

**T.C.**  
**İSTANBUL AYDIN ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN**  
**ÖNERME SİSTEMİ**

**YÜKSEK LİSANS TEZİ**

**Ali SARIKAYA**

**Bilgisayar Mühendisliği Ana Bilim Dalı**  
**Bilgisayar Mühendisliği Bilim Dalı**

**AĞUSTOS, 2023**



**T.C.**  
**İSTANBUL AYDIN ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN**  
**ÖNERME SİSTEMİ**

**YÜKSEK LİSANS TEZİ**

**Ali SARIKAYA**  
**(Y2213.010010)**

**Bilgisayar Mühendisliği Ana Bilim Dalı**  
**Bilgisayar Mühendisliği Bilim Dalı**

**Tez Danışmanı: Prof. Dr. Ali GÜNEŞ**

**AĞUSTOS, 2023**

## **ONAY FORMU**

## ONUR SÖZÜ

Yüksek Lisans tezi olarak sunduđum “Derin Öğrenme Yöntemleri Kullanılarak Ürün Önerme Sistemi” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya’da gösterilenlerden oluştuđunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (.../08/2023)

Ali SARIKAYA

## ÖNSÖZ

Bu çalışmada Derin Öğrenme Yöntemleri Kullanılarak Ürün Önerme Sistemine ilişkin tüketici tutumunu belirleyen faktörlerin etkisi araştırılmıştır.

Tez çalışmamda planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Prof. Dr. Ali Güneş'e sonsuz teşekkürlerimi sunarım.

Ağustos 2023

Ali SARIKAYA

# DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN ÖNERME SİSTEMİ

## ÖZET

Makine öğrenmesi, klasik algoritma yapısı içerisinde mevcut olan şartta uygun şekilde aksiyon alma halleri için aksiyona ait parametrelerin algoritmalar vasıtası ile değişik değişkenler kullanılarak test edilmesi ile, ulaşılan neticeye göre modele ait verim düzeyini yükseltebilecek biçimde mimarisinin yenilenmesine imkân tanımaktadır. Algoritma, var olan mimari yapı kullanılarak çözümlenmeler gerçekleştirilir. Algoritma, gerçekleştirilen çözümlenmeler aracılığıyla sonuca varan parametrelerin değişmesi neticesinde hangi duyarlılık ile etkilediğini model haline getirmektedir.

Genel olarak değerlendirdiğimizde, makine öğrenmelerinde ileri düzey sinir ağları oluşturmaları ile modelleme kurgusu esnasında büyük çaplı verilerin işlenebilme özelliğine sahip algoritmalar geliştirilmiştir. Sanal ortamda potansiyel alıcıların ürüne ve hizmete olan ilgi ve alakasını, alıma etki eden etmenleri, verim miktarı, mala ait fiyat, kalite hareketlilikleri ile ilgili etkisini ve parametrelerde mevcut olan değişimlerin eyleme çevrilme olasılıklarını en doğru biçimde modelleyebilme uğraşlarında makine öğrenmesi algoritmaları da kullanılmaktadır.

Günümüz internet ortamında çeşitli sayfalarda reklamlarda kullanıcılara ürünler önerilmektedir. Bu ürün önermelerinde çeşitli parametreler yardımıyla, makine öğrenmesi aracılığıyla gerçekleşmektedir. Bu çalışmada ürün önerme sistemine yönelik literatür incelemesine yer verilmiş olup, bir ayakkabı firması için derin öğrenme yöntemi kullanılarak ürün önerme sistemi değerlendirilecektir. Bu kapsamda araştırmacı tarafından geliştirilen veri seti baz alınarak değerlendirilecektir.

**Anahtar Kelimeler:** Denetimli Öğrenme, Ürün Önerme Sistemi,

# **PRODUCT RECOMMENDATION SYSTEM USING DEEP LEARNING METHODS**

## **ABSTRACT**

Machine learning allows to renew the architecture in a way that can increase the efficiency level of the model according to the result achieved by testing the parameters of the action by using different variables through algorithms for the cases of taking action in accordance with the condition existing in the classical algorithm structure. Analyzes are performed using the algorithm and the existing architectural structure. The algorithm models the sensitivity with which it affects as a result of the changes in the parameters that reach the result through the analyzes carried out.

In general, algorithms with the ability to process large-scale data have been developed during modeling with the creation of advanced neural networks in machine learning. Machine learning algorithms are also used in the effort to model the interest and relevance of potential buyers to the product and service in the virtual environment, the factors affecting the purchase, the effect of the amount of efficiency, the price of the product, the quality movements, and the possibilities of turning the changes in the parameters into action in the most accurate way.

In today's internet environment, products are offered to users in advertisements on various pages. This is realized through machine learning, with the help of various parameters in product propositions. In this study, a literature review for the product recommendation system is included, and the product recommendation system will be evaluated for a shoe company by using deep learning method. In this context, it will be evaluated based on the data set developed by the researcher.

**Keywords:** Supervised Learning, Product Recommendation System,



# İÇİNDEKİLER

## Sayfa

ONUR SÖZÜ .....	i
ÖNSÖZ.....	ii
ÖZET.....	iii
ABSTRACT .....	iv
İÇİNDEKİLER .....	v
KISALTMALAR LİSTESİ.....	vii
ÇİZELGELER LİSTESİ.....	viii
ŞEKİLLER LİSTESİ.....	ix
<b>I. GİRİŞ .....</b>	<b>1</b>
A. Amaç, Kapsam ve Tanımlar .....	1
B. Amaç ve Kapsam .....	1
C. Tanımlar.....	2
<b>II. MAKİNE ÖĞRENMESİ.....</b>	<b>4</b>
A. Denetimli Öğrenme .....	7
1. Regresyon .....	7
2. Sınıflama.....	20
B. Denetimsiz Öğrenme .....	22
1. Kümeleme Analizi .....	22
2. Faktör Analizi .....	24
C. Takviyeli Öğrenme .....	24
D. Yarı Denetimli Öğrenme .....	25

E. İşbirlikçi filtreler (Collaborative Filtering) metodu.....	25
F. İçerik bazlı (Content Based) öneri metodu.....	25
G. İlgili Çalışmalar .....	26
<b>III. DERİN (PEKİŞTİRMELİ) ÖĞRENME .....</b>	<b>27</b>
A. Yapay Sinir Ağları (YSA) .....	29
B. Otomatik kodlayıcılar .....	31
C. Tekrarlayan yapay sinir ağları .....	33
<b>IV. DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN ÖNERME SİSTEMİ ÜZERİNE BİR UYGULAMA.....</b>	<b>34</b>
A. Öneri sistemi modelinin eğitilmesi.....	34
B. Bulgular .....	37
1. Veri setinin hazırlanması .....	38
2. Otomatik kodlayıcı modelinin oluşturulması.....	42
3. Öneri sistemi modelinin eğitilmesi .....	43
4. Optimizasyon algoritmaları .....	44
5. Giriş veri miktarının sistem başarısına etkisi.....	55
<b>V. TARTIŞMA VE SONUÇ .....</b>	<b>56</b>
<b>VI. KAYNAKLAR .....</b>	<b>57</b>
<b>ÖZGEÇMİŞ.....</b>	<b>60</b>

## KISALTMALAR LİSTESİ

- AKT** : Artık Kareler Toplamı
- CNN** : a convolutional neural network
- GPU** : Graphics processing unit
- IBM** :The International Business Machines Corporation
- YSA** : Yapay Sinir Ağları

## ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 1. Bağımlı, Bağımsız ve Tahmini Değişken .....	12
Çizelge 2. Bağımlı, Bağımsız ve Tahmini Değişken .....	13
Çizelge 3. Uzaklık Ölçüleri.....	23
Çizelge 4. Ayakkabı veri seti .....	37
Çizelge 5. Kullanıcı veri seti.....	38
Çizelge 6. Puan veri seti.....	38
Çizelge 7. GA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası .....	45
Çizelge 8. GA Algoritması 0,05 öğrenme oranı ve 600 devir ile test hatası .....	46
Çizelge 9. GA Algoritması 3 öğrenme oranı ve 25 devir ile test hatası .....	47
Çizelge 10. HGA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası.....	48
Çizelge 11. HGA Algoritması 0,05 öğrenme oranı ve 150 devir ile test hatası.....	49
Çizelge 12. HGA Algoritması 2,7 öğrenme oranı ve 25 devir ile test hatası.....	50
Çizelge 13. RmsProp Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası ..	52
Çizelge 14. RmsProp Algoritması 0,05 öğrenme oranı ve 100 devir ile test hatası	52
Çizelge 15. RmsProp Algoritması 1,8 öğrenme oranı ve 25 devir ile test hatası ....	53
Çizelge 16. Adam Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası .....	54

## ŞEKİLLER LİSTESİ

### Sayfa

Şekil 1.	Makine Öğrenimi Şeması.....	5
Şekil 2.	Makine Öğrenmesi Yöntemlerini.....	7
Şekil 3.	Bir YSA türü olan Otomatik Kodlayıcılar .....	32
Şekil 4.	Öneri Sistemi.....	34
Şekil 5.	Verilen şekilde algoritma akış şeması gösterilmektedir.....	36
Şekil 6.	İş akış şeması.....	37
Şekil 7.	Öneri sisteminin Otomatik Kodlayıcı ile modellenmesi.....	40
Şekil 8.	Otomatik Kodlayıcı giriş vektörü.....	41
Şekil 9.	W ve b parametresine göre maliyet fonksiyonunun değişimi.....	44
Şekil 10.	GA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası.....	46
Şekil 11.	GA Algoritması 0,05 öğrenme oranı ve 600 devir ile test hatası.....	47
Şekil 12.	GA Algoritması 3 öğrenme oranı ve 25 devir ile test hatası.....	48
Şekil 13.	HGA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası.....	49
Şekil 14.	HGA Algoritması 0,05 öğrenme oranı ve 150 devir ile test hatası.....	50
Şekil 15.	HGA Algoritması 2,7 öğrenme oranı ve 25 devir ile test hatası.....	51
Şekil 16.	RmsProp Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası.....	52
Şekil 17.	RmsProp Algoritması 0,05 öğrenme oranı ve 100 devir ile test hatası...	53
Şekil 18.	RmsProp Algoritması 1,8 öğrenme oranı ve 25 devir ile test hatası.....	54
Şekil 19.	Adam Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası.....	55

# I. GİRİŞ

## A. Amaç, Kapsam ve Tanımlar

Makine öğrenmelerinde algoritmaların karışık yapıya sahip ileri düzey sinir ağları oluşturmaları ile modelleme kurgusu esnasında büyük çaplı dataları işleyebilme yeteneği olan algoritmalar üretilmiştir. Alıcıların ürüne ve hizmet rağbetlerine olan alakasını, rağbet üzerine etki eden etmenleri, imalatçıların mallarında ulaşacakları verim miktarını, mala ait fiyat, kalite hareketliliklerinin aksiyon üzerine olan etkisini ve parametrelerde mevcut olan değişimlerin eyleme çevrilme olasılıklarını en doğru biçimde modelleyebilme uğraşlarında makine öğrenmesi algoritmaları da kullanılmaktadır.

Şirketlerin ciro imalatında başarı oranlarını en üst seviyeye çıkarmalarında, alıcı memnuniyetlerini giderme hususundaki verimlerini koruma ve yükseltme uğraşlarında, ürün ve hizmet arzlarındaki eksikliklerin giderilmesinde, rekabetçi taraflarını muhafaza etmede, uygulama verimlerinin daha iyi hale getirilmesinde seçilebilecek algoritma yapıları makine öğrenmesi yöntemleri ile kurgusu yapılabilmektedir. Finansal verilerde ise makine öğrenmesi faaliyetlerinden, her gün değişim gösteren piyasa koşulları ve hareketlerinin hesaplanması konusunda, finansal cihazların arz edilme süreçlerinde rağbetin ne düzeyde olacağını, arzın talebi giderip gideremeyeceğini, rağbetin farklı zaman periyotlarına göre arz işleminin başarılı olup olmama durumunun tespitinde faydalanılmaktadır.

## B. Amaç ve Kapsam

Makine öğrenmesi, klasik algoritma yapısı içerisinde mevcut olan şarta uygun şekilde aksiyon alma halleri için aksiyona ait parametrelerin algoritmalar vasıtası ile değişik değişkenler kullanılarak test edilmesi ile, ulaşılan neticeye göre modele ait verim düzeyini yükseltebilecek biçimde mimarisinin yenilenmesine imkân tanımaktadır. Algoritma, var olan mimari yapı kullanılarak

çözümlemeler gerçekleştirilir. Algoritma, gerçekleştirilen çözümlemeler aracılığıyla sonuca varan parametrelerin değişmesi neticesinde hangi duyarlılık ile etkilediğini model haline getirmektedir. Optimal modelde ise faaliyete alınır ve var olan model yenilenir. Klasik algoritmaların gruplandırılma prensibini kendiliğinden meydana getirmesini sağlaması gereksinimini gidermek için, ortaya koyulan makine öğrenmesi, parçaya ait yapı üzerinde inceleme yapma ve sonunda ise elde edilen verilerin analizini yaparak parçaları uygun olan ölçüde en iyi hale getirebilecek yapıya ulaşmışlardır. Neticeye varan yollara ait parametreleri tespit eden, parametre değişimlerin neticesi hangi boyutta etkilediği konusunda inceleme yapabilen ve var olan veriler vasıtası ile belirsizlikleri açıklayabilen ve öngörülerini doğru bir biçimde modelleme yaparken kullanabilme kapasitesine sahip olan makine öğrenmesi yöntemleri kullanılmaktadır. Mevcut verilerin ve varsayımların modelinin oluşturulması sırasında kullanılan önemli yöntemler içerisinde pekiştirmeli, gözetimli, mantıksal, gözetimsiz, lineer ve derin öğrenme metotları kullanılmaktadır.

