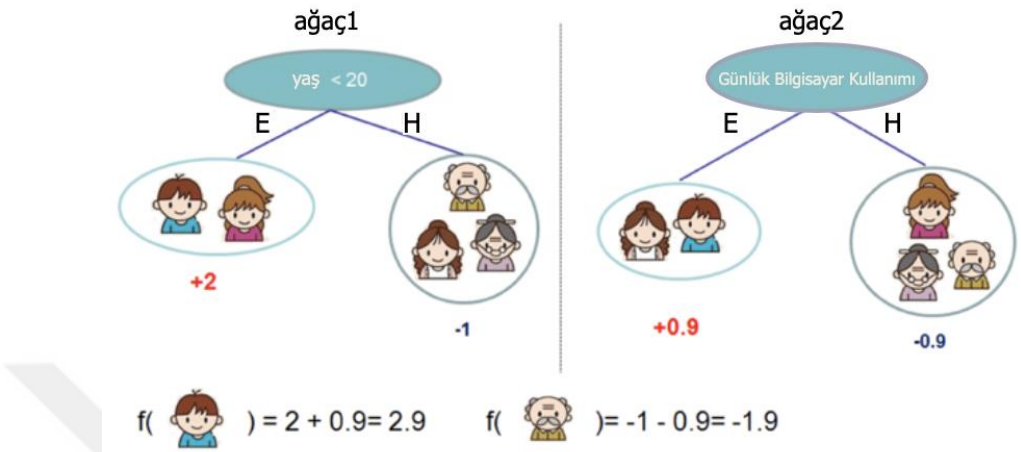



**Şekil 2.9:** Rastgele Orman Modeli (Khan ve diğ., 2021).

### 2.3.3 Aşırı Gradyan Arttırma (XGBoost)

XGBoost, temelinde ağaç yapısını uygulayan ve Gradyan Arttırma (Gradient Boosting) , algoritmasının belirli parametreleri geliştirerek optimize edilmesiyle ortaya çıkmış bir algoritmadır. XGBoost'u anlamak için önce ağaç yapısını ve Gradyan Arttırma algoritmasını anlamak gerekir. Gradyan Arttırma, çıktıyı tahmin etmek için her adımda mevcut işleve yeni

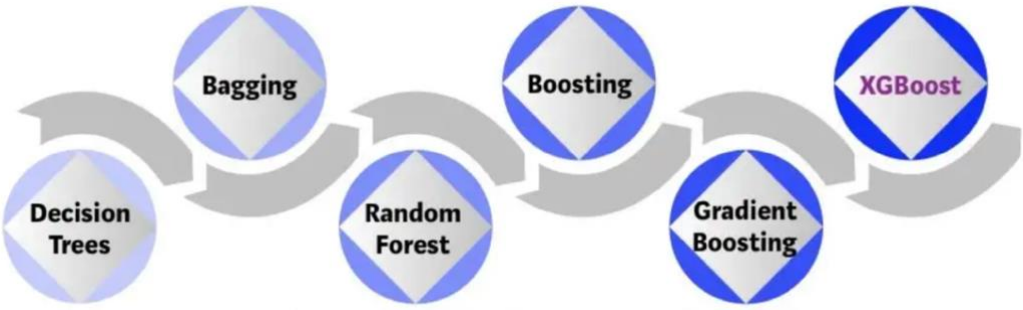
bir işlev ekler ve sonucu başlangıçtan tamamen farklı bir işlevdir. Gradyan Arttırma Modelleri işlemleri sırasıyla yaparken XGBoost, tüm işlemleri paralel şekilde uygulamaktadır. Şekil 2.10 Gradyan Arttırma Modellerinin ağaç yapısını göstermektedir.




**Şekil 2.10:** Belirli bir kullanıcının bilgisayar oyunlarını sevip sevmediğini tahmin etmek için oluşturulan ağaç topluğu modeli (Shaw, 2017).

(+2), (+0.1), (-1), (+0.9), (-0.9) her yapraktaki tahmin puanlarıdır. Belirli bir kullanıcı için son tahmin, her ağaçtan gelen tahminlerin toplamıdır (Shaw, 2017).

Ağaç tabanlı algoritmaların günümüze kadar gelişimi Şekil 2.8'de gösterilmiştir (Morde, 2019). Ağaç tabanlı diğer algoritmalardan ayrılan ve son zamanlarda çeşitli problemde sıklıkla kullanılmasını sağlayan bazı özellikleri, regülarizasyon, budama, boş değerler ile çalışılabilesi ve sistem optimizasyonudur. Algoritma Chen ve Guastrin tarafından 2016 yılında tanıtılmıştır. Chen ve Guastrin (2016) yayınladıkları makalede, mevcut sistemlerden çok daha az kaynak kullanarak milyarlarca örneğin ötesine ölçeklenebilir olduğunu ve diğer popüler algoritmalardan çok daha hızlı çalışabildiğini söylemektedirler.

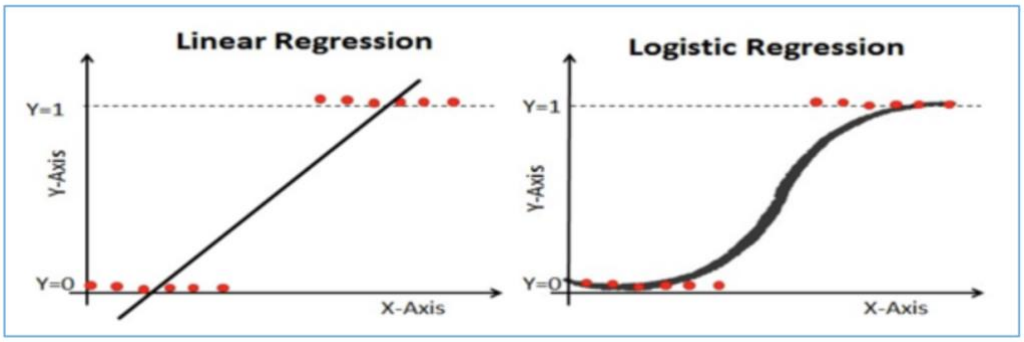



**Şekil 2.11:** Ağaç algoritmalarının gelişimi (Talun ve diğ., 2020).

Çevrimiçi iş ilanı sitelerinden gelen metinlerin içeriğini sınıflandırmak için iki model kullanmışlardır. FastText ve XGBoost modellerinin her ikisi de deneysel aşamada %88 başarı oranı elde etmişlerdir. Kurniawanda ve diğ. (2022) Instagram'dan alınan yorumlarda siber zorbalık yapılıp yapılmadığını tespit eden bir yöntem önermişlerdir. Yorumları TF-IDF kelime temsil yöntemi ile vektörleştirmişler ve bu kelime vektörlerini kullanarak XGBoost ile sınıflandırmışlardır. Birlikte kullanılan yöntemler sonucunda performans değerini f1-skor ile ölçülmüş ve %78 başarı oranı elde etmişlerdir.

### 2.3.4 Lojistik Regresyon (LR)

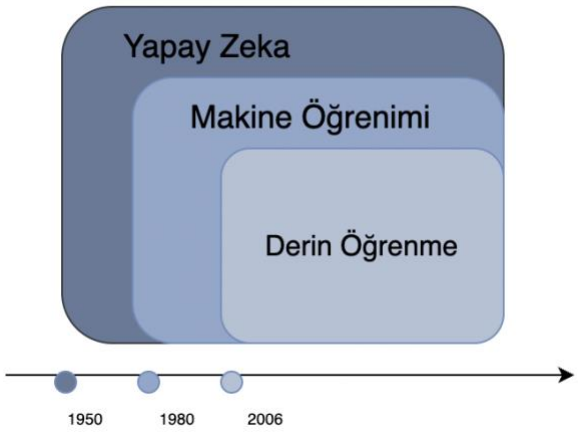
1958'de istatistikçi David Cox tarafından tanıtıldı (Kamran ve diğ. 2019). Lojistik regresyonun, lineer regresyon ile arasındaki en büyük fark iki sınıfı birbirinden ayıracak çizgiyi nasıl uyguladığıdır. Lineer regresyon, optimum çizgiyi çizmek için "En Küçük Kareler Yöntemi" (Least Squares) kullanırken, lojistik regresyon "Maksimum Olabilirlik" (Maximum Likelihood) kullanır. Lojistik regresyon, sınıflandırma yapmak için Sigmoid (Lojistik) fonksiyonu kullanır, Sigmoid fonksiyonu "S" şeklinde bir eğridir. Şekil 2.12'de Lojistik Regresyon ve Lineer Regresyon modeli gösterilmiştir.




**Şekil 2.12:** Lojistik Regresyon (Tutor Team, 2021).

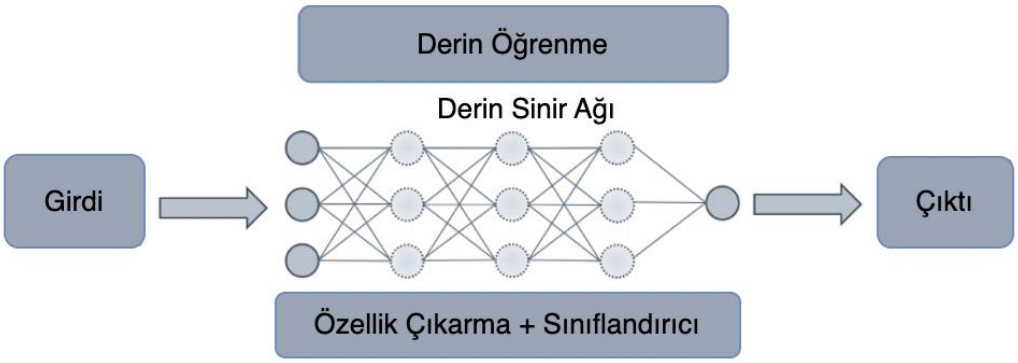
## 2.4 DERİN ÖĞRENME TEKNİKLERİ

Makine Öğreniminin Yapay Zeka'nın bir alt dalı olduğundan bahsetmiştik. Derin Öğrenme de Makine Öğreniminin bir alt dalı olarak ortaya çıkmıştır. Şekil 2.13'te aralarındaki ilişki gösterilmiştir.




**Şekil 2.13:** Yapay Zeka, Makine Öğrenimi ve Derin Öğrenme ilişkisi (Security Team, 2018).

Makine Öğrenimi yöntemleri tek katmanda gerçekleştirilirken, derin öğrenme ise çok katmanlı sinir ağlarından oluşmaktadır. Derin Öğrenme, geri yayılım algoritmasını kullanarak bir makinenin önceki katmanındaki çıktıdan her katmandaki çıktıyı hesaplamak için sistem parametrelerinin nasıl değiştirilmesi gerektiğini bulur. Bu şekilde büyük veri kümelerindeki karmaşık yapıyı keşfeder (LeCun ve diğ., 2015).



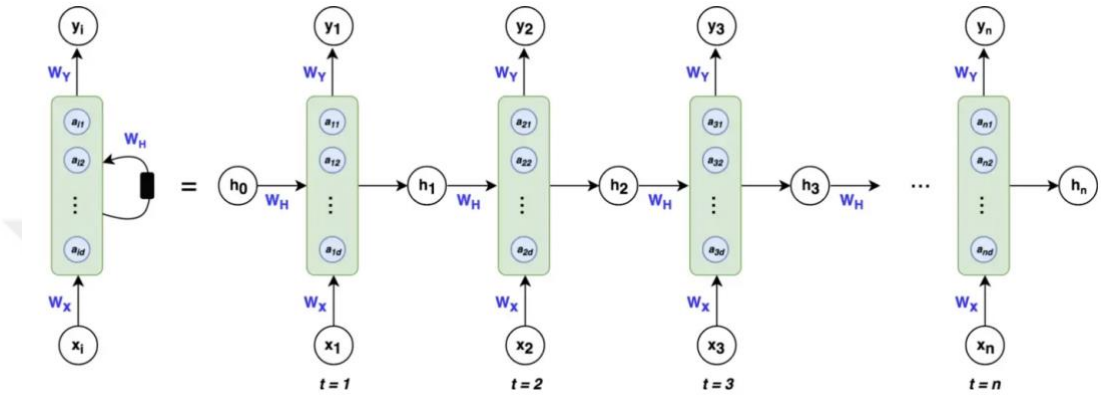

**Şekil 2.14:** Derin öğrenme uygulama süreçleri (Security Team, 2018).

Bu bölümde Derin Öğrenme algoritmalarından RNN, LSTM, GRU ve CNN açıklanmıştır.

### 2.4.1 Yinelenen Sinir Ağları (RNN)

Elman tarafından 1990'da ortaya çıkarılan tekrarlayan sinir ağı, nöronlar arasındaki bağlantıları yönlendirilmiş bir döngüye dönüştüren sinir ağları sınıfıdır. Şekil 2.15'te gösterildiği gibi önceki adımdan çıkan sonuç bir sonraki adımı beslemektedir (Khuong, 2019). İleri beslemeli

sinir ağlarından farklı olarak RNN, bir dizi girişi işlemek için dahili belleği kullanabilir, bu da onu sıralı bilgilerin işlenmesi gereken problem için kullanışlı kılar. RNN'nin bir hafızası vardır. Buradaki hafıza, her öğesi için aynı görevi gerçekleştirdiği ve her çıktının önceki tüm hesaplamalara bağlı olduğu anlamına gelir. RNN bu hafıza sayesinde şu ana kadar işlenenlerle ilgili bilgileri hatırlamış olur.