Genel olarak değerlendirdiğimizde, makine öğrenmelerinde ileri düzey sınır ağları oluşturmaları ile modelleme kurgusu esnasında büyük çaplı verilerin işlenebilme özelliğine sahip algoritmalar geliştirilmiştir. Sanal ortamda potansiyel alıcıların ürüne ve hizmete olan ilgi ve alakasını, alıma etki eden etmenleri, verim miktarı, mala ait fiyat, kalite hareketlilikleri ile ilgili etkisini ve parametrelerde mevcut olan değişimlerin eyleme çevrilme olasılıklarını en doğru biçimde modelleyebilme uğraşlarında makine öğrenmesi algoritmaları da kullanılmaktadır. Günümüz internet ortamında çeşitli sayfalarda reklamlarda kullanıcılara ürünler önerilmektedir. Bu ürün önermelerinde çeşitli parametreler yardımıyla, makine öğrenmesi aracılığıyla gerçekleşmektedir. Bu çalışmada ürün önerme sistemine yönelik literatür incelemesine yer verilmiştir. Ayrıca çalışmanın uygulama kısmında bir ayakkabı firması için derin öğrenme yöntemi kullanılarak ürün önerme sistemi değerlendirilecektir. Bu kapsamda araştırmacı tarafından geliştirilen veri seti baz alınarak değerlendirilecektir.

### **C. Tanımlar**

**Derin Öğrenme:** Derin öğrenme, makine öğrenmesi sınıflarından biridir. Derin öğrenme, özellik çıkarma ve dönüştürme için birçok doğrusal olmayan

işlem birimi katmanını kullanılmaktadır. Bu sistemde yer alan her ardışık katman, kendisinden bir önceki katmanda yer alan çıktıyı girdi olarak baz almaktadır.

Makine Öğrenmesi: bilgisayarların açık bir biçimde programlanma yapmadan öğrenmesini sağlaması gerekli olan bir çalışma konusudur.



## II. MAKİNE ÖĞRENMESİ

“Arthur Lee Samuel” ilk defa 1959’da makine öğrenimi kavramını kullanmıştır. “Arthur Lee Samuel” IBM için hazırladığı dama oyununu dizayn ederken bu terimi ortaya atmıştır. Arthur Lee Samuel makine öğrenimi için; “bilgisayarların açık bir biçimde programlanma yapmadan öğrenmesini sağlaması gerekli olan bir çalışma konusudur” şeklinde bir ifade kullanmıştır.<sup>1</sup> 1962’de bir dama oyunu uzmanı olan Robert Nealey, IBM vasıtası ile imal edilen IBM7094 bilgisayarına karşı gerçekleştirilen yarışmada, oyunda bilgisayara karşı yenilmiştir. Dama oyunu oynamak için dizayn edilen program oldukça az miktarda bilgisayar hafızası bulundurduğundan tahtada yer alan taşların gerçekleştireceği hamleler için bir çeşit puanlama işlevi dizayn edildi. Puanlama görevinin esas amacı iki taraftaki bireyin de oyunu alma şansını hesaplamaktı. Programda bir sonraki hamlesinin seçimi yapılırken “minimax algoritması” tercih edilerek oyun taktiğini kullanmıştır. Oynanan oyunun çok daha iyi bir düzeye çıkması için Samuel yalnızca bununla kalmayıp, birtakım mekanizmalar da hazırladı. “Ezberci öğrenme” şeklinde isimlendirilen öğrenme yöntemi ile, program içerisine tüm pozisyonlar kaydedildi ve daha sonra bunları hatırlamasını sağladı. Bununla beraber Ödül fonksiyonunda yer alan değerler ile bir araya getirdi. Gerçekleştirilen bu çalışma ile Samuel aracılığıyla ortaya atılan “Makine Öğrenmesi” kavramı meydana geldi.<sup>2</sup>

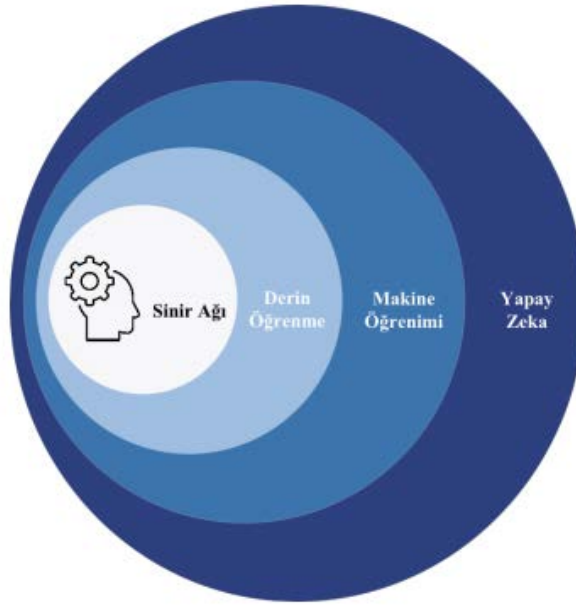
Makine öğrenmesi, klasik algoritma yapısı içerisinde mevcut olan şartta uygun şekilde aksiyon alma halleri için aksiyona ait parametrelerin algoritmalar vasıtası ile değişik değişkenler kullanılarak test edilmesi ile, ulaşılan neticeye göre modele ait verim düzeyini yükseltebilecek biçimde mimarisinin yenilenmesine imkân tanımaktadır. Algoritma, var olan mimari yapı kullanılarak

---

<sup>1</sup> Andreas MOCKENHAUPT (2021). Maschinelles Lernen." Digitalisierung und Künstliche Intelligenz in der Produktion. Wiesbaden: Springer Vieweg.

<sup>2</sup> Keith D. FOOTE (2019). A brief history of machine learning. dataversity web page, Homepage, Data education, Smart Data News, Articles, & Education.

çözümlemeler gerçekleştirilir. Algoritma, gerçekleştirilen çözümlemeler aracılığıyla sonuca varan parametrelerin değişmesi neticesinde hangi duyarlılık ile etkilediğini model haline getirmektedir. Optimal modelde ise faaliyete alınır ve var olan model yenilenir. Klasik algoritmaların gruplandırılma prensibini kendiliğinden meydana getirmesini sağlaması gereksinimini gidermek için, ortaya koyulan makine öğrenmesi, parçaya ait yapı üzerinde inceleme yapma ve sonunda ise elde edilen verilerin analizini yaparak parçaları uygun olan ölçüde en iyi hale getirebilecek yapıya ulaşmışlardır. Neticeye varan yollara ait parametreleri tespit eden, parametre değişimlerin neticesi hangi boyutta etkilediği konusunda inceleme yapabilen ve var olan veriler vasıtası ile belirsizlikleri açıklayabilen ve öngörülerini doğru bir biçimde modelleme yaparken kullanabilme kapasitesine sahip olan makine öğrenmesi yöntemleri kullanılmaktadır. Mevcut verilerin ve varsayımların modelinin oluşturulması sırasında kullanılan önemli yöntemler içerisinde pekiştirmeli, gözetimli, mantıksal, gözetimsiz, lineer ve derin öğrenme metotları kullanılmaktadır.



Şekil 1. Makine Öğrenimi Şeması

Şekil 1’de de gösterildiği şekilde, makine öğrenmesi esasında yapay zekâ ve bunun yanı sıra da derin öğrenme ile alakalıdır. Yapay zekânın meydana gelebilmesi için makine öğrenmenin gerçekleşmesi mecburidir. Veri grubunu bir model haline getirmek için makine öğrenimi en doğru algoritma yetisidir. Makine

öğrenimi tekniğinin istatistiksel metotların da kullanılması ile, gruplandırma ve öngörülerde bulunulması için algoritmalar düzenlenirler. Geniş veriler büyüyüp daha geniş hale gelse bile makine öğreniminin fonksiyonsuzlaşması yerine, çok daha iyi öngörülerde bulunulması ve neticelerin daha kesin hale gelmesini sağlar.

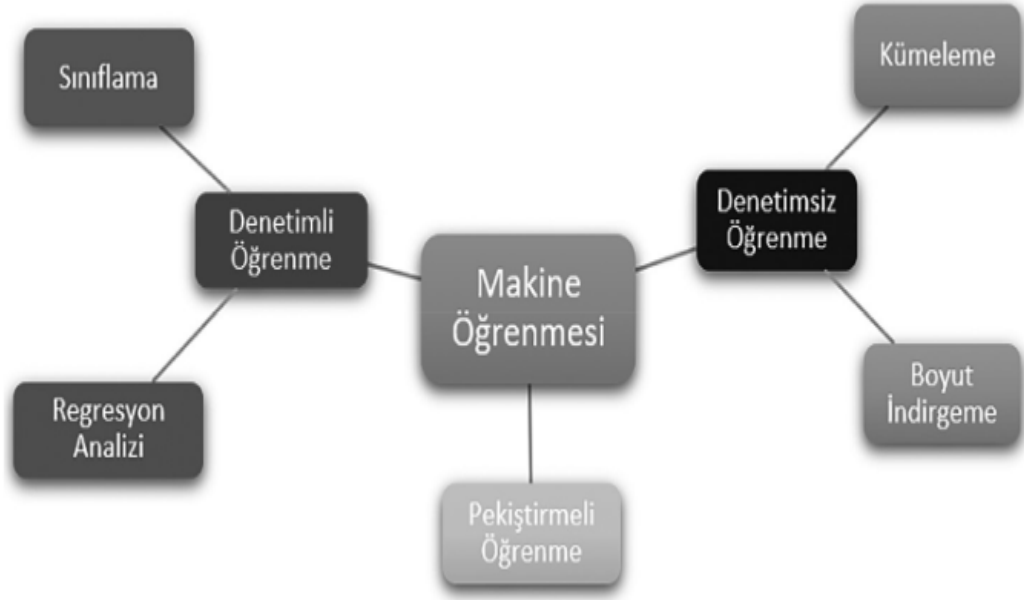
Makine öğrenmelerinde algoritmaların karışık yapıya sahip ileri düzey sinir ağları oluşturmaları ile modelleme kurgusu esnasında büyük çaplı data'ları işleyebilme yeteneği olan algoritmalar üretilmiştir. Alıcıların ürüne ve hizmet rağbetlerine olan alakasını, rağbet üzerine etki eden etmenleri, imalatçıların mallarında ulaşacakları verim miktarını, mala ait fiyat, kalite hareketliliklerinin aksiyon üzerine olan etkisini ve parametrelerde mevcut olan değişimlerin eyleme çevrilme olasılıklarını en doğru biçimde modelleyebilme uğraşlarında makine öğrenmesi algoritmaları da kullanılmaktadır. Şirketlerin ciro imalatında başarı oranlarını en üst seviyeye çıkarmalarında, alıcı memnuniyetlerini giderme hususundaki verimlerini koruma ve yükseltme uğraşlarında, ürün ve hizmet arzlarındaki eksikliklerin giderilmesinde, rekabetçi taraflarını muhafaza etmede, uygulama verimlerinin daha iyi hale getirilmesinde seçilebilecek algoritma yapıları makine öğrenmesi yöntemleri ile kurgusu yapılabilmektedir. Finansal verilerde ise makine öğrenmesi faaliyetlerinden, her gün değişim gösteren piyasa koşulları ve hareketlerinin hesaplanması konusunda, finansal cihazların arz edilme süreçlerinde rağbetin ne düzeyde olacağını, arzın talebi giderip gideremeyeceğini, rağbetin farklı zaman periyotlarına göre arz işleminin başarılı olup olmama durumunun tespitinde faydalanılmaktadır.<sup>3</sup>

Başlıca makine öğrenmesi yöntemlerini aşağıdaki gibi gruplandırmak mümkündür.<sup>4</sup>

---

<sup>3</sup> Selahaddin Bilal ÖZGÜR. Algoritmalar, Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Uygulamaları: Beşeri Fayda Üretiminin Yazılımlar Tarafından Karşılanması. Ekonomi ve Yönetim Araştırmaları Dergisi, 10(1), 1-29.

<sup>4</sup> Pelin CANBAY (2021). Sağlıkta Yapay Zeka: Makine Öğrenmesi Yöntemleri Ve Uygulamaları.



Şekil 2. Makine Öğrenmesi Yöntemlerini

## A. Denetimli Öğrenme

Denetimli öğrenme makine öğrenimi içerisinde en sık tercih edilen ve en çok tanınan tekniktir. Eğitim etiketlerinin her biri bir çıktı etiketi oluşmasına yol açar. Etiketli datalar ile sistem öğretilir. Eğitim vasıtası ile sistem problemleri öğrenir. Gerçekleştirilen eğitimler kelimeler, resimler gibi manuel etiketler ile beslenmesi sağlanır.<sup>5</sup> Bazı kaynaklarda denetimli öğrenme, “gözetimli öğrenme” şeklinde de isimlendirilmektedir. Öngörüler gerçekleştirilirken, sınıflandırma veya regresyon tekniğinden yararlanılmaktadır. Gelmiş olan mailin spam olduğunu belirlemek için bu metot kullanılacağı zaman, spam şeklinde gruplandırılan e-postalar kümelenecek spam olarak saptanacak oldukça önemli örnekler makine öğrenimi algoritması beslenmesi mecburidir.

Denetimli öğrenme iki grupta incelenebilir. Bunlar; Regresyon ve Sınıflandırmadır.