**Şekil 2.15:** RNN mimarisi (Zhang ve diğ. 2018).

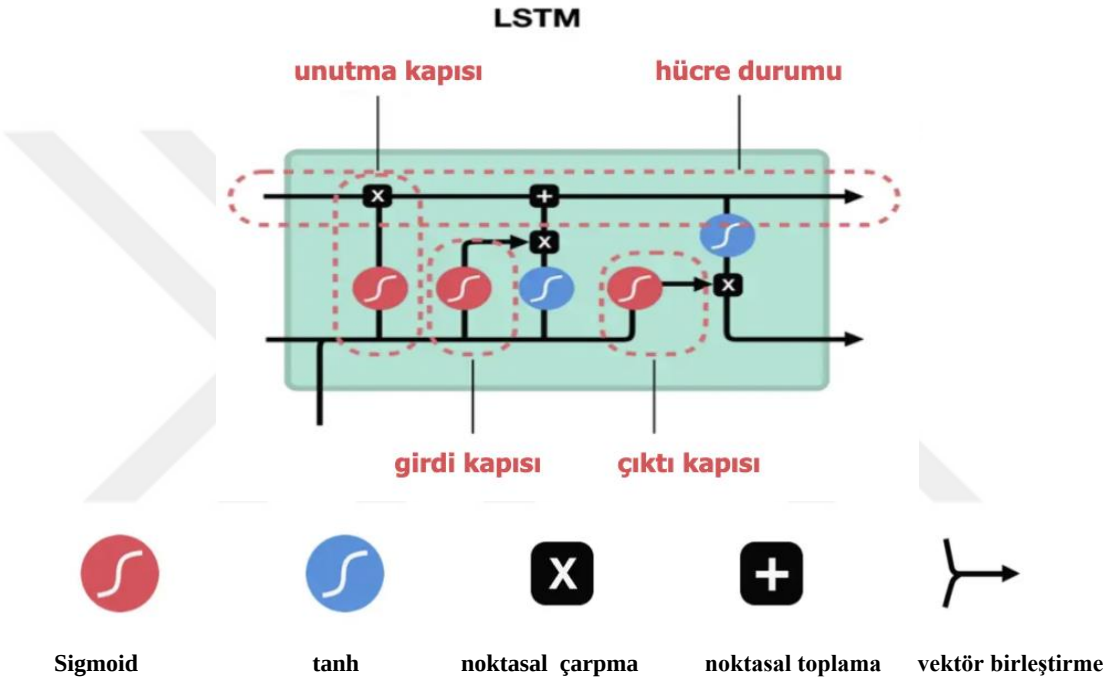
RNN mimarisi açıklayacak olursak, t zamanında görüldüğü gibi, mimari, önceki gizli katmandan h'yi ve t zamanındaki x girişini dahil ederek t-1'de kullanır. Bu, ağın sırayla geçerli girdinin arkasında bulunan önceki girdilerden gelen bilgileri kullanarak bir sonraki katmana akrarır. Sıfırıncı h vektörünün her zaman 0'ların bir vektörü olarak başlayacağına dikkat etmek önemlidir, çünkü algoritma dizideki ilk öğeden önce gelen hiçbir bilgiye sahip değildir (Khuong, 2019).

### 2.4.2 Uzun Kısa Süreli Hafıza (LSTM)

Sinir ağları ileri besleme işi bittiğinde geri yayılım yaparak parametreleri optimize etmektedir. Her bir yinelenen katmanda ht'den ht-1'e geri yayılım yapılırken W matrisi ile çarpma işlemi uygulanır. Tüm ağ boyunca kullanılan W matrisi aynı olduğu için W matrisi sürekli kendisi ile çarpılmaktadır. Eğer büyük bir ağ içerisindeysek çarpım değeri katlanarak artacaktır. Bu durum artan gradyan patlaması olarak adlandırılır. Kullanılan aktivasyon fonksiyonuna göre W matrisindeki değerler 0 ile 1 arasındaki değerlerden oluşuyorsa bu defa her çarpımda elde edeceğimiz çarpım değeri daha da küçük hale gelecektir. Bu durum ise gradyan kaybolması olarak adlandırılır (Çelikten, 2020). Değerlerin çok büyük ya da küçük olması katmanların

yeteri kadar öğrenememesine sebep olur. Gradyan patlaması ve gradyan kaybolması problemlerine çözüm olarak ise LSTM ve GRU algoritmaları ortaya çıkmıştır.

Hochreiter ve Schmidhuber (1997) tarafından tanıtılan LSTM'in yukarıdaki problemlere çözümü ise modele yeni kapılar eklemektir. Bu kapılar ile hangi bilgilerin unutulacağına ya da hangi bilgilerin hatırlanacağına karar vermektedir. Hatırlanması gereken önemli bilgiler bir sonraki aşamaya aktarılır. Bu işlem Hücre (Cell) ve Kapı (Gate) state ile gerçekleşir. Şekil 2.16'da LSTM modelinin yapısı gösterilmiştir.




**Şekil 2.16:** LSTM Modeli (Phi, 2018).

Bilgilerin unutulmasını veya hatırlanıp sonraki katmanlara taşınmasını sağlayan kapılar, sigmoid fonksiyonunu kullanılır. Sigmoid fonksiyonunda değerler 0 ile 1 arasına sıkıştırılır. Bu fonksiyon, verileri güncellemek veya unutmak için aracıdır. Çünkü 0 ile çarpılan herhangi bir sayı 0'dır, bu da değerlerin unutulması anlamına gelmektedir. 1 ile çarpılan herhangi bir sayı değişmez dolayısıyla bu değer tutulur.

- **Unutma kapısı (Forgat Gate)**, hangi bilgilerin atılması veya saklanması gerektiğine karar verir. Önceki gizli durumdan gelen bilgiler ve geçerli girdiden gelen bilgiler sigmoid işlevinden geçirilir. 0 ile 1 arasına sıkıştırılan değerlerin, 0'a yaklaştıkça unutulacağı, 1'e yaklaştıkça da saklanacağı anlamına geliyor.

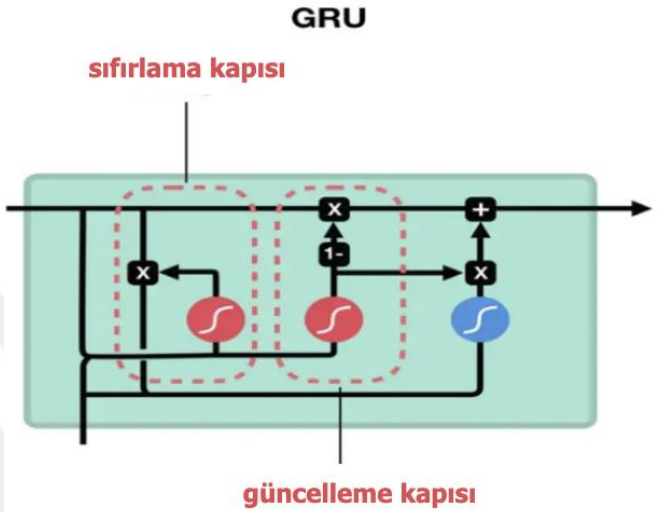
- **Giriş Kapısı (Input Gate),** hücre durumunu (cell state) güncellemek için kullanılır. Sigmoid fonksiyonu ile değerler 0-1 arasında olacak şekilde dönüştürülerek hangi değerlerin güncelleneceğine karar verir. Ağdaki değerleri düzenlemek için tanh fonksiyonu ile değerler -1,1 arasına getirilir. Tanh çıktısı, Sigmoid çıktısı ile çarpılır.
- **Hücre Durumu (Cell State),** bilgileri hücreler arasında taşımakla görevlidir. Unutma kapısından gelen vektör ile önceki katmandan gelen sonuç çarpılır, girdi kapısından gelen değer ile toplanır. Sonuc bize yeni hücre durumunu verir (Akça, 2021).
- **Çıkış Kapısı (Output Gate),** bir sonraki gizli durumun ne olması gerektiğine karar verir. Gizli durum önceki girişler hakkında bilgi içermektedir. Gizli durum, tahminler için de kullanılır. Önceki gizli durum ve mevcut giriş bir sigmoid fonksiyonuna gönderilir. Ardından yeni değiştirilen hücre durumu tanh fonksiyonuna aktarılır. Gizli durumun hangi bilgileri taşıması gerektiğine karar vermek için tanh çıktısını sigmoid çıktısıyla çarpılır. Çıktı gizli durumdur. Yeni hücre durumu ve yeni gizli durum daha sonra bir sonraki zaman adımına taşınır (Phi, 2018).

Semberecki ve Maciejewski (2017), 7 kategori bulunan bir Wikipedia makaleleri veri setini kullanarak belgelerin sınıflandırılmasında hangi yöntemin daha iyi sonuç vereceğini öğrenmek için farklı teknikler kullanmışlardır. Word2vec kelime temsil yöntemi ve basit vektörleştirme yöntemlerini LSTM ve CNN ile birlikte kullanıldığında, hangi yöntemlerin daha yüksek başarı sağladığını karşılaştırmalı bir şekilde gözlemlemişlerdir. LSTM ve Word2Vec birlikte kullanıldığında diğer yöntemlerden daha iyi sonuç vererek, %86 oranında başarı elde etmişlerdir.

### 2.4.3 Kapı Özyinelemeli Geçitler (GRU)

GRU yinelen sinir ağlarını kullanan yeni nesil algoritmalardan biridir. GRU, daha az tensor işlemi barındırdığı için LSTM'den daha hızlı çalışırlar. LSTM modeline çok benzer yapıdadır. LSTM'den farkı ise hücre durumu (cell state) modelden çıkarılmasıdır. Hücre durumu LSTM modelinde bilgi aktarımını sağlıyordu, burada bilgi aktarımı gizli durum tarafından yapılmaktadır. Modeldeki diğer bir fark da yalnızca iki kapının bulunmasıdır. Bu kapılar Sıfırlama Kapısı (Reset Gate) ve Güncelleme Kapısıdır (Update Gate). Şekil 2.15'te GRU model yapısı gösterilmiştir (Phi, 2018).

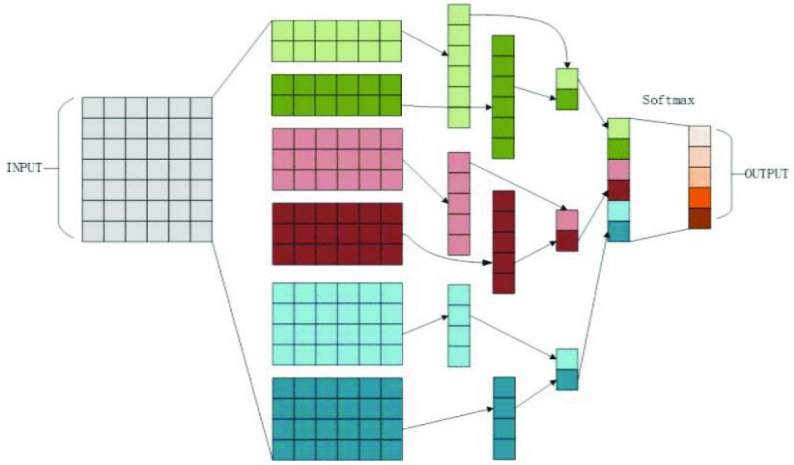
- **Güncelleme Kapısı (Update Gate)**, LSTM'nin unutma kapısı ve giriş kapısına benzer işleri yapar. Hangi bilgilerin atılacağına ve hangi yeni bilgilerin ekleneceğine karar veren kapıdır.
- **Sıfırlama Kapısı (Reset Gate)**, önceki katmanlardan gelen bilgilerin ne kadarının unutulacağına karar veren kapıdır.




**Şekil 2.17:** GRU Modeli (Phi, 2018).

### 2.4.4 Evrişimsel Sinir Ağları (CNN)

CNN görüntü işleme alanında sıklıkla kullanılmaktadır. Başarılı sonuçlar elde edilmesi sebebiyle sadece görüntü verileri değil başka veriler ile birlikte de kullanılmaya başlanmıştır. CNN, bir tür tipik yapay sinir ağıdır. Bu tür bir ağda, her katmanın çıktısı bir sonraki nöron katmanının girdisi olarak kullanılır. Çok katmanlı evrişim işlemi, çıktı katmanına kadar her katmanın sonuçlarını doğrusal olmayan şekilde dönüştürmek için kullanılır. Metin analizinde kullanılan evrişimli sinir ağı modeli, dört bölümden oluşur: gömme katmanı, evrişimli katman, havuzlama katmanı ve tamamen bağlı katman. TextCNN modeli, CNN'nin NLP alanında kullanılan yöntemidir (Wang ve Qu, 2017). TextCNN modelinde, n-gram'a benzer metinlerden öznitelikleri çıkarabilmek için farklı boyutlarda filtreler ayarlanır ve kelime vektörlerinden oluşan matris, evrişim katmanı ve havuzlama katmanı aracılığıyla işlenir. Son olarak Softmax işlevi aracılığıyla çıktılar sınıflandırılır. TextCNN'nin yapısı Şekil 2.18'de gösterilmiştir.




**Şekil 2.18:** CNN Mimarisi (Song ve diğ., 2019).

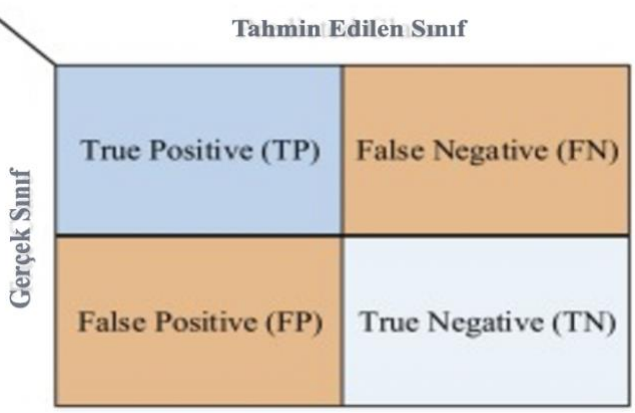
Cai ve diğ., (2018) tek etiketli, çok sınıflı haber metinlerinin sınıflandırılması için derin öğrenme algoritmalarından yararlanmışlardır. CNN ve LSTM kullanarak sınıflandırma sonuçlarını karşılaştırmışlar ve CNN ile %85 başarı oranı elde etmişlerdir.