### 1. Regresyon

“Regression” ingilizce bir kavramdır ve kelime anlamına bakıldığında, istatistik biliminde yer alan “alışılmışa doğru çekilme” anlamını karşılamaktadır.

<sup>5</sup> Çoban, Ö. (2016). Metin sınıflandırma teknikleri ile türkçe twitter duygu analizi. Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü.

Bugün kullanılmakta olan manası ile “regression” terimi; bağımlı değişkeni, değişik bağımsız değişkenlerle ilişki içerisine sokarak öngörü veya çıkarım yapma amacı güden istatistiksel bir tekniktir. Mevcut iki ya da ikiden fazla değişken içerisindeki ilişkiyi tespit etmek ve söz konusu ilişki vasıtası ile o husus ile ilişkili önseziler veya kestirimler oluşturabilmek amacı ile gerçekleştirilir.

Başka bir deyişle istatistiksel manada iki değişken arasında mevcut olan ilişki, değişkenlere atfedilen değerlerin karşılıklı biçimdeki değişimleri içerisinde bir bağımlılık şeklinde algılanır. Aslında X değişkeninin atfedilen değeri değişirken bu duruma bağlı biçimde Y'nin değerinde de bir değişim oluyorsa, X ve Y arasında bir ilişkinin mevcut olduğu ifade edilebilir.

Regresyon analizi gerçekleştirilirken kullanılan değişkenler bağımlı ve bağımsız olmak üzere iki ayrı başlığa ayrılmak mecburiyetindedir. Bağımlı olarak nitelendirilen değişken, bağımsız değişken ya da değişkenler aracılığı ile açıklanan değişken olarak tanımlanabilir. Bağımlı değişken diğer bir değişkeni bulmak ya da tahmin etmek aracılığı ile kullanılan değişken şeklinde de açıklanır. Bağımlı değişken bağımsız olan değişkenin yaşadığı değişimlerin etkisini gören ve bağımsız değişkenin verileri vasıtasıyla öngörülme çalıřılan değişkendir. Başka bir deyişle bağımlı değişken etkilenen ya da açıklanan değişken şeklinde de ifade edilebilir. Bağımlı değişken genellikle Y ile temsil edilir. Y, X'e bağılı biçimde değişim gösterebilen veya X' den etkilenebilen değişkendir. Çoğunlukla bağımsız olan değişken ya da değişkenler X ( $X_1, X_2, \dots, X_n$ ) biçiminde, bağımlı değişken ise Y biçiminde ifade edilir. Etkileyici ve açıklayıcı değişken olarak tanımlanan değişken ise bağımsız değişkendir.

Dağılım grafiđi bağımsız ve bağımlı değişkenlerin kendi içlerinde birbirlerine kıyasla aldıkları değerleri ortaya koyan grafiđtir. Dağılım grafiklerinde çoğunlukla x eksenini bağımsız değişkeni temsil ederken, y eksenini bağımlı değişkeni temsil eder. İki veya ikiden fazla değişken içerisinde bir ilişkinin mevcut olup olmadığını ölçen ve bu durumu eğrisel olmakla beraber doğrusal biçimde de açıklayan denklem modelleri ise regresyon şeklinde isimlendirilir. Denklemlerde bağımlı değişken çoğunlukla bir tanedir. Fakat bağımsız değişken miktarı birden fazla sayıda olabilir. Yalnızca bir adet bağımsız değişkenin mevcut olduğu denklemler “Basit Doğrusal Regresyon” şeklinde

isimlendirilir. Denklemden iki ya da ikiden çok bağımsız değişken mevcut ise bu kez “Çoklu Doğrusal Regresyon” şeklinde isimlendirilir.

Bağımlı olan değişken ile bağımsız değişken arasında mevcut olan ilişkinin açıklanması regresyon analizinin amaçlarından bir tanesidir. Örnek olarak X ile Y arasında aşağıda verilen  $Y_i = \alpha + \beta X_i + \varepsilon_i$  ( $i=1, 2, 3, \dots, n$ ) şeklinde doğrusal olarak değerlendirilen bir ilişki olduğu tahmin ediliyor ise burada yapılması gereken ilk şey  $\alpha$  ve  $\beta$  parametrelerinin bulunmasıdır. Bu modelde yer alan ve bilinmeyen parametreler ön görüldüğünde bağımsız değişkenlerin alabileceği farklı değerler için, bağımlı değişkene verilebilecek değerleri bulmaya çalışmak ise regresyon analizine ait diğer bir amaçtır. Bağımsız değişkenlerin farklı olan değerlerinin her bir tanesi için bağımlı değişken değişmeyen yani sabit bir değere sahipse ortada çözülecek herhangi bir problem bulunmamaktadır.

Regresyon analizinde olduğu gibi varyans çözümlemesinde de bağımlı değişken minimum eş aralıklı seviyede ölçülür. Regresyon analizi yapılırken, birtakım özel haller haricinde, bağımsız değişken ya da bağımsız değişkenler de minimum eşit aralıklı seviyede ölçülmelidir. Varyans çözümlemesi gerçekleştirilirken ise yine bağımlı değişken minimum eş aralıklı seviyede ölçülür. Varyans çözümlemesinde bağımsız değişken ya da değişkenler gruplama veya sıralama ölçme seviyesinde ölçülür. Dolayısıyla regresyon analizi yapılırken ve varyans çözümlemesi esnasında esas amaç neredeyse aynıdır. Bununla birlikte hangisinin tercih edileceği bağımsız değişken ya da değişkenlerin ölçme seviyesine bağımlı biçimde değişiklik gösterir.

Regresyon analizi, bağımsız değişken sayısına göre;

- Basit regresyon analizi (Tek bağımsız değişken)
- Çoklu regresyon analizi (Birden fazla bağımsız değişken)

Fonksiyon tipine göre;

- Doğrusal regresyon analizi
- Doğrusal olmayan regresyon analizi (Eğrisel)

Verilerin kaynağına göre;

- Ana kütle verileriyle regresyon analizi

- Örnek verileri ile regresyon analizi
- Zaman serilerinde regresyon analizi (Eşleştirilmiş zaman serileri) şeklinde gruplandırılır.

Regresyon analizi gerçekleştirilirken mevcut değişkenler arasında yer alan ilişki bir fonksiyon ile özetlenir. Bu nedenle söz konusu yapı kapalı fonksiyon biçiminde  $Y = f(X)$  denklemi ile ifade edilir. Regresyon eşitliklerinde bağımlı ve bağımsız değişken haricinde hata terimleri de bulunmaktadır. Bu durum regresyon analizini diğer matematiksel fonksiyonlardan ayırmaktadır. Bağımsız ve bağımlı değişkenler ile ilişkili veriler toplanırken, birtakım sebepler dolayısıyla meydana gelen terimlere ait değerler yalnızca denklem içerisindeki katsayıların bulunmasının ardından hesaplanacaktır. Dolayısıyla regresyon denkleminin geçerli olabilmesi, hata terimi değerlerine ilişkin altta verilmiş olan varsayımların geçerliliğine bağlıdır.

- Denklemdaki hata teriminin rasgele bir değişken olması gereklidir.
- Hata terimlerinden beklenen değer 0'a eşit olmalıdır. Diğer bir deyişle hata terimlerine ait ortalama sifira eşit olmalıdır  $E(\epsilon) = 0$
- Hata terimleri 0 çevresinde seyreden normal düzeyde bir dağılım göstermelidir.
- Mevcut hata terimleri içerisinde bir ilişki söz edilmemelidir. Yani hata terimlerine ait kovaryanslar sifira eşit olmalıdır.  $Cov(\epsilon_i, \epsilon_j) = 0, i \neq j$ .
- Hata terimlerine ait varyans değeri sabit bir değere eşit olmalıdır. Diğer bir deyişle örneklem her biri için varyans eşit olmalıdır.  $Var(\epsilon) = a$ , a sabit sayı.
- Bağımlı değişken tesadüfiyken bağımsız değişken rastgele olmamalıdır.

mevcut iki değişkenin arasında yer alan ilişki doğrusal bir ilişki ise bu durumda fonksiyon  $Y = \beta_0 + \beta_1 X + \epsilon$  şeklinde gösterilir. Verilen denklemde;

Y: Bağımlı değişken

X: Bağımsız değişken,

$\beta_0$ :  $X=0$  olduğunda bağımlı değişkenin alacağı değer (kesim noktası)

$\beta_1$ : Regresyon katsayısı

$\varepsilon$  : Hata terimi (Ortalaması = 0 ve Varyansı =  $\sigma^2$ 'dir)

Regresyon Katsayısı ( $\beta_1$ ) : Bağımsız değişken değerinde meydana gelecek bir birime tekabül eden bir değişikliğin, bağımlı olan değişkenin değerinde meydana getireceği ortalama değişimi miktarı olarak gösterilmektedir.

Regresyon denklemini meydana getiren parametrelerin öngörülmesi amacıyla birbirinden değişik teknikler kullanılmaktadır. Uygulama sırasında çoğunlukla en küçük kareler tekniği tercih edilir. En küçük kareler yönteminde ifade edilene göre, X ve Y tesadüfi değişkenlerinin sahip olduğu değerlerden meydana gelen serpilme diyagramı kapsamında bulunan noktalar üzerinden geçirilerek çizilen doğrunun Y bağımlı değişkeninin aldığı değerler ile tahmin denklemi vasıtası ile ortaya çıkarılacak teorik değerler arasında yer alan farklar minimuma çekilmek için uğraşılır. Tahmin denklemi aracılığı ile bulunacak olan teorik değerlerden meydana gelen tesadüfi değişken öngörüsü Y biçiminde isimlendirilir ve  $\hat{Y}$  şeklinde gösterilir. Artık veya hata terimi kavram ise esas Y değişkeniyle tahmini Y arasındaki farkları ifade etmektedir. En küçük kareler tekniği tarafından belirtildiğine göre, serpilme diyagramı içerisinde bulunan noktalar içerisinde sınırsız sayıda doğru çizilebilse dahi ilişkiyi en doğru biçimde açıklayan doğru, bağımlı değişkene ait gözlemlenen değerler ile ön görülen değerler arasında bulunan fark oranını minimuma çeken doğrudur.<sup>6</sup> Bahsedilen tekniğin kullanılmasıyla regresyon denklemi kapsamında bulunan katsayıların değerleri formül haline getirilebilir. X bağımsız, Y ise bağımlı değişkeni temsil eder. Bu değişkenler için n adet gözlem değerinin mevcut olduğu düşünelim. Böyle bir durumda X ve Y'nin altta verilmiş olan tablo içerisinde bulunan değerleri aldığını düşünelim. Öngörülen Y için hazırlanacak denklemin de  $aX + b$  biçimde olduğunu düşünersek;

---

<sup>6</sup> Dilek ALTAŞ (2013), İstatistiksel Analiz, 1. Baskı-Şubat,



Çizelge 1. Bağımlı, Bağımsız ve Tahmini Değişken

Y	X	$\hat{Y} = aX+b$
$Y_1$	$X_1$	$aX_1 +b$
$Y_2$	$X_2$	$aX_2 +b$
$Y_3$	$X_3$	$aX_3 +b$
...	...	...
$Y_n$	$X_n$	$aX_n +b$

En küçük kareler tekniğinde asıl Y ile tahmini Y değerlerinin arasında bulunan farkı en düşük seviyeye çekmek istenir. Diğer bir deyişle, iki değer arasında yer alan fark miktarı minimize edilmek istenmektedir. Söz konusu durumda öncelikli olarak, çizelge 1’de bulunan esas Y ile  $\hat{Y}$  değerlerini birbirinden çıkartarak bir fark elde edelim. Değişkenler totalde n kadar değer alır. Dolayısıyla bütün farkların toplamının en düşük seviyeye çekilmesi gerekmektedir. Yani, ilk olarak değerlerden her biri için artıkları, hata terimlerini bir diğer ifade ile ön göremediğimiz bölümlerin hesabı yapılır. Bunun ardından söz konusu değerlerin toplam değerini en aza indirgeyecek a ve b katsayılarının bulunması adımına geçilir. Bulunan toplam değerini minimum düzeye indirirken pozitif ve negatif farklar özellikle itina gösterilmesi gereken adımlarda bir tanesidir. Nitekim, Y değişkenine atfedilen değerini 3 olduğunu düşünelim. Analiz neticesinde ise söz konusu değeri 2 olarak öngördüğümüzü varsayalım. Diğer bir gözlemden ise Y’ye atfedilen değer 6 olduğunu, bununla beraber analiz neticesinde bu değeri 7 şeklinde ön gördüğümüzü düşünelim. Bahsettiğimiz ilk durum için asıl Y ile  $\hat{Y}$  ‘nin çıkarılması ile elde edilen fark 3-2 işleminden 1 olarak bulunur. Söz edilen ikinci durumda ise asıl Y değeri ile tahmini  $\hat{Y}$  değerinin çıkarılması ile 6-7 işlemi gerçekleştirilir ve neticede -1 sonucu elde edilir. Söz konusu problemi çözme tekniklerinden bir tanesi de artıkların karelerinin toplam değerini minimum seviyeye çekmektir. Çizelge 2’de verilen listede tahmini, bağımlı, ve bağımsız değişken ve bunların yanı sıra artıkların meydana getirdiği değişken ile artıkların karesinden meydana gelen değişken bulunmaktadır.

Çizelge 2. Bağımlı, Bağımsız ve Tahmini Değişken

Bağımlı Değişken	Bağımsız Değişken	Tahmini Y	Artık	Artıkların Kareleri
Y	X	$\hat{Y} = aX+b$	$Y-\hat{Y}$	$(Y-\hat{Y})^2$
$Y_1$	$X_1$	$aX_1 +b$	$Y_1- aX_1 +b$	$(Y_1- aX_1 +b)^2$
$Y_2$	$X_2$	$aX_2 +b$	$Y_2- aX_2 +b$	$(Y_2- aX_2 +b)^2$
$Y_3$	$X_3$	$aX_3 +b$	$Y_3 - aX_3 +b$	$(Y_3 - aX_3 +b)^2$
...	...	...	...	...
$Y_n$	$X_n$	$aX_n +b$	$Y_n - aX_n +b$	$(Y_n - aX_n +b)^2$

değerinin sıfıra yakın olması gerekir. Buradaki toplam değeri artıkların karelerinden elde edilen toplam olduğu değeri olduğundan AKT biçiminde ifade edilmektedir. Söz konusu durumda AKT'nin en az olması için gereken a ve b değerlerinin hesaplanması gerekmektedir. Yani, iki bilinmeyenli olan bir fonksiyonda kritik değerler olarak ifade edilen değerler bulunmalıdır. Bahsedilen değerleri bulabilmek için ilk olarak fonksiyonun bilinmeyen değerlere göre türevini hesaplamalıyız. Bulunan değer sıfıra eşitlendiğinde elde edilen değerler fonksiyona ait kritik değerleri göstermektedir.

$$AKT = (Y_1 - aX_1 - b)^2 + (Y_2 - aX_2 - b)^2 + (Y_3 - aX_3 - b)^2 + \dots + (Y_n - aX_n - b)^2 \quad \text{formül 1}$$

En küçük kareler tekniği aracılığı ile regresyon denklemine ait katsayıları hesaplamak için altta verilmiş olan 3. stepten faydalanılır.

**1.Adım:** AKT fonksiyonunun a'ya göre türevini alıp sıfıra eşitleyelim.

**2.Adım:** AKT fonksiyonunun b'ye göre türevini alıp sıfıra eşitleyelim.

**3.Adım:** (1).ve (2). adımlarda bulunan denklemleri birlikte çözerek a ve b değerlerini hesaplayalım.

**1.Adım:** AKT fonksiyonunun a'ya göre türevini alıp sıfıra eşitleyelim.

$$AKT = (Y_1 - aX_1 - b) + (Y_2 - aX_2 - b)^2 + (Y_3 - aX_3 - b)^2 + \dots + (Y_n - aX_n - b)^2 \quad \text{formül 2}$$

AKT değerini toplam sembolü kullanarak tekrar yazarsak,  $AKT = \sum_{k=0}^n (Y_k - aX_k - b)^2$  eşitliği sağlanır. AKT'nin türevini alıp sıfıra eşitlersek, sırayla aşağıdaki eşitlikler elde edilir.

$$\frac{dAKT}{da} = \sum_{k=0}^n 2(Y_k - aX_k - b)(-X_k) = 0 \quad \text{formül 3}$$

$$2 \sum_{k=0}^n (Y_k - aX_k - b)(-X_k) = 0 \text{ formül 4}$$

$$\sum_{k=0}^n (Y_k - aX_k - b)(-X_k) = 0 \text{ formül 5}$$

$$\sum_{k=0}^n Y_k - aX_k - b)(-X_k) = 0 \text{ formül 6}$$

$$\sum_{k=0}^n (aX_k Y_k - aX_k X_k + bX_k) = 0 \text{ formül 7}$$

$$\sum_{k=0}^n (X_k X_k + bX_k) = \sum_{k=0}^n X_k Y_k \text{ formül 8}$$

$$\sum_{k=0}^n X_k^2 + b \sum_{k=0}^n X_k = \sum_{k=0}^n X_k Y_k \text{ formül 9}$$

Üstte verilmiş olan denklemin taraflarından her ikisi de değişkenin aldığı total değer sayısına yani gözlem sayısına bir diğer ifade ile n değerine bölünürse, altta verilen eşitliğe ulaşılır.

$$aE(X^2) + b E(X) = E(XY) \text{ formül 10}$$

**2. Adım:** AKT fonksiyonunun b'ye göre türevini alıp sıfıra eşitleyelim.

$$AKT = \sum_{k=0}^n (Y_k - aX_k - b)^2 \text{ formül 11}$$

$$\frac{dAKT}{db} = \sum_{k=0}^n 2(Y_k - aX_k - b)(-1) = 0 \text{ formül 12}$$

$$2 \sum_{k=0}^n (Y_k - aX_k - b)(-1) = 0 \text{ formül 13}$$

$$\sum_{k=0}^n (Y_k - aX_k - b)(-1) = 0 \text{ formül 14}$$

$$\sum_{k=0}^n (-Y_k + aX_k + b) = 0 \text{ formül 15}$$

$$\sum_{k=0}^n aX_k + \sum_{k=0}^n b = \sum_{k=0}^n Y_k \text{ formül 16}$$

Yukarıdaki eşitliğin her iki tarafını gözlem sayısına yani değişkenin aldığı toplam değer sayısına, yani n'e bölersek, aşağıdaki eşitlik elde edilir.

$$a E(X) + b = E(Y) \text{ formül 17}$$

Üstteki eşitlikten E(Y)-a E(X) b katsayısı şeklinde ifade edilir. Bu sayede, b katsayısının bulabilmesi için ve a katsayısının değerine ve Y ve X tesadüfi değişkenlerinin beklenen değerlerine gereksinim olduğu görülür.

**3.Basamak:** 1.basamakta bulunan  $aE(X^2)+bE(X) = E(XY)$  ve 2. Basamakta bulunan  $b=E(Y) - aE(X)$  eşitliklerini beraber çözümlenerek a değerini bulalım.

$aE(X^2) + bE(X) = E(XY)$  eşitlikte b değerini getirip kendi yerine koyarsak sıra ile alttaki denkleme ulaşılır.

$$aE(X^2)+(E(Y)-aE(X))E(X) = E(XY) \text{ formül 18}$$

$$aE(X^2)+E(Y)E(X)-aE(X)^2 = E(XY) \text{ formül 19}$$

$$aE(X^2)+ -aE(X)^2 = E(XY)- E(Y)E(X) \text{ formül 20}$$

$$a(E(X^2)+ -E(X)^2 )= E(XY)- E(Y)E(X) \text{ formül 21}$$

$$a \text{ Var}(X)= \text{Kov}(X,Y) \text{ formül 22}$$

$$a = \frac{\text{Kov}(X,Y)}{\text{Var}(X)} \text{ formül 23}$$

Verilen bilgiler ışığında basit regresyon denkleminde ait katsayılar altta verilen formüller vasıtasıyla bulunur.

$$a = \frac{\text{Kov}(X,Y)}{\text{Var}(X)} \text{ ve } b = E(Y) - aE(X) \text{ formül 24}$$

Regresyon denklemi kapsamında bulunan parametrelere ait yorum ise altta verilen biçimde kısaca özetlenebilir.

a: Doğruya ait eğim miktarını temsil etmektedir. Yani bağımsız değişkende yaşanan bir birime karşılık gelen herhangi bir değişmeye karşın bağımlı değişken değerinde gözlemlenen değişme oranını ifade eder.

b: Dikey eksenin doğru tarafından kesildiği yeri tanımlar. Diğer bir ifade ile bağımsız değişkene ait değerlerin sıfıra eşit olması halinde bağımlı olarak değerlendirilen değişkenin eşit olacağı değeri ifade eder.

Regresyon denklemi vasıtası ile ulaşılan bağımlı değişkene ait değerler kesin değildir. Bu değerler asıl değerler ile farklı olabilirler. Görsel biçimde ise serpilme diyagramı üzerinde Y değişkeni ile X değişkenlerinin sahip oldukları karşılıklı değerlerinin regresyon doğrusu olarak isimlendirilen doğru üzerinde bulunmayıp, bu doğrunun çevresinde bulunması halidir. Serpilme diyagramı üzerindeki doğruya yaklaşması öngörülerin doğruluğunun arttığını gösterir. Yani, öngörülere ait standart hata daha küçük hale gelmektedir. Bu noktada standart hata kavramı standart sapma ile aynı anlamı karşılamaktadır.

En küçük kareler tekniği aracılığı ile bulunan regresyon doğrusuna ait katsayılar, a ve b şeklinde temsil edilen değerler spesifik bir örneklem ile bulunduğu için değişik örneklem için değişik sonuçlara ulaşılabilir. Fakat a ve b ana kütle regresyon denklemi içerisinde bulunan  $\beta_0$  ve  $\beta_1$  katsayıları çevresinde yayılım gösterirler. Dolayısıyla ulaşılan değerler için güven aralıkları yaratmak gereklidir.

### **\*\* Kukla değişken**

Regresyon analizi kapsamında kullanılmakta olan bağımsız ve bağımlı değişkenler kimi zaman nicel özellikte olurken genellikle nitel özellikte de bulunabilir. Nicel özelliğe sahip değişkenler çoğunlukla zaman serileri şeklinde de isimlendirilir. Örnek olarak bir devletin geride kalan son kırk senesi için işsizlik oranları, fiyatlar, ev ücretleri ve buna benzerleri gibi... Fakat, uygulama esnasından genellikle nitel özellikteki değişkenler de regresyon analizi kapsamına alınmak durumunda kalınabilir. Nitel değişkenlerin ikili yapıları vardır. Kadın/erkek, eve sahip olan/evi olmayan, beyaz/beyaz haricindekiler (yalnızca siyah değil, beyaz renk dışında kalan diğer bütün renkler), yabancı/yerli, ölüm cezası bulunan/bulunmayan devletler şeklide.

Özet olarak, nitel değişkenler genellikle kişinin dini, dili, kuzey, güney, doğu veya batı gibi dünyaya geldiği bölge, cinsel kimliği, ırkı, eğitim seviyesi şeklinde sıralanabilecek olan durumlar üzerinde inceleme ve karşılaştırma

yapmak için tercih edilen deęişken türleridir. Bütün bu sözü edilen deęişkenler nitel tarzda enformasyon bulundurlar.

Örnek olarak bir ankete dahil olan kişilerin cinsiyetini belirlemek amacı ile kullanılan deęişken, 1 ya da 0 deęeri verilerek meydana getirilir. Gruplardan bir tanesini 1 ötekisini de 0 şeklinde tanımlamamız gerekmektedir. Kategorilerden hangisinin 0 hangisinin ise 1 olduęu regresyon analizi neticesinde elde edilecek sonucu deęiştirmeyecektir. Fakat yorumlanabilmesi için kategorilerden hangisinin 1 hangisinin 0 olduęunun bilinmesi gerekmektedir. Nitel grupları tespit etmek için esasında 1 ve 0'dan hariç bambaşka iki deęerden de faydalanılabilir. Fakat 1 ve 0 şeklinde ifade edildiklerinde katsayıların yorumlanması oldukça kolay hale gelir.

Bernoulli daęılım ise tıpkı bu sözü edilen deęişkene benzer biçimde tanımlanan başka bir rastgele deęişken daęılımıdır. Bu daęılım da dięerleri gibi yalnızca 1 ya da 0 deęerlerini almaktadır. Fakat bu daęılımda parametre deęeri (p) olarak ifade edilen bir deęer bulunmaktadır. Ve bu p deęeri 1 olma ihtimali gösterir. P'nin 1'e eşit olması halinde deęişkenin 0'a eşit olma ihtimali (1-p) biçiminde ifade edilir. Özet olarak, p parametreyle Bernoulli daęılım gösteren X rastgele deęişkeni p ihtimalle 1 ve (1-p) ihtimalle 0 deęerine eşit olur (p olasılık deęerini ifade ettięinden, p daima 0 ile 1 arasında yer alır).

Özet olarak, yükseköğrenim eğitimi alma, cinsel kimlik, bir bölge içerisinde hayatını sürdürme, herhangi bir makinenin çalışması şeklinde sıralanabilecek olan durumlar için tanımlanmış olan deęişkenler nitel deęişkenler olarak adlandırılır. Herhangi bir regresyon modeli içerisinde nitel deęişkenlerden bağımsız deęişken olarak faydalanılabilir. Bu tarz deęişkenlere arzuya göre 1 ya da 0 deęerleri atfedilerek var olan durum gözlem şeklinde kayıt altına alınır ve deęişken kazanılmış olur. Bu tarz deęişkenler ise kukla dięer bir adı ile gölge deęişken şeklinde isimlendirilir. Altta bağımsız deęişkeni kukla deęişken olan bir regresyon denklemi bulunmaktadır.

$$Y = \beta_0 + \beta_1 K \text{ formül 25}$$

Söz konusu olan model içerisinde Y şirketlere ait aylık gelir miktarlarını tanımlayan deęişken, K ise kukla deęişken şeklinde isimlendirdiğimiz bahsi geçen şirketlerin ülkemizde bulunup bulunmadığını ifade etsin. Eğer şirket

lkemizde bulunuyorsa  $K_1$ 'e eŒit olsun. Œirketin lkemizde olmaması halinde ise 0'a eŒit olsun. En kk kareler tekniđi kullanılarak alttaki katsayıların hesaplandığını dŒnelim.

$$Y = 280 + 100K$$

stte verilen eŒitlikte  $K=0$  durumunda, diđer bir deyiŒle Œirket lkemizde bulunmuyorsa, Œirkete ait aylık gelir miktarı 280 birime eŒit olacaktır. Sz konusu deđer lkemizde bulunmayan Œirketlerin ay bazındaki ortalama gelir miktarlarını tanımlar. lkemizde bulunan Œirketlere ait aylık gelir miktarlarını bulabilmek iin  $K$  yani kukla deđiŒkeninin  $1$ 'e eŒit olduđu halin zerinde durmamız gerekmektedir.  $K$  deđerinin  $1$ 'e eŒit olması halinde  $Y$  deđerı 380'e eŒit olacaktır. Bu sayede, lkemizde yer alan Œirketlere ait aylık ortalama gelir miktarları da 380 birime eŒit olacaktır.