## 2.5 SINIFLANDIRMA MODELLERİNDE PERFORMANS ÖLÇÜTLERİ

Bir problemin çözümüze uygun birden çok yöntem bulunabilir. Sınıflandırma algoritmalarında da durum böyledir. Doğru modeli seçmek için bir kritere ihtiyaç duyulmaktadır. Modelin başarısını değerlendirmek için birçok performans ölçeği bulunmaktadır.

### 2.5.1 Hata Matrisi (Confusion Matrix)

Hata matrisi, çıktının 2 veya daha fazla sınıfa ait olabileceği sınıflandırma modellerinin performans ölçütü olarak kullanılır. Tahmin edilen ve gerçek değerlerin 4 ayrı kombinasyonundan oluşur. Şekil 2.19'da 2 sınıflı bir model için gösterilen hata matrisidir.




**Şekil 2.19:** Hata Matrisi (Demir, 2022).

“TN” çıktısı, doğru şekilde sınıflandırılan negatif örneklerin sayısını gösteren doğru-negatif anlamına gelir. Benzer şekilde “TP”, doğru olarak sınıflandırılan pozitif örneklerin sayısını gösteren doğru-pozitif anlamına gelir. "FP" terimi, yanlış olarak sınıflandırılan pozitif örneklerin sayısını gösteren yanlış-pozitif anlamına gelir ve "FN", yanlış olarak sınıflandırılan negatif örneklerin sayısını gösteren yanlış-negatif anlamına gelir. Sınıflandırma problemlerinde en sık kullanılan ölçütlerden biri doğruluk (accuracy) ölçütüdür. Bir modelin doğruluğu denklem (2.4)’te verilen formül (Demir, 2022) kullanılarak hesaplanır.

$$\textit{Doğruluk (Accuracy)} = \frac{TN+TP}{TN+FP+FN+TP} \tag{2.4}$$

Doğruluk, dengesiz veri kümeleriyle kullanıldığında yanıltıcı olabilir. Bu nedenle performansı değerlendirmek için faydalı olabilecek hata matrisine dayalı başka ölçütler vardır. Python'da “sklearn” kütüphanesinin bir parçası olan “confusion_matrix()” fonksiyonu kullanılarak hata matrisi elde edilebilmektedir. Bu işlev, "sklearn.metrics'ten import Constraint_matrix" kullanılarak Python'da kullanılabilir. Hata matrisini elde etmek için fonksiyona gerçek değerleri ve tahmin edilen değerleri verilir (Kulkarni ve diğ., 2020). Modellerin başarısını ölçülen yaygın olarak kullanılan diğer ölçütler Kesinlik (Precision) ve Geri çağırma (Recall) ölçütleridir. Sırasıyla denklem (2.5) ve (2.6)’da Kesinlik (Precision) ve Geri Çağırma (Recall) formülleri (Kulkarni ve diğ., 2020) verilmiştir.

$$\textit{Kesinlik (Precision)} = \frac{TP}{TP+FP} \tag{2.5}$$

$$\textit{Geri Çağırma (Recall)} = \frac{TP}{TP+FN} \tag{2.6}$$

F-skor, bir sınıflandırıcının hem kesinlik puanını hem de hatırlama puanını kullanan F-değeri olarak da bilinir. F-değeri, sınıflandırma problemlerinde sıkça kullanılan başka bir metriktir. F-ölçüsü, kesinlik ve geri çağırma arasında ağırlıklı bir harmonik ortalama kullanılarak hesaplanır (Kulkarni ve diğ., 2020). Denklem (2.7)’de F-ölçüsü metriğinin yaygın olarak kullanılan f1-skor ölçütünün formülü (Kulkarni ve diğ., 2020) verilmiştir.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (2.7)$$

Bu çalışmada da kullanılan modellerin performanslarını ölçmek için hata matrisi, doğruluk, kesinlik, geri çağırma ve f1-skor ölçütleri kullanılmıştır.

## 2.6 YAKE

Yake, bir metnin en alakalı anahtar kelimelerini seçmek için tek bir belgeden çıkarılan istatistiksel metin özelliklerine dayanan, hafif, denetimsiz, otomatik bir anahtar kelime çıkarma yöntemidir (Campos ve diğ., 2020). Yake algoritmasının ilk adımında metinler ön işlemelerden geçirilir. Sonraki adımda bir dizi istatistiksel yöntemlerin birleştirilmesi ile kelimelerin ağırlık değerleri hesaplanır. Bir terimin büyük/küçük harf kullanımı, anahtar kelimelerin çıkarılması söz konusu olduğunda önemli bir özelliktir. Kelimenin tüm harflerinin büyük olduğu durumlar kısaltma olarak kabul edilip, anahtar kelime çıkarımında dikkate alınmaktadır. Denklem (2.8)'de bir aday terimin büyük/küçük harf durumunun sayısal olarak etkisi hesaplanmaktadır. Terimin önemini gösteren başka bir göstergesi de konum olarak kabul edilmektedir. Denklem (2.9)'da $\mathrm{Sen_t}$ ilgili terimin geçtiği cümlelerin konumunu ifade etmektedir. Belgedeki t terim sıklığını gösteren denklem (2.10)'da gösterilmiştir. Denklem (2.12), aday terim t ile beraber kullanılan farklı terimlerin sayısı arttıkça, t teriminin o kadar az önemli olacağını gösterir. Denklem (2.13), bir aday terimin farklı cümlelerde ne sıklıkta göründüğünü ölçer. SF(t), aday terim t'nin cümlede geçme sıklığı ve #Sentences metindeki toplam cümle sayısıdır. Anahtar kelimenin ağırlık puanı hesaplamak için, tüm özellik ağırlıkları benzersiz bir S(t) puanında toplanır, denklem (2.14)'te gösterilmiştir. Değer ne kadar küçükse, bi-gramlık terim (t) o kadar anlamlıdır. Denkle (Campos ve diğ., 2020).

$$T_{Case} = \frac{max(TF(U(t)), TF(A(t)))}{ln(TF(t))} \quad (2.8)$$

$$T_{Position} = ln(ln\,(3 + Median\,(Sen_t))) \quad (2.9)$$

$$TF_{Norm} = \frac{TF(t)}{MeanTF + 1 * \sigma} \quad (2.10)$$

$$DL\lfloor DR\rfloor = \frac{|A_t \ , \ w|}{\sum k \ \epsilon A_t \ w \quad CoOccur_t \quad k} \tag{2.11}$$

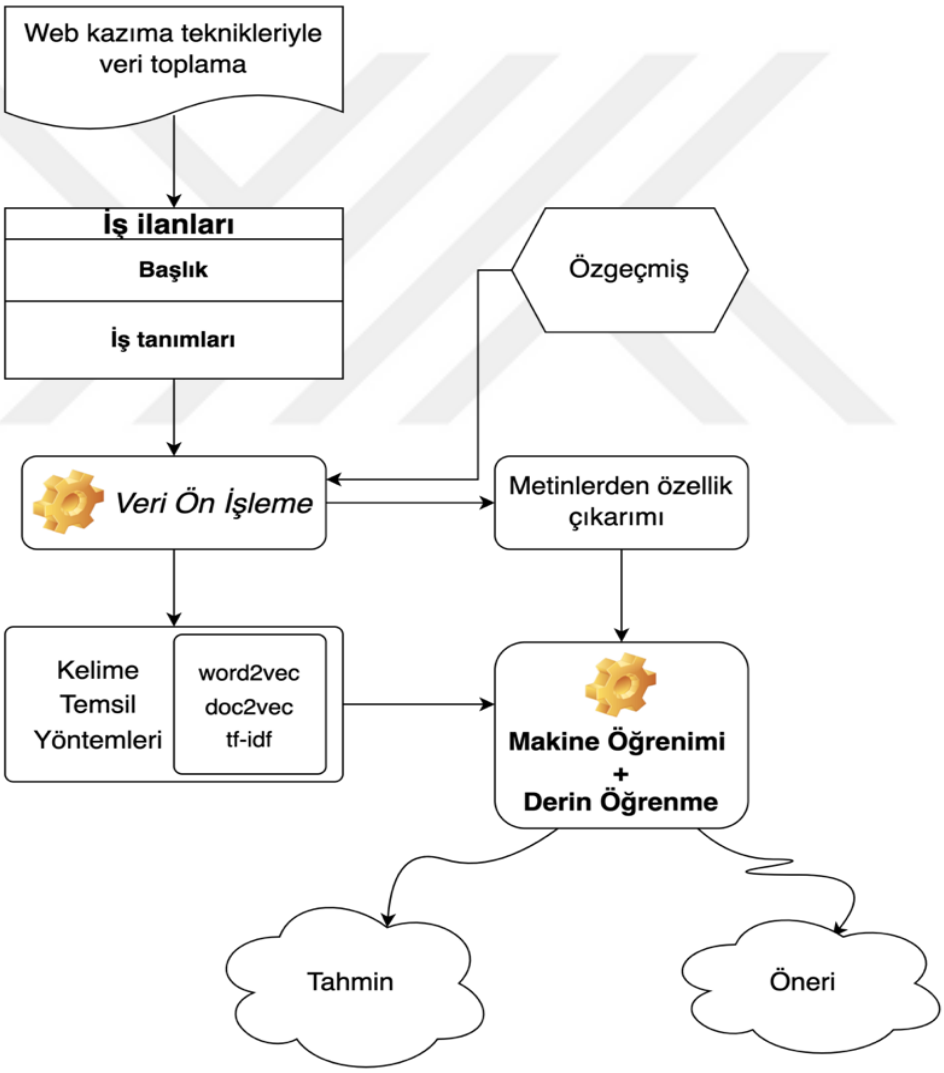
$$T_{Rel} = 1 + (DL + DR.\ldots.) * \frac{TF(t)}{MaxTF} \tag{2.12}$$

$$T_{Sentence} = \frac{SF(t)}{\#Sentences} \tag{2.13}$$

$$S(t) = \frac{T_{Rel} * T_{Position}}{T_{Case} + \frac{TF_{Norm}}{T_{Rel}} + \frac{T_{Sentence}}{T_{Rel}}} \tag{2.14}$$

# 3. YÖNTEM

Bu bölümde veri setinin toplanmasından, verilerin açıklanmasından, verilerin modelde eğitilebilmesi için geçirilmesi gereken ön işleme tekniklerinden bahsedilecek ve çalışmada kullanılan sınıflandırma yöntemleri anlatılacaktır. Özgeçmişlerde bulunması gereken yetkinliklerin önerilmesinde kullanılan öznitelik çıkarma tekniklerinden de bahsedilecektir. Sistem mimarisinin gösterildiği diyagram Şekil 3.1'de gösterilmiştir.




**Şekil 3.1:** Sistem mimarisi diyagramı.

## 3.1 VERİ SETİ

Çalışmada kullanılan veri seti, çevrimiçi bir platformdan toplanan iş ilanları ve açık kaynak olarak paylaşılan veri setlerinin bulunduğu Kaggle platformundan alınan iş ilanları verilerinden oluşmaktadır. Ham veri seti 3827 satır, 2 kolondan oluşmaktadır. Verilerin toplanması sırasında ilanların, linkleri, paylaşılma tarihleri ve şirket isimleri de alınmıştır ancak çalışmada sadece başlık ve iş tanımları kullanılmıştır. İlanların başlıkları aynı formatta yazılmamış şekilde geldiği için kategorilerine göre başlıkları düzenlenmiş ve toplamda 17 sınıftan oluşan veri seti hazırlanmıştır. Günümüzde bilişim alanında en çok iş ilanı yayınlanan ve aranan en popüler başlıklara göre oluşturulan 17 sınıf şu şekildedir; QA Engineer, Java Developer, .Net Developer, Web Developer, Network Engineer, Frontend Developer, Ios Developer, Game Developer, Backend Developer, Android Developer, Data Scientist, Project Manager, Devops Engineer, Devops Engineer, Database Administrator, Data Engineer, Full Stack Developer, Cyber Security. Şekil 3.2'de Dataframe formatına getirilen ham veri seti gösterilmiştir.

| | Title | RequiredQual |
|---|---|---|
| 0 | Database Administrator | - Degree in Computer Science, Information Tech... |
| 1 | Web Developer | - At least 2 years experience as a Web Develop... |
| 2 | Java Developer | Operational experience on a position of the\r\... |
| 3 | Network Engineer | - Good knowledge of Win 2000 Server, Win 2003 ... |
| 4 | Java Developer | - Minimum of 3-5 year Java Web Development exp... |
| ... | ... | ... |
| 3822 | Android Developer | job description job title android developer du... |
| 3823 | Data Scientist | leading financial services firm seeks principa... |
| 3824 | DevOps Engineer | evotek north america premier enabler digital b... |
| 3825 | Cyber Security | security analyst forrest jones company inc aff... |
| 3826 | Network Engineer | job description job title nginx plus software ... |

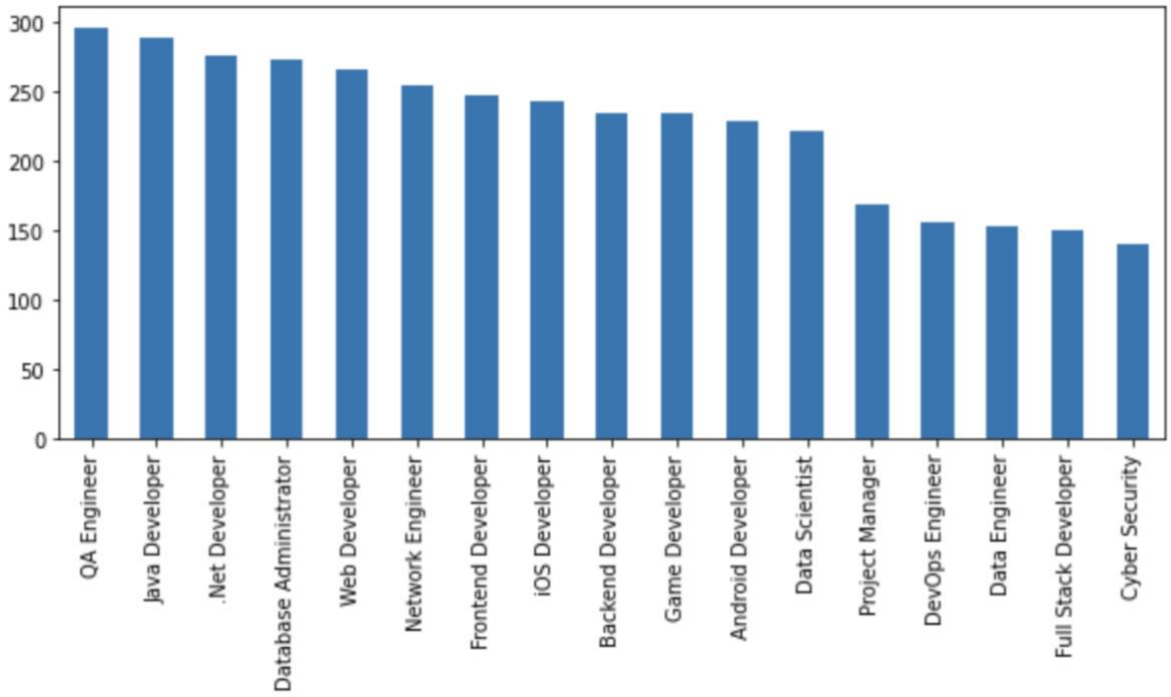
3827 rows × 2 columns

**Şekil 3.2:** Çalışmada kullanılan veri seti örneği.

### 3.1.1 Veri Ön İşleme

Verilerin sınıflandırma modelinde girdi olarak kullanılması için birtakım işlemlerden geçirilmesi gerekmektedir. Toplanan veriler istenmeyen kelimeler, noktalama işaretleri ve sembollerden oluştuğu için öncellikle bunların temizlemesi gerekmektedir. Doğal Dil İşleme

bölümünde bahsedilen NLTK kütüphane ile Python üzerinde veri ön hazırlık işlemleri yapılabilmektedir. Veri setinin iş ilanlarının tanımlarından oluşan RequiredQual kolonunda bulunan metinlerde İngilizce dışında başka dillerde yazılan ilanlarda gelmektedir. Bu çalışmada sadece İngilizce veriler ile çalışılacağından dolayı başka dilleri tespit edebilmek için Python'da bulunan "langdetect" kütüphanesi kullanmıştır. Bu kütüphane sayesinde metnin hangi dilde yazıldığı tespit edilebilmektedir. Çalışmada ihtiyaç duyulan İngilizce veriler olduğu için bunun dışında gelen ilanlar veri setinden çıkarılmıştır. Bu işlemler sonucunda veri sayısı 3824 olmuştur. Kategorilere göre veri dağılımları Şekil 3.3'te gösterilmiştir.




**Şekil 3.3:** Kategorilere göre verilerin dağılımı.

Veri ön işleme adımlarından bir diğeri ise metinlerden durdurma kelimelerinin kaldırılmasıdır. NLTK kütüphanesinde bulunan "stopwords" modülü ile İngilizce metinlerde bulunan "and, is, are, some" vb. kelimeler sınıflandırma aşamasında anlamlı değerler taşımadıkları için kaldırılmıştır. Şekil 3.4'te kütüphanede bulunan durdurma kelimelerinden bazıları gösterilmiştir.

```
i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he,
```

**Şekil 3.4:** Durdurma kelimeleri.

Durdurma kelimelerinin kaldırılmasından sonra metindeki tüm kelimelerin aynı şekilde temsil edilebilmesi için “lower()” fonksiyonu ile hepsi küçük harflere dönüştürülmüştür. Küçük harfe dönüştürülen metinlerde bulunan semboller, sayılar ve noktalama işaretleri kaldırılmıştır. Bu kısımda dikkat edilen c#, c++ gibi yetkinliklerde bulunan sembollerin kaldırılmaması için “#” ve “++” hariç tutulmuştur. Metinlerde belirli bir format olmadığı için ve algoritmalar “c#” gibi kelimeleri görmediği için bu ifadeler “csharp” ve “cplus” gibi sembollerden arındırılarak aynı şekilde temsil edilmesi sağlanmıştır. Ek alan ancak kökü aynı olan kelimeler de aynı şeyi ifade etmeleri sebebiyle eklerinden ve köklerinden arındırılması içe “Lemmatization” işlemi yapılmıştır. “lemmatize()” fonksiyonu ile kökleri aynı olan kelimeler, tek bir öğe olarak temsil edilebilmiştir. Hala istenilenden uzun metinlerin olması sebebiyle anlamsız kelimelerin kaldırılmasına devam edilmiştir. Metin parçası etiketleme (Part of sppech tagging) işlemi yapılarak metinlerdeki isim, sıfat, zarf, fiil gibi etiketlerle sözcük türleri belirlenmiştir. Veriler genel olarak analiz edildiğinde sınıflandırma modeli için ihtiyaç olan kelimelerin isim ya da sıfat olarak bulunduğu tespit edilmiştir. Bu kelimeler ilanların kategorilerini belirginleştirilen “java”, “android” gibi yetkinlik bildiren kelimelerdir. Tüm bu işlemler sonucunda veri seti temizlenmiş, istenmeyen kelimelerden arındırılarak hazır hale getirilmiştir.

## 3.2 İŞ İLANLARININ SINIFLANDIRILMASI

İlanlarının sınıflandırılması için iş tanımlarından yararlanılmıştır. Metin formatında olan iş tanımları veri ön işleme adımlarından geçirilerek hazır hale getirilmiştir. Bu verilerin bilgisayarların anlayacağı dile dönüştürülebilmesi için kelime temsil yöntemleri kullanılmıştır.

### 3.2.1 Metinlerin Sayısal Verilere Dönüştürülmesi

Metin formatında verilerle çalışıyorsak bu verileri modelin kullanabilmesi için sayısal veriler haline getirmemiz gerekmektedir. Bunun için birçok yol mevcuttur. Literatür araştırması sonucunda en çok kullanılan yöntemlerin TF-IDF, Word2Vec ve Doc2Vec olduğu görülmüştür. Bu çalışmada da üç kelime temsil yöntemi kullanılmıştır. Sınıflandırma algoritmalarıyla birlikte kullanılan bu üç yöntemin sınıflandırma performansları karşılaştırılmıştır. Sonuçlar 4. Bulgular tablo halinde verilecektir. Çalışmanın akış şeması Şekil 3.4’te verilmiştir.

### 3.2.2 Sınıflandırma Problemi için Kullanılan Algoritmalar

Çalışmanın ilk bölümdeki amacı metni verilen bir ilanın hangi sınıfa ait olduğu tespit etmektir. Çeşitli makine öğrenimi ve derin öğrenme teknikleri kullanılarak modellerin başarı oranları

ölçülmüştür. SVM, XGBoost, Rastgele Orman ve Lojistik Regresyon çalışmada kullanılan makine öğrenimi algoritmalarıdır. CNN, LSTM ve GRU ise çalışmada kullanılan derin öğrenme algoritmalarıdır. Modeller sırasıyla eğitilip başarı oranları ölçülmüştür. Doğruluk, f1-skor, kesinlik ve geri çağırma metriklerine göre karşılaştırılan algoritmalar sonucunda en yüksek başarı oranı SVM algoritması ile sağlanmıştır. %90,45 başaranı oranı sağlayan SVM daha önce hiç görmediği çevrimiçi bir platformdan alınan iş ilanı metni üzerinde ve manuel olarak yazılan özgeçmiş üzerinde test edilmiş ve ilk öneride doğru kategorileri getirmiştir.

## 3.3 YETKİNLİKLERİN ÖNERİLMESİ

Veri seti temizlemişlerinden önceki bölümlerde bahsedilmişti. Metinlerdeki isim, sıfat, zarf, fiil gibi etiketlerle belirlenen sözcüklerinden sıfat ve isim olanlar ayıklandı ve yeni bir kolon oluşturuldu. Veri seti kategorilere göre gruplanmış ve her kategoriye ait ilan tanımları birleştirilmiştir. Yetkinlik öneri sisteminde bu yeni oluşturulan metinler kullanılmıştır.

### 3.3.1 Yake ile Özellik Çıkarımı

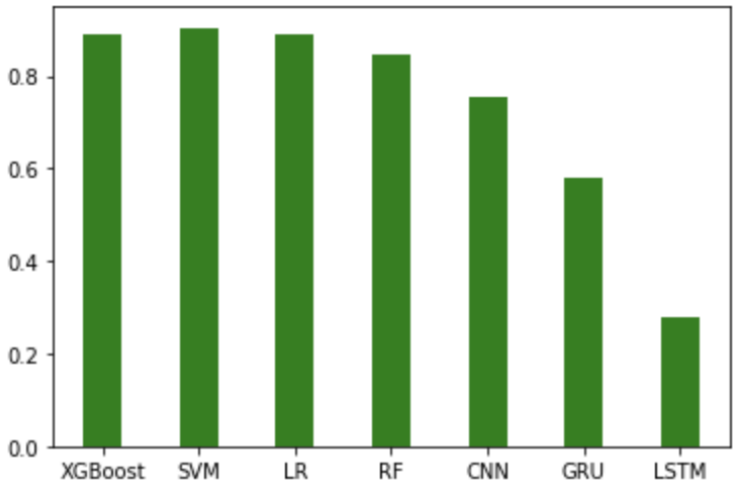
Python'a dahil edilerek kullanılan "yake" modülü ile metinlerden bilgi çıkarımı yapılabilmektedir. Birçok dilde kullanılabilen "yake", İngilizce metinlerde de başarılı sonuçlar vermektedir. Dil, maksimum sıralı kelime sayısı, seçilecek kelime sayısı vs. gibi parametreleri kullanarak ilgili kelimeleri çıkarmaktadır. İçerisine kategoriyi, kelime seçimi yapılacak metni ve liste şeklinde bir yetenek sözlüğünü parametre olarak alan fonksiyon yardımıyla yeni bir veri seti oluşturulmaktadır. Fonksiyonun içine aldığı sözlük manuel olarak oluşturulan bir yetenekler dizisinden oluşmaktadır. Kısa bir araştırma ile veri setindeki kategorilerle ilgili sahip olunması gereken programlama dilleri, araçlar gibi yetkinlikler bulunarak diziye eklenmiştir. Yake ile oluşturulan öznitelik çıkarımları bu sözlük kullanılarak yetenek ve yetenek değil şeklinde etiketlenmiştir. Yeni veri setinde özellik, yetenek/yetenek değil ve kategori kolonları bulunmaktadır. Bu veriler özgeçmişlerden veya veri setinde kullanılan iş ilanlarının tanım metinlerinden yetkinliklerin çıkarılabilmesi için eğitim verileri olarak kullanılmıştır. Veri seti LSTM ile eğitilmiştir. LSTM, yetkinlik önerilerinde %98 başarı oranı elde etmiştir. Yetkinlik önerileri 4. Bulgular bölümünde gösterilecektir.

# 4. BULGULAR

Bu bölümde sınıflandırma yöntemlerini ve kelime temsil yöntemlerini birlikte kullanarak performans ölçütleri ile başarı oranları karşılaştırıldı. En iyi sonucu veren yöntemler seçildi ve iş ilanlarının tanımları kullanılarak daha önce görmediği veriler üzerinde test edildi. Yetkinlik önerisi için gerekli işlemlerden geçirilen veriler üzerinde kategorilere ait yetkinlikler önerildi.

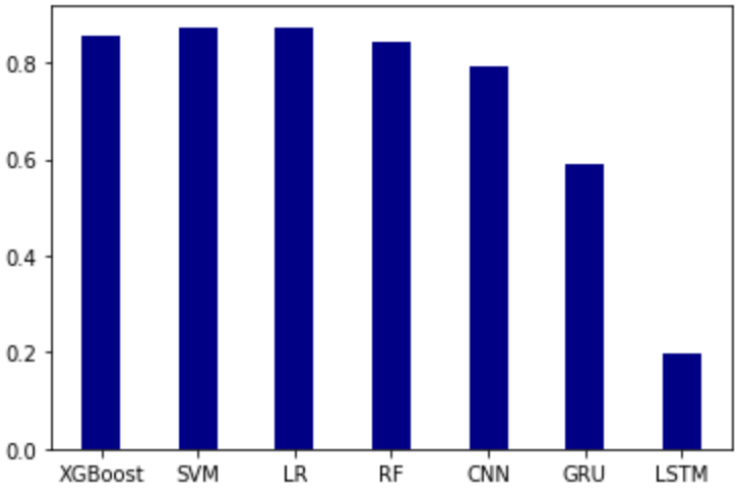
## 4.1 KELİME TEMSİL YÖNTEMLERİ VE SINIFLANDIRMA MODELLERİN BAŞARILARININ KIYASLANMASI

TF-IDF kelime vektörleştirme yöntemi kullanılarak makine öğrenimi ve derin öğrenimi modelleri aynı veri seti üzerinde eğitilmiş ve test verileri üzerindeki sonuçları gözlemlenmiştir. Çalışmada kullanılan algoritmaların TF-IDF ile kullanıldığında elde edilen başarı oranları Doğruluk (Accuracy) metriğine göre bar grafiği ile gösterilmiş, ilgili grafik Şekil 4.1'de verilmiştir.