Kukla deđiŒkeni yalnızca ikili durumlarda kullanılmaz. Bunun yanı sıra birden ok neticesi olan durumlar ii de tercih edilebilir. rnek olarak,  $Y$ 'nin ailelerin aylık gelir miktarlarını temsil eden bir deđiŒken olduđunu varsayalım.  $Y$ 'de bulunan ailelerin ise lkemizin herhangi bir blgesinde yer aldığını varsayalım. Blgesel aıdan farklılığın mevcut olduđu bir olayda da kukla deđiŒkenleri kullanılabilir. Varsaydığımız durumda lkemizde Ege, Karadeniz, Akdeniz, Marmara, Gney ve Dođu Anadolu ile İ Anadolu Œeklinde sıralayabildiğimiz yedi ayrı blge bulunduđundan birden ok kukla deđiŒken vasıtasıyla regresyon denklemi oluŒturulabilir. Kukla deđiŒkenlerden birincisini  $K_1$  ile ifade edelim.  $K_1$  bahsedilen ailenin Marmara Blgesinden olma durumunu tespit etsin. Bahsedilen durumda  $K_1$  ile ifade edilen aile Marmara Blgesinde hayatını srdryor ise  $1$ 'e eŒit olur. Eđer ki bu blgede yaŒamıyorsa 0'a eŒit olur.  $K_1$  'in 0'a eŒit olması bahsedilen bu ailenin kalan altı blgeden herhangi birinde bulunduđunu ifade eder. Kukla deđiŒkenlerinden ikincisini ise  $K_2$  ile belirtelim.  $K_2$  'nin ise ailenin Ege blgesinde hayatını srdrp srdrmediđini tespit eden baŒka bir kukla deđiŒken olduđunu varsayalım. Bahsedilen durumda eđer ki aile Ege Blgesinde bulunuyorsa  $K_2$   $1$ 'e eŒit olur. Aile Ege'de yaŒamıyorsa  $K_2$  0'a eŒit olur. Btn blgeler iin burada verilen rneklere benzer olacak biimde kukla deđiŒkenler altta verilen Œekilde tanımlanır.

$K_1$ : Marmara Blgesi

K<sub>2</sub>: Ege Bölgesi

K<sub>3</sub>: Karadeniz Bölgesi

K<sub>4</sub>: Akdeniz Bölgesi

K<sub>5</sub>: İç Anadolu Bölgesi

K<sub>6</sub>: Doğu Anadolu Bölgesi

K<sub>7</sub>: Güneydoğu Anadolu Bölgesi

Şu an meydana getireceğimiz modelde Y ailelere ait aylık gelir miktarlarını ifade eden değişken, K ile ifade edilen değişkenler ise kukla değişken olacak biçimde bahsedilen ailelerin ülkemizin hangi bölgesinde yaşamlarını sürdürüp sürdürmediklerini gösterebilir.

$$Y = \beta_0 + \beta_1 K_1 + \beta_2 K_2 + \beta_3 K_3 + \dots + \beta_6 K_6 + \beta_7 K_7 \text{ formül 26}$$

Üst tarafta meydana getirilen eşitlik matematiksel açıdan doğruymuş gibi görünüyorsa da kukla değişkenlerden bir tanesi bu denklemde fazladır. Örnek olarak, K<sub>1</sub>, K<sub>2</sub>, ..., K<sub>6</sub> şeklinde ifade edilen değişkenlerinin tamamının 0'a eşit olduğu bir durumu hayal edelim. Yan, K<sub>1</sub> = K<sub>2</sub> = ... = K<sub>6</sub> = 0 şeklinde olacak. Burada ifade edilen varsayım bu ailenin bahsettiğimiz bölgelerin altısında da yaşamadığı fikri ile birebir eşittir. Ülkemizde toplamda 7 tane bölge bulunduğu için, bu fikir ile bahsettiğimiz ailenin geriye kalan bölgede bulundu anlaşılır. Burada geriye kalan bölge en başta K<sub>7</sub> kukla değişkeni ile temsil edilen bölgedir. Böyle bir durum söz konusu olduğunda denklem içerisinde K<sub>7</sub> değerinin bulunmasına gerek yoktur. Bu problem ise kukla değişken tuzağı şeklinde isimlendirilir. Verilen bu bilgiler ışığında doğru bir regresyon denklemi altı adet kukla değişkenden meydana gelir.

$$Y = \beta_0 + \beta_1 K_1 + \beta_2 K_2 + \beta_3 K_3 + \dots + \beta_6 K_6 \text{ formül 27}$$

Denklem içerisinde çıkarılacak olan kukla değişkenin hangisi olduğu neticeyi ya da durumu etkilemez.

Özet olarak, bir taneden daha çok parametrenin bulunduğu ve yalnızca bir adet kukla değişkenin kullanımının tercih edilmediği durumlarda diğer bir ifade ile parametrelerden her biri için bir adet kukla değişkenin tercih edildiği hallerde doğrusal bağıllık sorunu ortaya çıkmaktadır. Bağıllık probleminin meydana



geldiği hallerde bahsedilen parametrelerde öngörüle bulunmak imkânsızdır. Sözü edilen durum “kukla değişken tuzağı” şeklinde isimlendirilmektedir. Kukla değişken tuzağına kapılmamak için bahsedilen sabit parametrelili model içerisinde kukla değişken tarafından temsil edilen durum ve karakter değeri sayısından bir birim eksik miktarda kukla değişken bulunmaktadır. Bağımsız kukla değişkenin kullanıldığı yerler aşağıda maddeler halinde verilmiştir.

- Yalnızca kukla değişkeni bulunan modeller (Örneğin; Varyans analiz modelleri),

$$Y_i = a_0 + a_1 K_1 + a_2 K_2 + \dots + a_n K_n + u_i \text{ formül 28}$$

- Eş zamanlı olarak hem sayısal değişkenin hem e kukla değişkeninin kullanıldığı modeller (Kovaryans analizi modeller)

$$Y_i = a_0 + a_1 K_1 + a_2 K_2 + \dots + a_n K_n + b_1 X_1 + b_2 X_2 + \dots + b_n X_n + u_i \text{ formül}$$

29

- Mevcut olan kukla değişkenlerin birbirleri üzerinde karşılıklı etkisinin olduğu modeller
- Mevsimsel yaşanan değişimlerin hesaplanması sırasında kullanmakta olan kukla değişkenler
- Parçalı Doğrusal Regresyon analizi kapsamında tercih edilen kukla değişkenler

## 2. Sınıflama

Veri madenciliğinde en popüler iş sınıflamadır. Girilen verilerin farklı özelliklerine uygun şekilde bir sınıflayıcı (model) vasıtasıyla gruplara ayrılmasını kapsayan işlemdir. Mevcut bulunan cisimlerin bir gruba verilip verilemeyeceğini veya grupların hangisine verileceğinin tespit edilmesi işidir. Diğer bir deyişle, cisimler ya da durumlar için uygun olan grubun ön görülmesidir. Sınıflama girdileri, her biri bir sınıfa ait etiketler ile işaretlenecek örneklerden ya da gözlemden meydana gelen bir eğitim sınıfıdır. Model tarafından gözlemlerden her biri için özellikleri baz alarak gerçekleştirilen atanan sınıflara ait etiketler ise çıktı şeklinde isimlendirilir. Makine öğrenmesine ait en mühim araçlardan biri de sınıflamadır. Eğitime ait örneklerden meydana gelen bir eğitim grubu ve test numunelerinden meydana gelen bir test grubu tüme varımsal şekilde

gerçekleştirilen öğrenmenin esas hatlarını meydana getirir. Yapılan sınıflama iki ayrı basamaktan oluşmaktadır. Bu adımlardan ilki verilerin eğitimi ve ikincisi ise modelin testidir. Eğitim grubundan çıkarım yapılarak modelin hazırlanması süreci eğitim adımını kapsar. Test adımında ise test kümesi aracılığıyla modelin kesin olup olmama durumu test edilir. Modellere ait keskinlik seviyesinin tespit edilmesi için test numunelerinin iyi olarak tanınan sınıfı, model vasıtası ile ön görülen sınıf ile kıyaslar. Test numunelerinin model aracılığıyla doğru biçimde sınıflanma yüzdesi kesinlik oranını bulmayı sağlar. Girdi verilerini kullanarak söz konusu çıktıları oluşturan model, ardından sınıf etiketi kaybolmuş olan ya da bilinmeyen yeni gözlem ya da numunelerin sınıf etiketini bulmak amacıyla da tercih edilebilir. Bayesgil sınıflayıcıları, karar ağaçları, diskriminant analizi, kaba kümeler, duruma dayalı uslamlama, bulanık kümeler, lojistik regresyon, eğer-sonra kuralları (kural çıkarımı), öteki mantığa dayalı formüller, yapay sinir ağları şeklinde temel sınıflama metotları olarak listelenebilir.<sup>7</sup>

Literatüre bakıldığında sınıflama konusuna dair oldukça farklı bir yaklaşıma rastlamak mümkündür. Sınıflama, söz konusu olan bu yaklaşıma göre, kısmi ve tam sınıflama şeklinde iki alt kategoriye ayrılmaktadır. Tam sınıflama veri içerisindeki bütün gruplar ve örnekleri içerisine alan modellerle alakalıdır. Örneğin; C4.5, CHAID, yapay sinir ağları, C&RT, CHAID, C5.0 ve bununla beraber öteki karar ağaçları şeklindeki akıllı yöntemlerle diskriminant analizi olarak isimlendirilen istatistiksel teknikler şeklinde sıralanabilir. Kısmi sınıflama söz konusu olduğunda ise, tıpkı tam sınıflamadan gibi veri kümelerine dair nitelikler ortaya konmaktadır. Fakat kısmi sınıflama modellerine bakıldığında bütün sınıflar ya da verilen sınıfın bütün örneklerini kapsayamadığı görülebilir. Birliktelik kuralları bahsi geçen modellerden biri olarak gösterilebilir.<sup>8</sup>

---

<sup>7</sup> Emel, Gül GÖKAY; Çağatan TAŞKIN (2005). Araç Rotalama Problemlerinin İki Aşamalı Çözümünde Genetik Algoritma Kullanımı. *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 7.1: 1-17.

<sup>8</sup> Bloemer, M. M. J., Brijs, T., Vanhoof, K. ve Swinnen, G. (2003); "Comparing Complete And Partial Classification For Identifying Customers At Risk", *International Journal Of Research In Marketing*, Vol:20, Issue:2, s. 117-131.

## B. Denetimsiz Öğrenme

Denetimsiz öğrenme, etikete sahip olan veriler kullanılmadan, girdi veri grupları üzerinden çıkarımlarda bulunmak için tercih edilen bir çeşit makine öğrenme tekniğidir. Bu öğrenme tekniği, gizli değişken modeller öğrenilmesi, kümeleme, sinir ağları, aykırılık tespiti şeklindeki farklı yaklaşımları kapsamaktadır. Ancak bunların arasında yer alan ve en yaygın olarak kullanılan denetimsiz öğrenme tekniği, gizli modelleri tespit edebilmek ya da verileri kümelemek amacıyla kullanılmakta olan kümeleme analizi tekniğidir.

### 1. Kümeleme Analizi

Bu analiz çeşidi, birbiri ile benzerlik gösteren veri nesnelere tamamını aynı kümenin içerisinde yer alacak biçimde gruplarken; farklı özelliklere sahip olan veri nesnelere ise birbirinden ayrı biçimde kümeleyerek verileri gruplandırır. Meydana gelen veri bölümü, verilerin daha iyi algılanmasını sağlar ve içeriğini ortaya koyar. Kümeleme algoritmaları tarafından, bir veri grubu sınıflara ayrılır ya da gruplandırılır. Bu işlem esnasında benzerlik gösteren veri nesnelere aynı gruba atanmalıdır. Birbiri ile farklı olan veri nesnelere ise ayrı kümeler verilmelidir. Bir başka ifade ile, kümeleme analizi, şirketler, devletler, faaliyetler, insanlar, markalar ve buna benzer şeyler gibi gözlemlenebilen verileri anlamlı kümeler, taksonomiler ya da gruplar biçiminde düzenlemek için keşif amacıyla kullanılmakta olan bir veri analiz yöntemidir; en başta bilinmeyen gruplar ya da kümeler içerisindeki farklılıklar. Burada benzerlik kavramı matematiksel alandaki benzerlik kavramı ile aynı anlamda algılanmalıdır.

İlk kümeleme analizi kavramını ortaya atan kişi Robert C. Tryon'dur. Geride kalan son birkaç senede, kümeleme analizi biliminin biyolojik, fiziksel, finansal ve ekonomik alanlarda oldukça kıymetli bir teknik olduğu anlaşılmıştır.

Kümeleme tekniğinde sonuçların bağlı olduğu şey farklılıkların (benzerlik) seçimidir. Dolayısıyla esas soru, örnekler arasında mevcut olan farklılığın (benzerliği) ne şekilde hesaplanması gerektiğidir. Numuneler arasında genel bir farklılık fonksiyonu seçimi Öklid uzaklığıdır. Çoğunlukla metrik uzaylarda, benzerlik bir mesafe normu aracılığı ile açıklanır. Sıklıkla mesafe normu ya da benzerlik daha öncesinden bilinmemektedir. Altta verilen nitelikleri sağlayan iki

boyutlu fonksiyon şeklinde kabul edilen x ve y (veri olarak) arasında yer alan uzaklıktır.