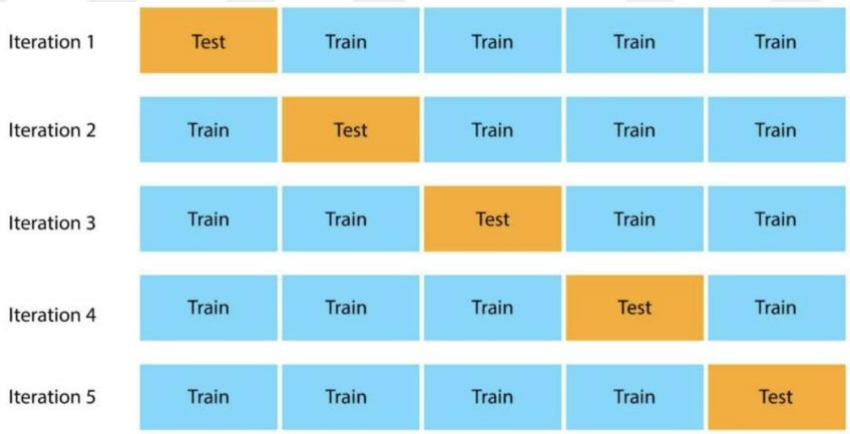

**Şekil 4.1:** Model performanslarının doğruluk değerleri.

Doğruluk performans ölçütüne göre en iyi sonuç TF-IDF ve SVM ile elde edilmiştir. Aynı yöntemlerin f1-skor performansları Şekil 4.2'de gösterilmiştir.



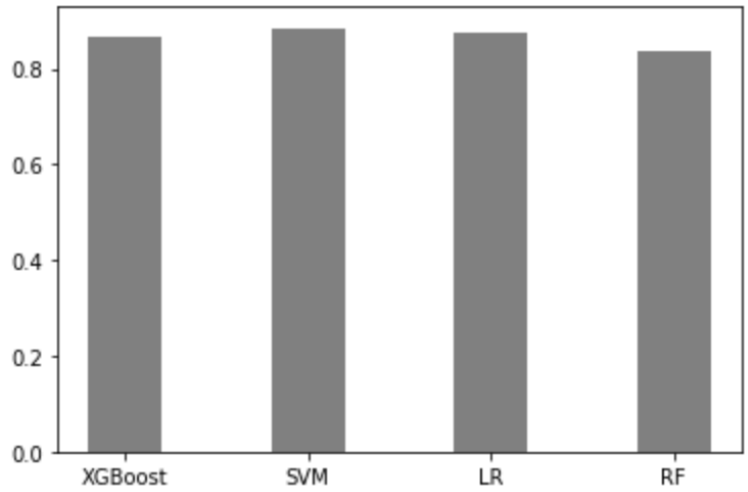

**Şekil 4.2:** Model performanslarının f1-skorları.

K-fold, çapraz doğrulama, modelin performansını doğrulamak için kullanılan bir tekniktir. Doğrulama seti olarak veri setinin farklı parçalarını kullanarak modeli değerlendirir. Veri setini K adet katlara ayırabiliriz. K, verilerinizi bölmek istediğiniz kat sayısını temsil eder. 5 kat olarak kullanırsak, veri seti beş bölüme ayrılır. Farklı yinelemelerde, bir parça doğrulama seti haline gelir (Affiah, 2022). Şekil 4.3'te 5 kat kullanarak bölünmüş veri seti örneği verilmiştir.




**Şekil 4.3:** K-fold.

Makine öğrenimi algoritmaları K-fold kullanarak test edilmiştir. Çalışmada kullanılan veri seti eşit dağılımda olmaması ve nispeten az veri bulunması sebebiyle sonuçlarda gözle görülür bir değişiklik sağlanmamıştır. Veri setine 10 kat çapraz doğrulama uygulanarak alınan sonuçlar Şekil 4.4'te verilmiştir.




**Şekil 4.4:** K-fold kullanarak eğitilen modellerin performans değerleri.

Word2vec, Doc2vec, TF-IDF yöntemlerinin sınıflandırılma modelleri ile kullanımı sonucunda elde edilen başarı oranları f1-skor, doğruluk, kesinlik ve geri çağırma metrikleri ile ölçülmüştür. TF-IDF ve modellerin birlikte kullanılması sonucunda ölçülen performans değerleri Tablo 4.1'de verilmiştir.

**Tablo 4.1:** TF-IDF ile elde edilen skorlar.

| | Accuracy | Precision | Recall | F1_score |
|---|---|---|---|---|
| SVM-TFIDF | 0.91 | 0.90 | 0.90 | 0.90 |
| RF-TFIDF | 0.85 | 0.85 | 0.85 | 0.84 |
| LR-TFIDF | 0.90 | 0.90 | 0.89 | 0.89 |
| XGBoost-TFIDF | 0.89 | 0.89 | 0.89 | 0.89 |
| CNN-TFIDF | 0.73 | 0.72 | 0.74 | 0.72 |
| LSTM-TFIDF | 0.22 | 0.23 | 0.23 | 0.19 |
| GRU-TFIDF | 0.57 | 0.54 | 0.56 | 0.53 |

Word2Vec ve Doc2Vec olmak üzere 2 farklı kelime temsil yöntemi ile beraber kullanılan modellerin performans değerleri sırasıyla Tablo 4.2 ve Tablo 4.3'te gösterilmiştir.

**Tablo 4.2:** Word2Vec ile elde edilen skorlar.

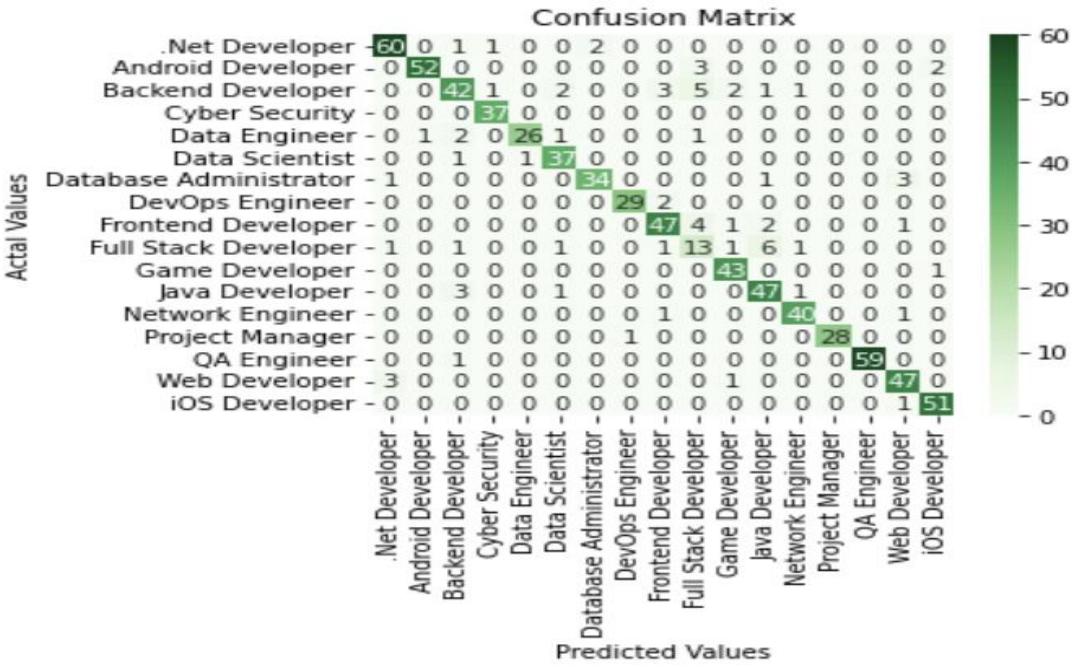
| | Accuracy | Precision | Recall | F1_score |
|---|---|---|---|---|
| SVM-Word2Vec | 0.84 | 0.83 | 0.83 | 0.83 |
| RF-Word2Vec | 0.76 | 0.78 | 0.76 | 0.75 |
| LR-Word2Vec | 0.84 | 0.84 | 0.83 | 0.83 |
| XGBoost-Word2Vec | 0.78 | 0.77 | 0.77 | 0.77 |
| CNN-Word2Vec | 0.71 | 0.71 | 0.70 | 0.68 |
| LSTM-Word2Vec | 0.71 | 0.70 | 0.71 | 0.68 |
| GRU-Word2Vec | 0.62 | 0.60 | 0.62 | 0.61 |

**Tablo 4.3.** Doc2Vec ile elde edilen skorlar.

| | Accuracy | Precision | Recall | F1_score |
|---|---|---|---|---|
| SVM-Doc2Vec | 0.53 | 0.51 | 0.52 | 0.51 |
| RF-Doc2Vec | 0.52 | 0.63 | 0.44 | 0.48 |
| LR-Doc2Vec | 0.58 | 0.57 | 0.57 | 0.57 |
| XGBoost-Doc2Vec | 0.48 | 0.52 | 0.48 | 0.49 |
| CNN-Doc2Vec | 0.51 | 0.50 | 0.48 | 0.47 |
| LSTM-Doc2Vec | 0.56 | 0.55 | 0.55 | 0.55 |
| GRU-Doc2Vec | 0.46 | 0.45 | 0.45 | 0.45 |

Makine Öğrenimi algoritmalarında tahmin performansını arttırmak için modellerin parametrelerinin en optimum değere sahip olması gerekmektedir. Optimum parametreler manuel olarak deneme yanılma yöntemiyle de bulunabileceği gibi bu yöntem kullanışlı değildir. Otomatik olarak bu işlemi yapan teknikler mevcuttur. Bu çalışmada kullanılan

GridSearchCV bu yöntemlerden bir tanesidir. Modelde denenmesi istenen hiperparametreler ve değerleri için bütün kombinasyonlar ile ayrı ayrı model kurulur ve belirtilen metriğe göre en başarılı hiperparametre seti belirlenir. Kelime temsil yöntemleriyle birlikte kullanılmış modellerin performansları karşılaştırılmış ve en yüksek başarı oranı SVM algoritmasının TF-IDF kelime temsil yöntemiyle birlikte kullanılması sonucunda elde edilmiştir. En yüksek performansa sahip SVM seçildikten sonra performansının arttırılabilmesi için GridSearchCV yöntemi kullanılmıştır. SVM modelinin "kernel" parametresi için "linear", "poly", "rbf" ve "C" parametresi her kernel için [0.0001, 0.001, 0.1, 1, 10, 100, 1000] değerlerinin denenmesi için GiridSearchCV algoritmasına verilmiştir. Algoritmanın her kombinasyonu denemesi sonucunda en iyi performansı kernel: linear, C:1 olması durumlarında gösterdiğini bulmuştur. Bu değerlere göre yeniden eğitilen SVM modelinin hata matrisi Şekil 4.5'te verilmiştir.

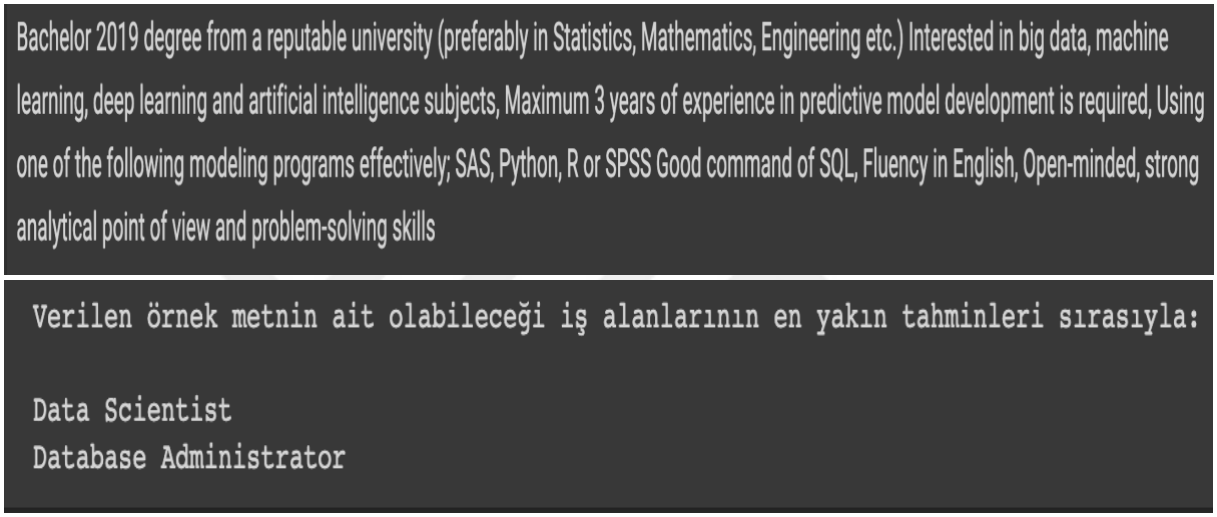



**Şekil 4.5:** SVM modelinin hata matrisi.

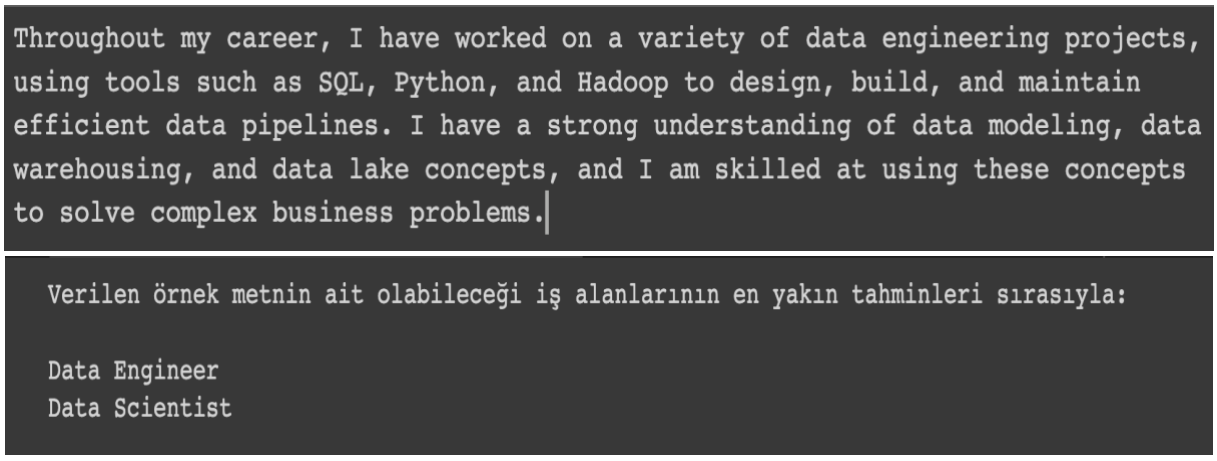
## 4.2 İŞ İLANLARININ SINIFLANDIRMASI

Sınıflandırma modellerinin farklı kelime vektörleştirme yöntemleriyle birlikte kullanılması sonucunda en yüksek başarı oranını sağlayan TF-IDF ve SVM modelinin birlikte kullanımıyla elde edilmiştir. Veri seti TF-IDF kullanılarak kelime vektörleri hazırlanmış ve vektörleştirilen veri seti Python'da kullanılan Scikit-Learn (Bressert, 2012) kütüphanesinin "train_test_split"

modülü ile %80 eğitim, %20 test verisi olmak üzere ayrılmıştır. Test verileri üzerinde %91 oranında başarı sağlayan SVM, hiç görmediği, çevrimiçi iş ilanı platformunda rastgele seçilen iş tanımı metni üzerinden doğru tahmini yapmıştır. Şekil 4.6'da verilen metinden, kategori tahmin örneği gösterilmiştir. Eğitim verilerinde iş ilanlarının tanımlarının kullanılması ilanların sınıflandırılması dışında test metni olarak özgeçmiş verildiğinde de doğru sınıflandırılması sağlanmıştır. İş tanımlarında geçen teknik beceriler, özgeçmişlerde de bulunması sebebiyle başarılı sınıflandırmalar yapılmaktadır. Şekil 4.7'de ChatGPT (OpenAI, 2022) Yapay Zeka sohbet robotu tarafından üretilen özgeçmiş üzerinden kategori tahmin örneği verilmiştir.



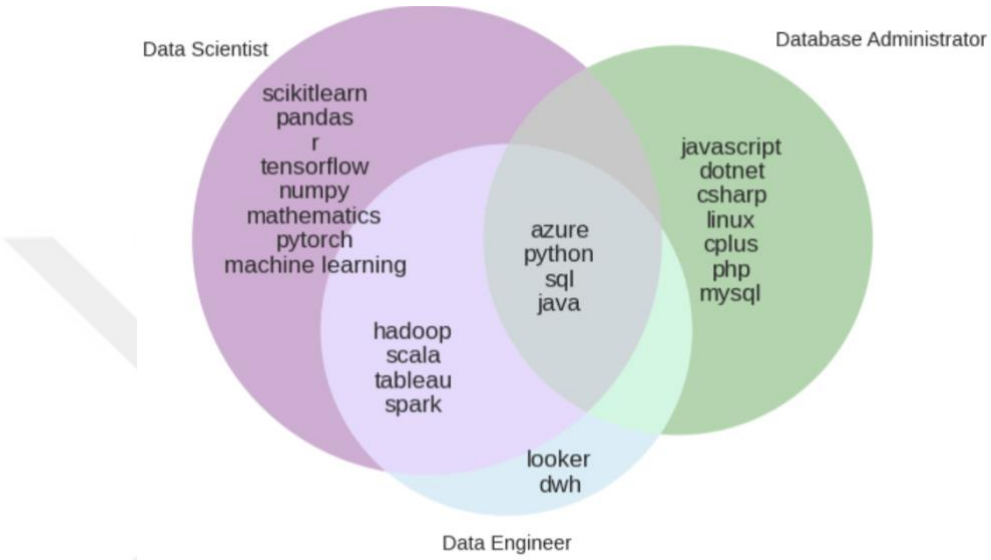

**Şekil 4.6:** Çevrimiçi platformdan alınan iş ilanı metninden kategori tahmin örneği.




**Şekil 4.7:** ChatGPT tarafından üretilen Data Engineer özgeçmiş örneğinden kategori tahmini.

Şekil 4.8'de "Data Scientist", "Data Administrator" ve "Data Engineer" kategorilerine ait belirgin özellikler ven şeması ile gösterilmiştir. Şekil 4.6 ve Şekil 4.7'de görüldüğü gibi model

bu kategorilerden birine ait olan metinleri tahmin etmeye çalıştığında birbirlerinin yerine ya da en yakın 2 tahmin olacak şekilde sonuçlar üretebiliyor. Ven şemasında da görüldüğü gibi metinlerde ortak yetkinlikleri barındırmaları sebebiyle bu şekilde sonuçlar ürettiğini söyleyebiliriz. Şekil 4.8 ve Şekil 4.11'de gösterilen ven şemaları, Python'da kullanılabilen "matplotlib_venn" kütüphanesi ile oluşturulmuştur.




**Şekil 4.8:** Benzer kategorilere ait özelliklerin venn şeması ile gösterimi.

```
I am using languages such as php, java and c#. Also using frameworks such as
node.js I have a strong understanding of object-oriented programming principles
and experience with REST API development. In addition to my technical skills, I
am also a team player and have excellent communication and problem-solving
abilities. I am able to work independently as well as collaborate with other
team members to find solutions to complex challenges.
```

```
Verilen örnek metnin ait olabileceği iş alanlarının en yakın tahminleri sırasıyla:

Backend Developer
Java Developer
```

**Şekil 4.9:** Backend Developer alanında çalışan bir kişinin özgeçmiş örneği ve kategori tahmini.

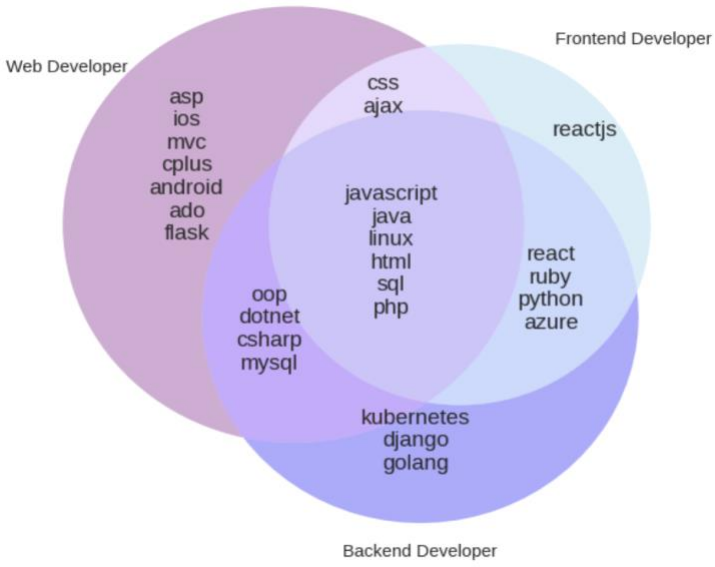
```
I have been responsible for designing, developing, and maintaining a variety of
web-based applications and systems. I have experience with technologies such as
CSS, HTML, javascript and RESTful APIs, and I am skilled at using these tools
to create user-friendly, visually appealing websites that meet the needs of the
business and its users.
```

```
Verilen örnek metnin ait olabileceği iş alanlarının en yakın tahminleri sırasıyla:

Web Developer
Frontend Developer
```

**Şekil 4.10:** Frontend Developer alanında çalışan bir kişinin özgeçmiş örneği ve kategori tahmini.

Şekil 4.9'da özgeçmiş örneğinden kategori tahmini yapılmıştır. Şekildeki özgeçmiş "Frontend Developer" alanında çalışan bir kişiye aittir ancak modelin ilk tahmini "Web Developer" olarak görülmektedir. Burada da başlıca sebebin "Web Developer", "Frontend Developer" ve "Backend Developer" gibi kategorilerdeki iş ilanlarında benzer yetkinliklerin aranmasından dolayı cümlelerde geçen kelimelerin de genellikle ortak olmasından kaynaklandığı düşünülmektedir. Örnek olarak bu 3 kategoriye ait yetkinliklerin ven şeması ile gösterimi Şekil 4.10'da gösterilmiştir. Ven şemasındaki kesişimler kategorilerde bulunan örnek yetkinlikleri belirtmektedir. Özellikle 3 kümenin kesişim noktasında bulunan yetkinlikleri temsil eden kelimelerin çok olmasından, bu kategorilere ait ilanlarda benzer özelliklerin arandığını söyleyebiliriz.




**Şekil 4.11:** 3 farklı kategoriye ait belirgin yetkinliklerin venn şeması gösterimi.

## 4.3 ÖNERİ SİSTEMİNİN GELİŞTİRİLMESİ

Metinlerden kategori tahmini yapılması öneri sisteminin önemli bir adımıdır. Kategorisi belirlenen metinden "Yake" ile özellik çıkarımı yapılmıştır. Yetenek sözlüğü kullanılarak dataframedeki kelimeler "0" yetenek değil, "1" yetenek anlamına gelecek şekilde etiketlenmiştir. Aslında buraya kadar yapılan işlemlerle sadece bu çalışma doğrultusunda doğru yetkinlik önerileri yapılabilmektedir. Fakat manuel olarak oluşturulan sözlük genel geçer yetkinlikleri barındırdığı için gelişen teknoloji ile yeni çıkan programlama dilleri, araçlar vb. yetkinliklerin tespit edilememe ihtimali vardır. Bu yüzden yetkinlikleri öğrenebilen bir model yaratılması amaçlanmıştır. Aynı yöntemler ile iş ilanlarını tanım metinleri kullanılarak bir eğitim veri seti oluşturulmuştur. Bu veri seti LSTM modelinde eğitim veri seti olarak kullanılmıştır. Eğitilen LSTM modeli yeni metinler üzerinde yetenek/yetenek değil şeklinde kelimeleri etiketleyebilmektedir. Modelin verilen metinden yetenek olarak çıkardığı kelimeler ile önceden hazırlanan özellik, yetenek ve kategori kolonlarından oluşan veri seti ile karşılaştırılmaktadır. Önce verilen metnin kategorisi belirlendiği için bu kategoriye ait yetenek olarak etiketlenmiş olan kelimeler veri setinde tespit edilir ve eğitilen modelin yetenek olarak çıkardığı kelime ve kelime gruplarında olmayan ama metnin ait olduğu kategoride yetenek olarak bulunan özelliklerin kişiye önerildiği bir sistem geliştirilmiştir. Şekil 4.10'da gösterilen örnek özgeçmiş SVM modeli ile sınıflandırılmış ve kategorisi model tarafından tahmin edilmiştir.

```
Throughout my career, I have worked on a variety of data engineering projects,
using tools such as SQL, Python, and Hadoop to design, build, and maintain
efficient data pipelines. I have a strong understanding of data modeling, data
warehousing, and data lake concepts, and I am skilled at using these concepts
to solve complex business problems.
```

```
Verilen örnek metnin ait olabileceği iş alanlarının en yakın tahminleri sırasıyla:

Data Engineer
Data Scientist
```

**Şekil 4.12:** Özgeçmiş örneği ve tahmin edilen kategorileri.

Kategorisi bulunan özgeçmişten çıkarılan öznitelikler Şekil 4.11'de verilmiştir. Yetenek olarak tahmin edilenler "1" diğerleri ise "0" olarak etiketlenmiştir. Verilen özgeçmişin tahmin edilen en yakın kategorisi "Data Engineer" olarak görülmüştür. "Data Engineer" alanında sahip olunması gereken nitelikler, iş ilanlarından çıkarılan kelime ve yetenek sözlüğü yardımıyla oluşturulan veri setinden kategori bilgisi parametre olarak kullanılarak kategoriye ait ilgili yetkinlikler bulunmuş ve Şekil 4.12'de gösterilmiştir. Modelin tespit ettiği özgeçmişte bulunan yetkinlikler dışında kalan, yani özgeçmiş sahibinin ilgili alanda iş bulabilmesi için sahip olması gereken yetkinliklerden eksik olanlar öneri olarak sunulmuştur. Öneri olarak sunulan yetkinlikler Şekil 4.13'te gösterilmiştir.