- x için;  $d(x, x) = 0$  formül 30
- x ve y için;  $d(x, y) \geq 0$  formül 31
- (x, y) için;  $d(x, y) = d(y, x)$  formül 32
- x, y ve z için;  $d(x, y) + d(y, z) \geq d(x, z)$  formül 33

Sürekli değişkenler halinde, oldukça uzun bir uzaklık fonksiyonları listesi bulunmaktadır (üst tarafta verilen nitelikleri karşılayan). Her bir uzaklık fonksiyonu, geometrileri dolayısıyla birbirinden farklı veri görünümünü tanımlar. Altta verilen çizelge, küme analizi işleminde çoğunlukla farklılığı hesaplayan tanımlar ile beraber birbirinden farklı uzaklık fonksiyonlarını ifade etmektedir.

Çizelge 3. Uzaklık Ölçüleri

Uzaklık Ölçüleri	Formül
Minkowski	$d(x, y) = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p}$
Hamming	$d(x, y) = \sum_{i=1}^n  x_i - y_i $
Öklid	$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Angular Aralığı	$d(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\left[ \sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2 \right]^{1/2}}$
Tchebyshev	$d(x, y) = \max_{i=1,2,\dots,n}  x_i - y_i $

SPSS, dataları gruplandırmak amacıyla kullanılabilecek birbirinden farklı üç prosedür bulundurmaktadır. Bunlar, iki adımlı kümeler, hiyerarşik küme analizi ve k-ortalımalı kümeler şeklinde sıralanabilir. Hiyerarşik kümeleme tekniği bütün durum çiftleri içerisinde bir uzaklık matrisine ihtiyaç duyar. K-ortalımalı kümesi, grupların içerisinde ve dışarısında durumların birbirine

kariřtirilmesini ve ilerleme sürecinde yer alan küme miktarının bilinmesine ihtiyaç duyar. Oldukça büyük bir veri grubunda, sıralı ya da devamlı veriler temelinde çabuk biçimde kümeler hazırlanmak için bir kümeleme yöntemi kullanılacaksa o zaman bu söz edilen iki yöntem de tercih edilmez.

## 2. Faktör Analizi

2009'da Hair ve arkadaşları tarafından ortaya atılan fikre göre; faktör analizinin gerçekleştirilmesindeki en temel amaç ortaya konan arařtırmadan elde edilen dataların yapı geçerliliğini denemesidir. Bunun yanı sıra çalışmadaki deęişkenler tarafından yaratılan yapının tanımlanmasına da fayda sağlar (Hair vd. 2009). Literatür içerisinde gerçekleştirilmiş çalışmalara bakıldığında faktör analizi kalitesinin incelenmesi hususunda birtakım ölçülerden yararlanılmaktadır. Bahsi geçen bu ölçülerden en az faktör yükü örnek grubu büyüklüğü için Kaiser-Meyer-Olkin (KMO) örneklem uygunluk ölçüsü aracılığı ile bulunurken, Barlett küresellik testi ve örneklem uygunluk ölçüsü şeklindeki analiz çeşitlerinden de faydalanılmaktadır.<sup>9</sup> Faktör analizinin “kabul edilebilir olması için bir faktörün minimum üç ayrı ifadeye sahip olması zorunludur.”<sup>10</sup>

## C. Takviyeli Öğrenme

Takviyeli öğrenme makine öğrenmesinin bir türüdür (Reinforcement Learning). Bu öğrenme çeşidi, yer aldığı çevreyi anlayan ve çevresinde hareket gerçekleştiren bağımsız bir ajanın, amacına erişebilmesinde doğru kararlar vermeyi ne şekilde öğrenebileceğini ortaya koyar.<sup>11</sup> Ajan, o alanda herhangi bir hareket gerçekleştirdiğinde, yeni durumun en uygun hale gelmesi için bir ceza ya da ödül değeri aracılığıyla ilave edilir. Ajan, hedefe varmak için ortaya koyabileceği en doğru davranışı bulmaya çalışır. Ajanın hedefi, en geniş kapsamlı toplam ödülü oluşturan hareketlere ait sırayı bulmaktır.

---

<sup>9</sup> Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E. (2009), *Multivariate Data Analysis*, 7th Ed., PrenticeHall, London, UK

<sup>10</sup> Tabachnick, B.G. & Fidell, L.S. (2007). *Using Multivariate Statistics* (5th ed.). Pearson Education, Inc. / Allyn and Bacon.

<sup>11</sup> Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237-285.

## **D. Yarı Denetimli Öğrenme**

Denetimli ve denetimsiz öğrenme sistemlerin bir karışımı ise yarı denetimli öğrenme şeklinde isimlendirilir. Verilerden kimisi etiketlenirken, verilerin kimisi ise etiketlenmemiştir. Burada esas amaç etiketlenmemiş olan daha küçük çaplı veri grubunun etiketlenmiş bulunan veriler ile birlikte kullanılması vasıtasıyla öğrenme sürecinde oldukça ciddi oranda iyileşme meydana gelebildiği görülmüştür.<sup>12</sup>

## **E. İşbirlikçi filtreler (Collaborative Filtering) metodu**

Bireye ait öneride bulunmak amacıyla tüketicilerin gerçekleştirdiği geri dönüşlerden yararlanan metot işbirlikçi filtreler metodu olarak isimlendirilir. Bugün pek çok öneri sistemi oldukça başarılı olması dolayısıyla bu filtre metodundan faydalanmaktadır.<sup>13</sup> Kullanıcı olan bireylerin daha eski tavır ve davranışlarından ulaşılan bilgilerden faydalanılan bir tekniktir. Tavsiye ettiği mallar, tüketici ile benzer tavır ve tutumlar sergileyen kullanıcıların dikatini çeken ancak tüketicilerin henüz ilgili olmadığı mallardan meydana gelir.

## **F. İçerik bazlı (Content Based) öneri metodu**

İçerik bazlı öneri metodunda ana yaklaşım kullanıcı olan bir kişiye daha öncesinde beğenmiş olduğu ürüne benzeyen bir malın tavsiye edilmesidir.<sup>14</sup> Oldukça benzer olan bir ürün aranırken diğer ürünlere ait özelliklerden yararlanılır. Ürün nitelikleri herhangi bir filmin yönetmeni, gösterim senesi, aktörü ve türü şeklinde olabilir. Nitekim, pek çok defa aynı yönetmen tarafından yönetilen filmleri beğenen bir kişiye aynı yönetmene ait yeni vizyona girmiş bir film tavsiye edilebilir.

---

<sup>12</sup> Sarıcaoğlu, C. (2019). Sözcüksel Analiz Kullanarak Kötü Niyetli Url'leri Derin Öğrenme Teknikleri İle Tespit Etme. Gazi Üniversitesi Yüksek Lisans Tezi.

<sup>13</sup> Karabulut, E. M. (2016) Investigation Of Deep Learning Approaches For Biomedical Data Classification. Doktora Tezi. Çukurova Üniversitesi. Adana

<sup>14</sup> Leskovec, Rajaraman ve Ullman, Stanford University, <https://www.youtube.com/watch?v=2uxXPzm-7FY>.

## G. İlgili Çalışmalar

Derin öğrenme uygulamalarının pek çok alanda, başka makine öğrenmesi algoritmalarına kıyasla çok daha başarılı neticeler vermesiyle, araştırmacı kişiler aşağıda verilmiş olan çalışmaları gerçekleştirmişlerdir.

- 2015'te Suvash Sedhain Otomatik Kodlayıcıları Netflix ve MovieLens veri kümesine uygulamıştır. Otomatik Kodlayıcıların RBM ve Matrisi çarpanlarına ayırma tekniklerine kıyasla çok daha iyi bir başarıma ulaştığı sonuçlarla belirtilmiştir.
- 2017'de Yu Zhu bir RNN türü olan LSTM'lere zaman dilimleri ekleyerek üç ayrı model tavsiye etmiş bu tekniğin tavsiye performansını yükselttiğinin neticelerini iki ayrı veri setinde ortaya koymuştur.
- 2007'de Ruslan Salakhutdinov tarafından ürün tavsiye sistemleri içerisinde Netflix şirketi tarafından sağlanan veri kümesi üzerinde yapay sinir ağı (YSA) türü olan RBM'leri (Restricted Boltzman Machine) uygulanmış ve Netflix'in şahsi tavsiye sistemi içerisinde %6 daha iyi netice verdiğini gözler önüne sermiştir.
- 2015'de Florian Strub SDA'ları (Stacked Denoising AutoEncoder) MovieLens 1M ve Jester data setinde test etmiş ve dört katmana sahip bir Otomatik Kodlayıcının iki katmanlı bir Otomatik Kodlayıcıyla kıyaslandığında çok daha başarılı sonuçlar elde edilebileceği görülmüştür.
- 2017'de Bahriye Akay MovieLens veri setinde DBN'leri (Deep Belief Network) değişik optimizasyon algoritmaları aracılığıyla kullanmıştır. Sonuçta Adamax algoritmasının ezberleme sorunu yaşadığını göstermiştir.
- 2017'de Daniel Sanchez tarafından yayını yapılan yüksek lisans tezi içerisinde görüntüleme, ürün satın alma gibi kullanıcı davranışlarının sıralı bir veri seti haline dönüştürülebileceğinden söz etmiştir. Veri setleri MovieLens ve Santander tarafından RNN'lerde kullanılarak, kullanıcıların bir sonraki adımda ortaya koyacakları tavır ve tutumlar öngörülme çalışılmıştır.

### III. DERİN (PEKİŞTİRMELİ) ÖĞRENME

Bolat ve Kızrak (2018) tarafından gereksinimler ve derin öğrenme “Derin öğrenmenin temelleri 1980 ve bu yılı takip eden süreçte atılmıştır. Derin öğrenme, doksanlı dönemde donanım dolayısıyla meydana gelen engeller sebebi ile bir çeşit duraklama devri içerisine girmiş olsa da bugün yapay zekânın en bilinen alt dalı olma niteliğine sahiptir. Bilgisayarlı görü her geçen gün gelişim göstermektedir. Bilgisayarlı görü, savunma sanayii, doğal dil işleme, sanat, finans, otonom araçlar, müzik ve güvenlik şeklindeki pek çok çalışma çevresinde faaliyetleri olan derin öğrenme alanında çalışma gerçekleştirirken yüksek hızlı grafik işlem birimlerine (GPU), yüksek hacimli bellekler ile veya bulut çalışma alanlarında gereksinim duyulmaktadır.” biçiminde ifade edilmektedir.

Donanım alanında meydana gelen gelişmelerin sonrasında derin öğrenme kavramının 1990 ve takip eden senelerde içerisine girdiği duraksama dönemi biterek gelişimini devam ettirmeyi sürdürmektedir. Büyük çaplı veri kümelerini işleme yeteneğine sahip olan algoritmalar derin öğrenme olarak isimlendirilir. Veri bir davranışın başladığı andan bittiği ana kadarki süre içerisinde imal edilen ve bu davranışın sona erdirilmesinden sonra imalatı süren deneyimin kaydedilebilen versiyonudur. Deneyimin güvenli kayıtlarını meydana getiren verilerin, bahsedilen deneyimi deneyimlemeyen veya tecrübe edemeyenler aracılığıyla karşılanıp, işlenebilir duruma getirilmesi, değişik amaçlar için kullanılarak, daha yeni verilerin elde edilmesi de oldukça önemlidir. Veri kümelerinin incelenmesi ve elde edilen bu yeni neticelerle tekrardan veri kümeleri meydana getirilme fırsatı mevcuttur. Veri işleme neticesi bulunan veri kaynağının ve gerçekleştirilen işlemlere bağımlı biçimde meydana getirilen yeni veri kümelerinin pek çok konuda ilerlemelere destek olabilecek yetenekte olduğunu ifade edebilir. Algoritmalara ait sorunları çözmek için başarı düzeylerini de yükseltecekleri tahmin edilmektedir. Makine öğrenme fonksiyonları içerisindeki en temel birimlerinden biri olarak veri gösterilmektedir. Verilerin ne kadar erişilebilir olduğu makine öğrenmesinin



ilerleyişini etkiler. Verinin ulaşılabilir nitelikte olması, veri manipülasyonu işleminin bugün basitçe uygulanır duruma gelmesi vasıtası ile derin öğrenme de yaygın hale gelmiştir. Veri işleme sürecinde ve derin öğrenme konusunda yüz yüze gelinebilecek önemli tehlikelerden bir tanesi ulaşılan ilk verilerin doğruluk ve güvenlik düzeyinin yeterli seviyede bulunmamasıdır. Bahsedilen tehlikenin ortadan kaldırılması için verinin doğru olup olmadığı kontrol edilmeli, çözüme ulaştırılacak sorun için kullanılmasının uygunluğu tespit edilmelidir. Bu sayede makine öğrenmesi işleminde kullanılacak derin öğrenme sürecinde meydana gelecek yenilikler algoritmaya ait verim düzeyini de yükseltmektedir.<sup>15</sup>

Derin öğrenmenin özelinde altı mimari yapı olduğu gözlemlenmektedir. İlk olarak, “Konvolüsyonel Sinir Ağları” mimarisine değinmek mümkündür. Bu mimaride bilgisayarlı görme görevlerinde etkili bir mimari olarak yer almaktadır. Ayrıca bu çalışmanın modellenmesinde ayrıntılı kullanılmaktadır. Derin öğrenme kavramına ait temel mimarilerden biri de CNN yapısıdır. Yann Le Cun (1988) tarafından geliştirilmiştir. 1998 yılına kadar sürekli gelişim göstermiştir. Le Net ilk CNN mimarisine örnek olarak verilmektedir.<sup>16</sup> CNN mimarisi görüntüleri girdi olarak almak için tasarlanmıştır. konvolüsyonel katman ve standart çok katmanlı bir sinir ağı gibi bir veya daha fazla tamamen bağlı katmanlardan meydana gelmektedir.<sup>17</sup> Bir CNN’yi oluşturan katmanlar; “*Giriş Katmanı, Konvolüsyon Katmanı, Düzleştirilmiş Doğrusal Birim Katmanı, Havuzlama Katmanı, Tam Bağlantılı Katman, DropOut Katmanı ve Sınıflandırma Katmanı*”dır.<sup>18</sup>

Diğer derin öğrenme mimari <sup>19</sup>yapılardan biri olan “Tekrarlayan Sinir Ağları”; birimler arası verilerin yönlendirilmiş bir döngü oluşturduğu yapay sinir ağı olarak tanımlanmaktadır.

---

<sup>15</sup> Özgür, S. B. Algoritmalar, Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme ve Uygulamaları: Beşeri Fayda Üretiminin Yazılımlar Tarafından Karşılanması. Ekonomi ve Yönetim Araştırmaları Dergisi, 10(1), 1-29.

<sup>16</sup> Y. Le Cun et al.; “Handwritten digit recognition: applications of neural network chips and automatic learning,” IEEE Commun. Mag., Vol. 27, no. 11, pp. 41–46, Nov. 1989

<sup>17</sup> Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, Vol. 521, pp. 436–444, 2015.

<sup>18</sup> Gündüz, G., & Cedimoğlu, İ. H. (2019). Derin öğrenme algoritmalarını kullanarak görüntüden cinsiyet tahmini. Sakarya University Journal of Computer and Information Sciences, 2(1), 9-17.

<sup>19</sup> Abdulkadir ŞEKER, “Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme” Doktora Semineri, YTÜ Bilgisayar Müh. Bölümü, İstanbul, 2017.

“Uzun-Kısa Vadeli Hafıza Ağları”; değerleri rastgele aralıklarla hatırlayan tekrarlayan sinir ağı mimarisi olarak tanımlanmaktadır.

“Sınırlı Boltzman Makineleri”; girdi seti üzerinde olasılık dağılımlarını öğrenebilen görünür katman ve gizli katman olmak üzere iki katmandan oluşan bir yapay sinir ağı yapısı olarak tanımlanmaktadır.

“Derin İnanç Ağları” yapısı; Sınırlı Boltzman Makineleri’nin yığını ve oto kodlayıcılarla bileşimi olarak tanımlanmaktadır.

“Derin Oto Kodlayıcılar” mimari yapısı ise; denetimsiz öğrenme için ve verinin üretken modellerini öğrenmek için kullanılan, girdi katmanı verilerini, çıktı katmanına kopyalayan bir yapı olarak tanımlanmaktadır.

Makine öğrenmesi ve derin öğrenme için farklı kurumlar, üniversiteler ve şirketler tarafından geliştirilmiş olan, farklı özellikler barındıran ve derin öğrenme çalışmalarını pratikleştirip kolaylaştıran çok sayıda hazır kütüphane ve API (Application Programming Interface-Uygulama Programlama Arayüzü) mevcuttur. Derin öğrenme sisteminin kendi yapısı içerisinde birden fazla kütüphane mevcuttur. Kullanıcının çalışılacağı konu özelinde, en uygun kütüphanelerin bilgisayara kurması gereklidir. Derin öğrenmede kütüphaneler, yapılacak işin basitleşmesine olanak sağlamaktadırlar. Bu özellik ise derin öğrenme için büyük öneme sahiptir. Bu bağlamda değerlendirdiğimizde derin öğrenme yapısında her kütüphanenin her biri farklı işleve sahiptir. Python programlama dilinde, bir düzineden fazla derin öğrenme kütüphanesi olduğu görülmektedir. “Theano, TensorFlow, Caffe, Keras, Mxnet, Pylearn2, Lasagne, Torch, Covnetjs, Sci-Kit Learn, Deeplearning4j, Apache Spark” yoğun olarak tercih edilen kütüphanelerden bazılarıdır.<sup>20</sup>

## **A. Yapay Sinir Ağları (YSA)**

İsminden de anlaşıldığı üzere, yapay sinir ağları, beynin oldukça basit bir nöron formunun benzetimidir. Bu sayede oluşturulan ağ vasıtası ile öğrenme meydana gelir. Yapay sinir ağları tahmini arıza analizi, haberleşme, kontrol ve

---

<sup>20</sup> Gündüz, G., & Cedimoğlu, İ. H. (2019). Derin öğrenme algoritmalarını kullanarak görüntüden cinsiyet tahmini. *Sakarya University Journal of Computer and Information Sciences*, 2(1), 9-17.

sistem tanımlama, görüntü ve ses tanıma, trafik, tahmin ve kestirim analizi, imalat yönetimi şeklinde sıralanabilecek oldukça fazla sayıda alanda kullanılabilir. Yapay sinir ağları, birey beyninin nitelikleri içerisinde bulunan öğrenme aracılığı ile yeni bilgiler oluşturabilme ve keşfedebilme şeklindeki yetileri, yardımsız kendi kendine meydana getirmek amacıyla biyolojik sistemde yer alan sinir ağlarından ilham alınarak tasarlanan bilgisayar sistemleri olarak ifade edilebilir.<sup>21 22</sup> Yapay sinir ağlarının herhangi bir olaydaki girdi ve çıktılar içerisindeki ilişkiyi, doğrusal olan ya da olmayan, mevcut olan örneklerden öğrenerek, öncesinde hiç rastlanmamış olan olayları, daha önceki örneklerden benzetim yaparak mevcut olaya dair çözümler hazırlayabilmesi, algoritmik çözümü keşfedilememiş sorunların çözülmesi konusunda kullanımını daha yaygın hale gelmiştir. Yapay sinir ağları içerisindeki en ciddi sorun kompleks problemlere çözüm bulmak için çok katlı ve çok sayıda nöron barındıran yapay sinir ağlarına gerek duyulmasıdır.<sup>23</sup>

Yapay Sinir Ağları yapay sinir hücrelerinin birleşmesi ile meydana gelir. Sinir hücreleri rastgele biçimde birleşmez. Çoğunlukla hücreler 3 tabaka şeklinde ve tabakalardan her biri içerisinde paralel biçimde birleşerek ağı meydana getirirler. Hücrelerin meydana geldiği bu katmanlar şu şekilde listelenebilir:<sup>24</sup>

- Girdi katmanı: Bu katman içerisinde yer alan işlem elemanları dışarıdaki dünyada yer alan bilgileri içeriye alır. Daha sonra bu bilgileri ara tabakalara aktarmakla görevlidir. Kimi ağlarda girdi tabakasında herhangi bir bilginin işlenmesi gerçekleşmez.
- Ara katmanlar: Girdi katmanını aşarak gelen bilgiler ara katmanda işlenir ve buradan da çıktı katmanına transfer edilir. Söz konusu bilgiler ara katmana geldiğinde işlenir. Mevcut bir ağ için bir taneden daha fazla sayıda ara katman bulunabilir.
- Çıktı katmanı: Çıktı katmanında yer alan işlem elemanları ara tabakadan gelmiş olan bilgiler üzerinde işleme yaparak, ağın girdi tabakasından

---

<sup>21</sup> Fausett, L. (1994), "Fundamentals of Neural Networks: Architectures, Algorithms and Applications", Prentice Hall.

<sup>22</sup> Öztemel, Ercan (2003), "Yapay Sinir Ağları, Papatya Yayıncılık", İstanbul.

<sup>23</sup> KOHONEN, T. (1987). "Int. Conf. on AI", State of the Art in Neural Computing, pp. 1-79, 1-90

<sup>24</sup> Öztemel, a.g.e.31-33

gelen örnek diđer bir ifade ile girdi seti için hazırlaması gerekli olan çıktıyı hazırlarlar. Meydana getirilen bu çıktı dış dünyaya yollar.

Genel olarak yapay sinir ađları modellerini ;

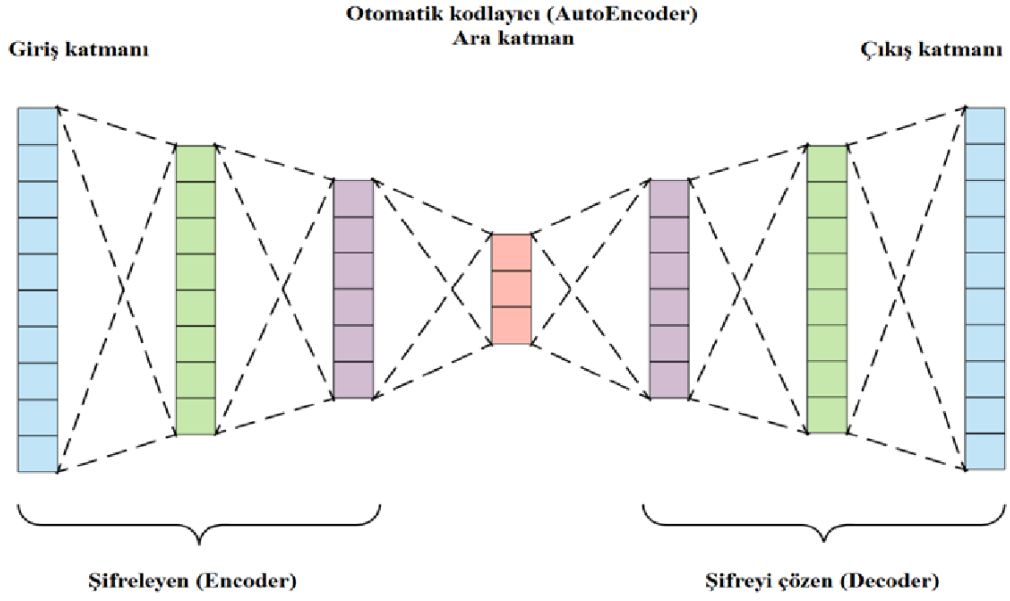
- a) Ađın yapısına,
  - b) İleri beslemeli ( feed forward ),
  - c) Geri beslemeli ( feed back ),
  - d) Ađrılık matrislerinin simetrik veya asimetrik oluđuna
  - e) Ađrılık matrisi deđerlerinin sabit veya deđişken oluşuna,
  - f) Ađda yer alan düđümlerin özelliklerine,
  - g) Kullanılan eşik fonksiyonuna,
  - h) Düđüme sadece analog / ikil veya sürekli deđerlerin uygulanabilmesine,
  - i) Eğitim veya öğrenme kurallarına
- bađlı olarak görölmektedir.<sup>25</sup>

## **B. Otomatik kodlayıcılar**

Otomatik Kodlayıcılar (AutoEncoder) bir YSA türüdür. Otomatik Kodlayıcılar sıklıkla sunulan giriş deđerleri ile çıkışta varılan deđerlerin kendi içinde yakın olmasını talep eden YSA'lar olarak bilinir. Bazen bir bazen de birden fazla gizli katman içerirler. Verilen şekilde de görüldüğü gibi YSA ilk önce giriş verisini şifreleyen (Encoder), ardından kendisi tarafından şifrelenen veriyi çözümleyen (Decoder) katmanlardan meydana gelmektedir.

---

<sup>25</sup> Şen, Z. (2004). Yapay Sinir Ađları İlkeleri, Su Vakfı Yayınları, İstanbul.



Şekil 3. Bir YSA türü olan Otomatik Kodlayıcılar

Burada verilen YSA'lar içerisinde sıklıkla maliyet fonksiyonu şeklinde Ortalama Karesel Hatanın Karekoku-Root Mean Squared Error (RMSE) tercih edilir. RMSE'de mevcut olması gerekli olan değerden öngörülen değer eksiltilir ve karesi alınır. Ardından tüm sonuçların toplamından bir karekök hesabı yapılarak, total örnek adedine bölünür ve hataya ulaşılır. Çoklu katmanlardan meydana gelen Otomatik Kodlayıcı, eğitilmesi sürecinde geri yayılım algoritmasından (backpropagation) faydalanılır.

Otomatik Kodlayıcılar sıklıkla dataların boyutlarını düşürmede ve gereken oranda veri mevcut olmadığı durumlarda veri çoğaltma amacıyla tercih edilir.

Tavsiye sistemleri bakımından matrisi çarpanlarına ayırma tekniği ile kıyaslandığında, matrisi çarpanlarına ayırma tekniği lineer diğer bir ifade ile doğrusal bir algoritmadan faydalandığında doğrusal gizli nitelikleri (latent factor) algılayabilme yetisine sahiptir. Otomatik Kodlayıcılarda ise lineer olmayan

aktivasyon fonksiyonları tercihe edildiğinden lineer olmayan oldukça karmaşık gizli nitelikleri de algılayabilir.<sup>26</sup>

### **C. Tekrarlayan yapay sinir ağıları**

Sıralı verileri eğitmek amacıyla RNN'lerden faydalanılır. Çözümü oldukça güç pek çok sorunun çözülmesi açısından kolaylık sağlayan RNN'ler yazının sese ve sesin yazıya dönüştürülmesi, dil çevirisi, müzik ve yazı yazma, bir yazı içerisinde bir sonra gelecek sözcüğü tahmin etme gibi konularda faydalanılmaktadır.