| | Feature | is_skill |
|---|---|---|
| 0 | data | 0 |
| 1 | concepts | 0 |
| 2 | career | 0 |
| 3 | problems | 0 |
| 4 | variety | 0 |
| 5 | engineering | 0 |
| 6 | projects | 0 |
| 7 | tools | 0 |
| 8 | sql | 1 |
| 9 | python | 1 |
| 10 | hadoop | 1 |
| 11 | build | 0 |
| 12 | efficient | 0 |
| 13 | pipelines | 0 |
| 14 | strong | 0 |
| 15 | understanding | 0 |
| 16 | warehousing | 0 |
| 17 | lake | 0 |
| 18 | complex | 0 |
| 19 | business | 0 |

**Şekil 4.13:** Özgeçmişten çıkarılan öznitelikler.

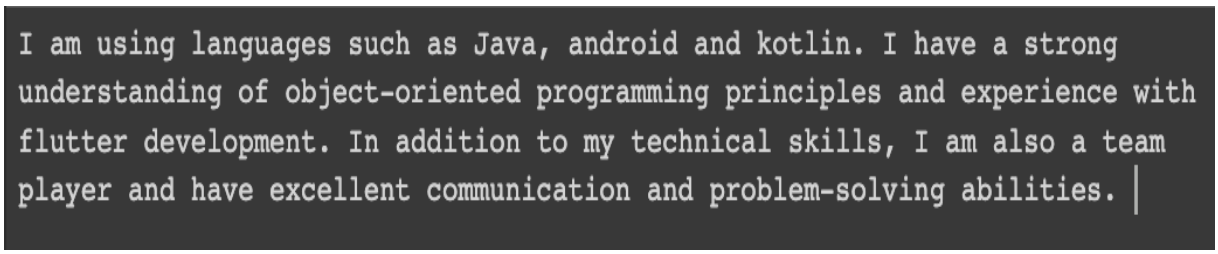
| | Feature | is_skill |
|---|---|---|
| 195 | sql | 1 |
| 318 | looker | 1 |
| 1584 | dwh | 1 |
| 2637 | numpy | 1 |
| 2729 | python | 1 |
| 3491 | airflow | 1 |
| 3654 | mssqlserver | 1 |
| 3822 | java | 1 |
| 4185 | hadoop | 1 |
| 4251 | mysql | 1 |
| 4732 | hadoop | 1 |
| 5480 | scala | 1 |
| 5702 | spark | 1 |
| 5765 | azure | 1 |
| 5857 | tableau | 1 |
| 6364 | spark | 1 |
| 7723 | python | 1 |
| 7779 | deep learning | 1 |
| 8133 | sql | 1 |
| 8336 | scala | 1 |

**Şekil 4.14:** Data Engineer ile ilgili yetkinlikler.




**Şekil 4.15:** Önerilen yetkinlikler.

Önerilen yetkinlikler arasında daha çok Data Scientist alanında görülen "deep learning", "numpy" gibi yetkinlikler de bulunmaktadır. Bunun sebebi eğitim verisi olarak kullanılan veri seti oluşturulurken iş ilanlarından yararlanıldığı için bazı "Data Engineer" ilanlarında "şu özellikler de olsa iyi olur", "şu özelliklerin olması bir artıdır" gibi cümlelerle ekstra yetkinlikler talep edilmesinden kaynaklandığı tahmin edilmektedir. Farklı bir özgeçmiş örneğinden çıkarılan öznitelikler ve tahmin edilen kategoriye ait, özgeçmişte eksik olan yetkinliklerin önerilmesi sırasıyla Şekil 4.14, Şekil 4.15, Şekil 4.16 ve Şekil 4.17'de gösterilmiştir. Bu örnek özgeçmişe göre önerilen kelimelerde de "Android Developer" kategorisinde beklemeyeceğimiz birkaç özellik gelmiş gözüküyor yukarıdaki örnekte olduğu gibi aynı sebepten dolayı bu yetkinliklerin geldiğini söyleyebiliriz. Özgeçmişin dahil olabileceği 2 en yakın kategori önerisinde 2. Öneri olarak "Game Developer" geldiği görülmektedir. Model performansının %91 olması nedeniyle elbette kesin tahmin üretememektedir. Bu sapmaların genel olarak 2. tahminlerde olması sebebiyle önemli derece bir sorun oluşturmamaktadır.




**Şekil 4.16:** Özgeçmiş ve tahmin edilen kategorisi.

```
Verilen örnek metnin ait olabileceği iş alanlarının en yakın tahminleri sırasıyla:

Android Developer
Game Developer
```

**Şekil 4.16 (devam)**

| | Feature | is_skill |
|---|---|---|
| 320 | swift | 1 |
| 624 | cplus | 1 |
| 998 | ui | 1 |
| 1641 | ios | 1 |
| 2229 | sql | 1 |
| 2446 | kotlin | 1 |
| 2664 | java | 1 |
| 3617 | html | 1 |
| 4271 | csharp | 1 |
| 5257 | javascript | 1 |
| 5914 | android | 1 |
| 5962 | flutter | 1 |
| 7158 | android | 1 |
| 7383 | oop | 1 |
| 7410 | kotlin | 1 |
| 8514 | linux | 1 |

**Şekil 4.17:** Android kategorisine ait yetkinlikler.

| | Feature | is_skill |
|---|---|---|
| 0 | challenges | 0 |
| 1 | team | 0 |
| 2 | languages | 0 |
| 3 | java | 1 |
| 4 | android | 1 |
| 5 | kotlin | 1 |
| 6 | strong | 0 |
| 7 | understanding | 0 |
| 8 | object | 0 |
| 9 | programming | 0 |
| 10 | principles | 0 |
| 11 | experience | 0 |
| 12 | flutter | 1 |
| 13 | development | 0 |
| 14 | addition | 0 |
| 15 | technical | 0 |
| 16 | skills | 0 |
| 17 | player | 0 |
| 18 | communication | 0 |
| 19 | problem | 0 |

**Şekil 4.18:** Özgeçmişten çıkarılan öznitelikler.

```
['linux',
 'ios',
 'cplus',
 'html',
 'csharp',
 'oop',
 'sql',
 'ui',
 'swift',
 'javascript']
```

**Şekil 4.19:** Özgeçmişte eksik olan yetkinliklerin önerilmesi.

# 5. TARTIŞMA

Bu çalışmada, iş tanımlarının kullanıldığı bir metin sınıflandırma çalışması yapılmıştır. Literatür incelediğinde metin sınıflandırma alanında genel olarak haber metinlerinin sınıflandırılması, belgelerin sınıflandırılması gibi uygulamalar yapılmıştır. Metin madenciliğinin uygulandığı diğer bir çalışma türü ise duygu analizidir. Farklı internet sitelerinden genellikle yorumlardan oluşan verilerde kullanıcıların hangi duyguyu ifade ettikleri tespit edilmektedir. Bu çalışmalarda negatif ve pozitif olmak üzere 2 sınıf bulunmaktadır. Bazı çalışmalarda yansız (nötr) sınıfı da bulunabilmektedir. Bahsedilen çalışmaların dışında metin madenciliği veya metin sınıflandırma yöntemlerinin kullanılabileceği birçok alan mevcuttur. Gelişen teknolojiyle birlikte internet kullanımı günümüzde bu kadar yaygınken neredeyse her yerde bir şeyler ifade eden metinler görebiliriz. Bu çalışmada da literatür incelemesi sonucunda diğerlerine göre daha az işlenmiş bir konu hakkında bir yöntem önerilmiştir. Çevrimiçi platformlardan alınan iş ilanları metin sınıflandırma yöntemleri ile kategorileri tespit edilmiş ve Bilişim alanında çalışmak isteyen kişilere bilmesi gereken yetkinliklerin önerildiği karar destek sistemi sunulmuştur. Bu sistem sayesinde iş tanımlarının bulunduğu metinler analiz edilerek önemli yetkinlikler de öne çıkarılabilmektedir. İş ilanlarının sınıflandırılması ya da iş ilanlarındaki yetkinliklerin önerilmesi literatürde diğer konulara göre daha az olsa da işlenen bir konudur. Giriş bölümünde de belirtilmiş olan benzer çalışmalarda genel olarak iş ilanları ile kişilerin özgeçmişlerinin eşleştirilmesi hedeflenmiştir. İçerik tabanlı filtreleme, işbirlikçi filtreme ve öğe tabanlı filtreme yöntemleri ile iş ilanları önerilmektedir. Bu çalışmada iş ilanları direkt bir öneri olarak sunulmamaktadır. Buradaki amaç daha çok yol gösterici olması açısından çalışılmak istenilen alana göre eksik yeteneklerin önerilmesidir. Veri setinde Bilişim alanı altında listelenen 17 kategori bulunmaktadır. İş ilanlarının sınıflandırılması ile ilgili yapılan çalışmalarda Bilişim alanında bu sayıda kategoriye sahip verilerle çalışılmamıştır. Roy ve diğ. (2020) yaptığı bir çalışmada 24 alana ait iş ilanlarını sınıflandırmışlardır. Fakat yapılan bu çalışmada Tarım, Mimarlık, Mühendislik gibi çok geniş kapsamlı alanlar kategori olarak belirlenmiştir. Makine öğrenimi algoritmalarından Rastgele Orman, Naif Bayes (Navie Bayes), Lojistik Regresyon ve Lineer SVM kullanmışlar ve en başarılı algoritma olarak SVM ile %78 başarı oranı elde etmişlerdir. Bu çalışmada da Lineer SVM kullanılmış ve Bilişim alanında bulunan 17 kategoride %91 başarı oranı ile sınıflandırma gerçekleştirilmiştir. Diğer bir

çalışmada ise Huynh ve diğ. (2020) farklı derin öğrenme yöntemlerini birleştirerek 25 kategoriden oluşan iş tanımlarından başlıklarını tahmin etmeye çalışmışlardır. Bilişim sektöründe bulunan kategorilerden oluşan veri seti çalışmanın benzer yönüdür ancak sınıflandırma için kullanılan modellerin hepsi derin öğrenme yöntemlerinden oluşmasıyla ve elde edilen %72,71 F1 skoru ile bu tez çalışmasından ayrılmaktadır. Daha az kategori bulunmasının ve veri setini farklı olmasının başarı oranının yükselmesinde etkili olduğu da söylenebilir. Öneri sisteminde ise kategorisi belirlenen ilanın yetkinlikleri çıkarılmaktadır. Alsaif ve diğ. (2022) Word2Vec yöntemi ile iş ilanları metinlerini vektörleştirmişler, Jaccar Katsayısı yöntemi ve Cosinüs Benzerliği yöntemini kullanarak 5 kategoriye ait iş ilanlarından öznitelikleri çıkarmışlardır. Çıkarılan özniteliklerle özgeçmişlerdeki yetkinliklerin benzerlikleri ölçülmüş ve en benzer yetkinliklerin bulunduğu iş ilanlarını önermişlerdir. Jaccar Katsayısı ile %72, Cosinüs Benzerliği ile de %85 başarı oranı elde etmişlerdir. Yapılan çalışmaya bakıldığında özniteliklerin çıkarılmasında farklı yöntemler kullanılmıştır. Bu çalışmada kelimeleri vektör uzayında temsil etmek için Word2Vec yöntemi kullanılmış ancak performans değerlerine bakıldığında TF-IDF yönteminin daha iyi sonuç verdiği görülmüştür. Ayrıca sınıflandırma modelleri için de derin öğrenme ve makine öğrenimi gibi yöntemler kullanılmıştır. En yüksek başarı oranı TF-IDF ile vektörleştirilen metinlerin SVM ile sınıflandırılmasından elde edilmiştir. Aynı yöntem özgeçmişlerin sınıflandırılması için de kullanılmış ve modelin tahmin sonuçları gerçeğe yakın olmuştur. Hangi sınıfa ait olan bir özgeçmiş analiz edilerek eksik yetkinlikler belirlenmiş ve ilgili iş alanına ait yetkinlik önerileri sunulmuştur. Çalışmada önerilen yöntemin elde edilen performans değerlerine bakıldığında benzer çalışmalara göre daha başarılı olduğu görülmektedir.

**Tablo 5.1** İlgili çalışmalar.

| Yazar | Makale | Yöntem | Sonuç |
|---|---|---|---|
| Roy, P. K., Chowdhary, S. ve Bhatia, R., (2020) | A Machine Learning approach for automation of Resume Recommendation system | Lineer SVM TF-IDF | %78 |
| Huynh, T. V., Nguyen, K. V., Nguyen, N. L. ve Nguyen, A. G., (2020) | Job Prediction: From Deep Neural Network Models to Applications | Bi-GRU-LSTM-CNN FastText | %71,20 |
| Alsaif, S.A., Hidri, M. S., Eleraky, H.A., Ferjani, I. ve Amami, R., (2022) | Learning-Based Matched Representation System for Job Recommendation | Kosinüs Benzerliği Word2Vec | %85 |
| Lin, Y., Lei, H., Addo, P. C., ve Li, X., (2016) | Machine learned resume-job matching solution | IBagging | %47 |
| Tran, H. T., Vo, H. H. P. ve Luu, S. T., (2021) | Predicting Job Titles from Job Descriptions with Multi label Text Classification | Bi-GRULSTM-CNN Bert | %47,44 |

# 6. SONUÇ VE ÖNERİLER

Farklı modellerin denenmesi sonucunda önerilen yöntem TF-IDF ve SVM yöntemlerinin birlikte kullanıldığı sınıflandırma modelidir. Makine öğrenimi algoritmalarının yerini almaya başlayan ve günden güne daha popüler hale gelen derin öğrenme yöntemlerindense yine bir makine öğrenimi algoritmasının hala bazı veriler üzerinde daha iyi sonuçlar ürettiği gösterilmiştir. Harcanan zaman bakımından da aynı veri seti üzerinde eğitim sürelerine bakılmış ve eğitim süresi derin öğrenme yöntemlerine göre daha hızlı sürede sonuçlanmıştır. Model kısa sürede iş ilanları verileri üzerinde %91 başarı oranı elde etmiştir. Veri setindeki 17 kategoriye ait ilan sayısı homojen dağılıma sahip değildir. Sonraki aşamalarda daha çok veri toplanarak veriler homojen hale getirilebilir. Bu durumda modelin performansı daha da artacaktır. İş ilanlarından Yake modülü ile öznitelikler çıkarılmış ve öneri sisteminin zemini hazırlanmıştır. Kural tabanlı sözlük yöntemiyle öznitelikler etiketlenmiş ve yeni bir veri seti oluşturulmuştur. Bu veriler LSTM modelin eğitim verileri olarak kullanılmıştır. Sınıfı tahmin edilen özgeçmiş veya bir iş ilanı metninden bu model ile yetkinlikler tespit edilmiştir. Eksik olan yetkinlikler önceden etiketlenen kategorilere ait yetkinlerden bulunarak öneri olarak sunulmuştur. Bu önerilerin doğruluğu verilen özgeçmiş metni ya da iş ilanı metnine göre değişebilmektedir. Bunun sebebi ise LSTM modeline verilen yetkinlik veri setinin nispeten az veriden oluşmasıdır. Model yeni test verileri geldikçe öğrenmeyi arttırarak performans değerini arttırması beklenmektedir. Sonraki çalışmalarda daha büyük yetkinlik veri setleri ile eğitilerek de yüksek doğruluk oranları elde edilebilir. Öneri sistemi için özgeçmiş girilmemesi durumunda istenilen bir kategorinin belirtilmesi halinde tüm yetkinlikler sözlük tabanlı yetkinlik listesinden önerilebilmektedir. Çalışmanın, yetkinlik öneri sistemi ve iş ilanlarının sınıflandırılması üzerine çalışmak isteyenler için ortam hazırlayabileceği ve daha iyi sonuçlar alabilmek için geliştirilmeye açık olduğu düşünülmektedir.

# KAYNAKLAR


Affiah, I., *How to Implement K fold Cross-Validation in Scikit-Learn*, https://www.section.io/engineering-education/how-to-implement-k-fold-cross-validation/, [Ziyaret Tarihi: 3 Aralık 2022]

Alsaif, S.A., Hidri, M. S., Eleraky, H.A., Ferjani, I. ve Amami, R., 2022, Learning-Based Matched Representation System for Job Recommendation, *Computers*, 11. (11), 161, doi.org/10.3390/computers11110161

Akça, M. F., *LSTM Nedir? Nasıl Çalışır?,* https://mfakca.medium.com/lstm-nedir-nasıl-çalışır-326866fd8869, [Ziyaret tarihi: 3 Aralık 2022]

Bilgin, M., ve Şentürk, I. F., 2017, Sentiment analysis on Twitter data with semi-supervised Doc2Vec, International Conference on Computer Science and Engineering (UBMK), doi.org/10.1109/UBMK.2017.8093492

Breiman, L., 2001, Random Forest, *Machine Learning* 45, 5–32, doi.org/10.1023/A:1010933404324

Bressert, E., 2012, *SciPy and NumPy: an overview for developers*. O'Reilly, 43, ISBN: 9781449361624

Cai, J., Li, J., Li, W. ve Wang, J., 2018, Deeplearning Model Used in Text Classification, *15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2018, 123-126, doi:10.1109/ICCWAMTIP.2018.8632592.

Chang, W., Xu, Z., Zhou, S. ve Cao, W., 2018, Research On Detection Methods Based On Doc2vec Abnormal Comments, *Future Generation Computer Systems*, 86, 2018, 656-662, ISSN 0167-739X,

Chen, T., ve Guestrin, C., 2016, XGBoost: A Scalable Tree Boosting System, *In Proceedings* of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), Association for Computing Machinery, New York, NY, USA, 785–794, doi.org/10.1145/2939672.2939785

Compos, R., Mangaavite, V., Pasqualli, A., Jorge, A., Nunes, C. ve Jatowt, A., 2020, YAKE! Keyword extraction from single documents using multiple local features, *Information Sciences*, 509, 257- 289, doi.org/10.1016/j.ins.2019.09.013

Çelik, Ö. ve Koç, B. C., 2021, TF-IDF, TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen Bilimleri Dergisi*, 23(67), 121-127, doi:10.21205/deufmd.2021236710

Çelikten, A., 2020, *RNN (Recurrent Neural Network) Nedir ve Nasıl Çalışır?*, https://medium.com/@azercelikten/rnn-recurrent-neural-network-nedir-ve-nasıl-çalışır-d1246b1a61fb, [Ziyaret tarihi: 3 Aralık 2022]

Dadgar, S. M. H., Araghi, M. S.M. ve Farahani, M., 2016, A Novel Text Mining Approach Based On TF-IDF And Support Vector Machine For News Classification, *IEEE International Conference on Engineering and Technology (ICETECH)*, 2016, pp. 112-116, doi.org/10.1109/ICETECH.2016.7569223.

Demir, F., 2022, Deep autoencoder-based automated brain tumor detection from MRI data, *Artificial Intelligence-Based Brain-Computer Interface*, In: Bajaj V., Sinha G.R. (ed.), Academic Press. 317-351

Di Gennaro, G., Buonanno, A. ve Palmieri, F. A. N., 2021, Considerations about learning, Word2Vec, *J Supercomput 77*, 12320–12335, doi.org/10.1007/s11227-021-03743-2

Gopani, A., 2022, *The History of Machine Learning Algorithms*, https://analyticsindiamag.com/the-history-of-machine-learning-algorithms/, [Ziyaret tarihi: 3 Aralık 2022]

Hajibabaee, P. et al., 2022 Offensive Language Detection on Social Media Based on Text Classification, *IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, 0092-0098, doi.org/10.1109/CCWC54503.2022.9720804.

Horat, S. I., Kara, K. C. ve Karakaş, A., 2019, Job Pre-Interview System with Artificial, Intelligence, *International Informatics and Software Engineering Conference (UBMYK)*, 1-4, doi.org/10.1109/UBMYK48245.2019.8965497

Huynh, T. V., Nguyen, K. V., Nguyen, N. L. ve Nguyen, A. G., 2020, Job Prediction: From Deep Neural Network Models to Applications, *RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, 1-6, doi.org/10.1109/RIVF48685.2020.9140760.

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. ve Brown, D., 2019, Text Classification Algorithms: A Survey, *Machine Learning on Scientific Data and Information*, 10 (4), 150, doi.org/10.3390/info10040150

Kınık, D. ve Güran, A., TF-IDF ve Doc2Vec Tabanlı Türkçe Metin Sınıflandırma Sisteminin Başarım Değerinin Ardışık Kelime Grubu Tespiti ile Arttırılması, *Avrupa Bilim ve Teknoloji Dergisi*, 21, 2021, 323-332

Khan, M. Y., Qayoom, A., Nizami, M. S, Siddiqui, M. S, Wasi, S. ve Syed, K. R. M, 2021, Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision Machine Learning, and Word Embedding-Based Deep Learning Techniques, *Complexity*, doi.org/10.1155/2021/2553199

Kolli, K., Charan, L., ve Puganuru, S., 2021, Classification of Genuinity in Job Posting Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, 9, doi: 10.22214/ijraset.2021.39580.

Kulkarni, J., Chong, D., Batarseh, F. A, 2020, *Foundations of data imbalance and solutions for a data democracy*, In: Batarseh, F. A, Yang R. (ed.), Data Democracy, Academic Press, 83-106, ISBN 9780128183663, doi.org/10.1016/B978-0-12-818366-3.00005-8.

Kurniawanda, M. R. ve Tobing, F. A. T, 2022, Analysis Sentiment Cyberbullying In Instagram Comments with XGBoost Method, *Ijnmt (Internatıonal Journal Of New Medıa Technology)*, 9 (1), doi.org/10.31937/ijnmt.v9i1.2670

LeCun, Y., Bengio, Y. ve Hinton, G., Deep learning, *Nature 521*, 436–444, doi.org/10.1038/nature14539

Lin, Y., Lei, H., Addo, P. C., ve Li, X., 2016, Machine learned resume-job matching solution, arXiv preprint arXiv:1607.07657.

Liu, Y., Zhang, G., Jin, X. ve Yuan, Ha., 2019 Multi-info Fusion Based Video Recommendation System. *Journal of Physics: Conference Series*, 1229, 012010, doi.org/10.1088/1742 6596/1229/1/012010.

Long, M. ve Yanqing, Z., 2015. Using Word2Vec to process big text data, *Big Data IEEE International Conference*, 2015, San Jose, CA, doi.org/10.1109/BigData.2015.7364114.

Luo, X., 2021, Efficient English Text Classification Using Selected Machine Learning Techniques, Alexandria Engineering Journal, 60 (3), 3401-3409, ISSN 1110-0168, doi.org/10.1016/j.aej.2021.02.009.

Mikolov, T., Chen, K., Corrado, G. ve Dean, J., Efficient Estimation of Word Representations in Vector Space, Computation and Language (cs.CL), doi:10.48550/ARXIV.1301.3781

Morde, V., 2019, *XGBoost Algorithm: Long May She Reign*, https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d, [Ziyaret tarihi: 4 Aralık 2022].

Naskar, A., 2020, *Continuous Bag of Words (CBOW) – Single Word Model – How It Works*, https://thinkinfi.com/continuous-bag-of-words-cbow-single-word-model-how-itworks/ [Ziyaret tarihi: 3 Aralık 2022].

Opuz, C., 2019, *Word2Vec*, https://www.veribilimiokulu.com/word2vec/, [Ziyaret tarihi: 25 Ekim 2022].

Otter, D. W., Medina, J. R. ve J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, 604-624, Feb. 2021, doi.org/0.1109/TNNLS.2020.2979670.

Patel, B., Kakuste, V. ve Eirinaki, M., 2017, CaPaR: A Career Path Recommendation Framework, IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), 23-30, doi.org/10.1109/BigDataService.2017.31

Phi, M., 2018, Illustrated Guide to LSTM's and GRU's: A step by step explanation, https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21, [Ziyaret tarihi: 3 Aralık 2022].

Ray, S., 2019, A Quick Review of Machine Learning Algorithms, *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, 35-39, doi:10.1109/COMITCon.2019.8862451.

Roy, P. K., Chowdhary, S. S. ve Bhatia, R., 2020, A Machine Learning approach for automation of Resume Recommendation system, *Procedia Computer Science*, 167, 2318-2327, ISSN 1877-0509, doi.org/10.1016/j.procs.2020.03.284.

Sakal, M., Makine Öğrenmesi Algoritmaları Kısa Açıklamaları, http://muratsakal.com/?p=230, [Ziyaret Tarihi: 20 Kasım 2022].

Security Team (Xantaro Deutschland GmbH), 2018, https://www.xantaro.net/en/techblogs/machine-and-deep-learning-cybersecurity/, [Ziyaret tarihi: 30 Ekim 2022].

Semberecki, P. ve Maciejewski, H., 2017, Deep learning methods for subject text classification of articles, *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, 357-360, doi: 10.15439/2017F414.

Shaw, R., 2017, *XGBoost: A Concise Technical Overview*, https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html, [Ziyaret tarihi: 25 Ekim 2022].

Song, P., Geng, C., ve Li, Z., 2019, Research on Text Classification Based on Convolutional Neural Network, *International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 2019, 229-232, doi: 10.1109/ICCNEA.2019.00052.

Talun, A., Drozda, P., Bukowski, L. ve Scherer, R., 2020, *FastText and XGBoost Content-Based Classification for Employment Web Scraping*, In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R. ve Zurada, J.M. (ed.), Artificial Intelligence and Soft Computing. ICAISC 2020. Lecture Notes in Computer Science(), 12416, Springer Cham, ISBN 978-3-030-61533-8, doi.org/10.1007/978-3-030-61534-5_39

Tran, H. T., Vo, H. H. P. ve Luu, S. T., Predicting Job Titles from Job Descriptions with Multi label Text Classification, *NAFOSTED Conference on Information and Computer Science (NICS)*, 2021, 513-518, doi.org/10.1109/NICS54270.2021.9701541

Vani, T., 2021, *Impetus to machine learning in cardiac disease diagnosis*, Image Processing for Automated Diagnosis of Cardiac Diseases, In: Chauhan K., Chauhan R. K. (ed.), Chapter 6, Academic Press, 99-116, ISBN 9780323850643, doi.org/10.1016/B978-0-323-85064-3.00009-1

Wang, J., Liu, T., Luo, X. ve Wang, L, 2018, An LSTM Approach to Short Text Sentiment Classification with Word Embeddings, *Conference on Computational Linguistics and Speech Processing ROCLING*, 2018, 214-223, doi.org/10.1016/j.future.2018.04.059

Wang, Z. ve Qu, Z., 2017, Research on Web text classification algorithm based on improved CNN and SVM, *IEEE 17th International Conference on Communication Technology (ICCT)*, 2017, 1958-1961, doi: 10.1109/ICCT.2017.8359971.

Yi,., Yang, C., Li, C., ve Zhang, Y., 2016, A job recommendation method optimized by position descriptions and resume information, *IEEE Advanced Information*

*Management, Communicates, Electronic and Automation Control Conference (IMCEC),* 2016, pp. 761-764, do.org/10.1109/IMCEC.2016.7867312.

Zhang, L., Wang, S. ve Liu, B., 2018 Deep learning for sentiment analysis: A survey, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discover*y, 8 (4), doi.org/10.1002/widm.1253