Öneri sistemleri içerisinde ise örnek olarak, kullanıcı olan bireylerden bir tanesine ait satın alma geçmişini zaman bazlı bir data haline dönüştürülüp, bu data aracılığıyla RNN'ler eğitildiğinde kullanıcı olan kişinin bir daha sonra alma olasılığı olan ürün ön görülebilir. Başka bir örnek verecek olursak, kullanıcı olan kişilerin takip ettiği filmler zamana bazlı ve sıralı bir veriye çevrilirse, bu kişilerin daha sonra izlemek isteyeceği filmler de ön görülebilir.

Teze ait ikinci kısımda kullanılan tavsiye sistemi dizayn edilirken teknik olarak tercih edilen Otomatik Kodlayıcıların öneri sistemleri içerisinde ne şekilde kullanıldığından söz edilecektir.

TensorFlow platformu içerisinde tercih edilen algoritmalar ve ulaşılan grafikler sunulacaktır.

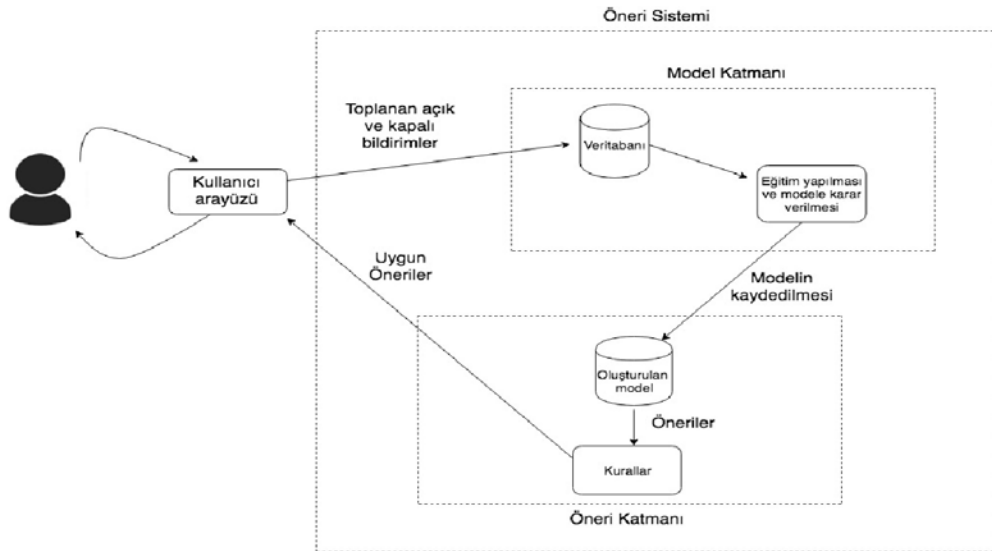
---

<sup>26</sup> Sedhain, S., Menon, A. K., Sanner, S., Xie, L. (2015, Mayıs). Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web (pp. 111-112). ACM

## IV. DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN ÖNERME SİSTEMİ ÜZERİNE BİR UYGULAMA

### A. Öneri sistemi modelinin eğitilmesi

Öneri sistemi içerisinde kullanıcı olan kişiler siteye giriş yaptığıında sistem faaliyete geçer. Kullanıcılardan alınan açık veya kapalı bildirimler sisteme ait eğitim katmanı içerisinde yer alan uygun bir veri tabanına depolanır ve kaydedilir. Bu veriler aracılığıyla makine öğrenmesi bazlı modeller eğitilir ve kullanılacak modele dair bir karar alınır ve ardından model kaydedilir. Burada kaydı gerçekleştirilen model sistemin gereksinimlerine göre zaman içerisinde yenilenebilir. Sistem tarafından siteye gelen kullanıcılara öneride bulunulmak istendiğinde öneri katmanına talep gerçekleştirilir. Yaratılan model uygun tavsiyeleri kural katmanından taşıyarak kullanıcı olan bireye uygun olacak tavsiyeler sunulur. Kural katmanı içerisinde ise sistem için tespit edilmiş olan kurallar yer almaktadır. Kullanıcılara tavsiye edilmemesi gerekli olan mallar veya kimi tarihlerde tavsiye edilmesi gereken ürünler örnek olarak gösterilebilir.



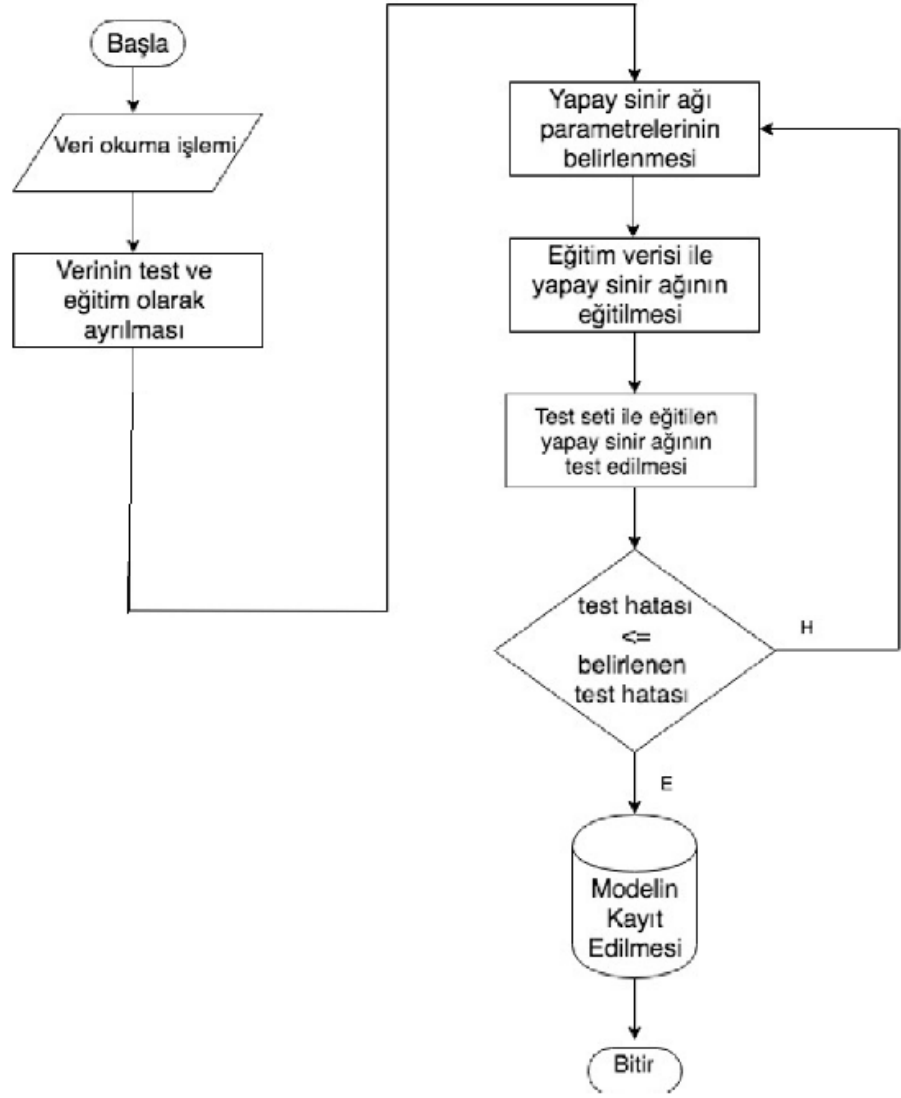
Şekil 4. Öneri Sistemi

Sırası ile öneri sistemine ait algoritması altta verilen aşamalardan meydana gelmektedir.

1. Kullanıcı olan kişilerden alınan açık veya kapalı bildirimler okunur.
2. Toplanan veriler test ve eğitim seti biçiminde ikiye ayrılır.
3. Amaçlanmış olan test hatası tespit edilir.
4. YSA'ya ait parametreler tespit edilir.
5. Tespit edilmiş olan parametreler kullanılarak YSA meydana getirilir
6. YSA eğitim verileri kullanılarak eğitilir.
7. YSA test verileri kullanılarak test edilir ve ardından test hatası hesaplanır.
8. Ulaşılan test hatası amaçlanmış olan test hatasından daha büyük bir değer ise  
4. Adıma yeniden başvurulur.
9. Ulaşılan test hatası amaçlanan test hatasından daha az ya da eşit çıkarsa model kaydedilir.

4. madde de tespit edilen YSA parametreleri YSA'nın kaç katmandan meydana geleceği, kaç devir altında eğitileceği yani tüm eğitim seti ile eğitime adedi de ve kullanılan aktivasyon fonksiyonu parametrelerinden meydana gelir.





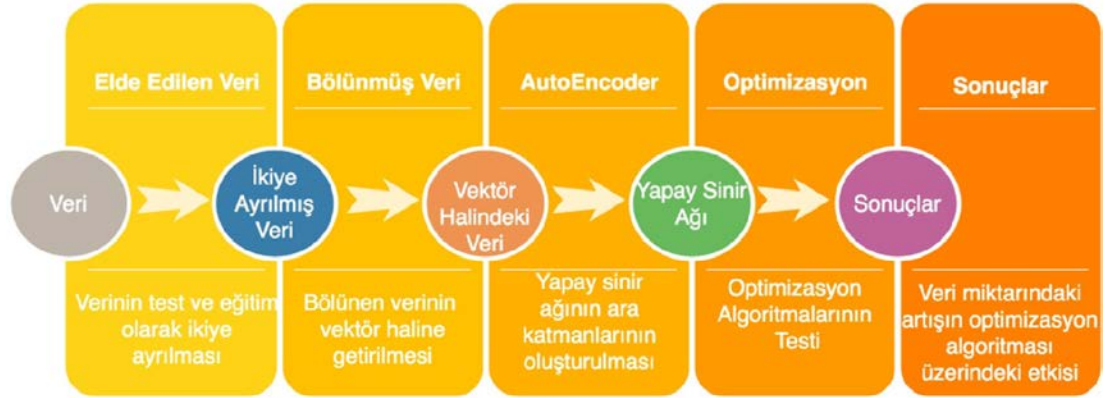
Şekil 5. Verilen şekilde algoritma akış şeması gösterilmektedir

Ortaya konan araştırma çalışmasına dair iş akış şeması basamakları altta maddelerle belirtilmiştir

internet sitesini kullanan kişilerden elde ettiği açık geri dönüşler tez çalışması kapsamında kullanılmıştır.

- Toplanan veri seti test ve eğitim amaçlı şekilde iki ayrı gruba ayrılmıştır. Eğitime dair veriler YSA'nın eğitimi sürecinde, test seti ise meydana getirilen YSA'yı test etmek amacıyla kullanılmıştır.
- Eğitim ve test seti kullanıcı olan kişilerden her biri için ayrı bir vektör halini almıştır.

- Otomatik Kodlayıcıların eğitilmesi amacıyla hazır duruma gelen veriler aracılığıyla YSA'nın kaç katmandan meydana geleceği, katmanlarda yer alan noron miktarı ve aktivasyon fonksiyonları tespit edilmiştir.
- Hazırlanan modelde 4 ayrı optimizasyon algoritması denenmiştir.
- Veri sayısındaki yükseliş ve düşüşün seçilmiş olan optimizasyon algoritmasına olan etkisi değerlendirilmiştir.



Şekil 6. İş akış şeması

## B. Bulgular

Yapılan çalışma da kullanılan veriler yazar tarafından oluşturulmuştur. Veri seti dört adet dosyadan oluşmaktadır.

README: Veri seti ile ilgili genel bilgilerden oluşan dosyadır.

ayakkabı.dat: model id, isim ve hangi sezona ait olduğu bilgilerinden oluşmaktadır.

Çizelge 4'de örnek olarak gösterilmiştir.

Çizelge 4. Ayakkabı veri seti

Id	İsim	Sezon
1	Spor	Yaz
80	Babet	Kış
150	Topuklu	Outlet
220	Bot	Bahar

users.dat: Kullanıcıların yaş, cinsiyet, meslek ve posta kodu bilgilerinden oluşmaktadır.

Çizelge 5’de örnek olarak gösterilmiştir. Meslek kolonunda bulunan rakamların hangi meslek grubuna ait olduğu README dosyasında yer almaktadır.

Çizelge 5. Kullanıcı veri seti

Id	Cinsiyet	Yaş	Meslek	Posta Kodu
45	F	50	10	72000
450	M	25	15	16050
4650	M	45	8	34030

ratings.dat: Bir kullanıcının hangi ayakkabıyı ne zaman kaç puan verdiği bilgisinden oluşmaktadır. Dosyada toplam 30.000 farklı ürün ve 4865 farklı kullanıcı bulunmaktadır. Kullanıcıların daha önceden verdiği siparişler ve bu siparişlere ait puanlamalar yer almaktadır.

Çizelge 6. Puan veri seti

Kullanıcı Id	Ürün	Puan
1	1250	5
2	775	2
2	110	3
5	2230	3
9	500	5

## 1. Veri setinin hazırlanması

Tez çalışmasında ratings.dat dosyasındaki veriler kullanılmıştır. Otomatik Kodlayıcılar herhangi bir zaman bilgisi beklemediğinden, veri setine zaman verisi eklenmemiş olup uygulamada kullanılmamıştır.

Verilerin ilk olarak test ve eğitim setine bölünmesi gerekmektedir. Toplam 30.000 farklı ürün ve 4865 farklı kullanıcı bulunmaktadır ve bu kullanıcılara ait 1 ile 5 arasındaki puanları vardır. Veri seti ikiye bölünürken her kullanıcı hem test setinde hem de eğitim setinde olacak şekilde bölünmüştür. Her kullanıcının verdiği toplam puan sayısının %70’ı eğitim setine %30’u ise test setine dahil edilmiştir. Veri seti bölünürken test verisini oluşturan Java dilinde yazılan kod parçacığı aşağıdaki gibidir.

```
JavaRDD<String> stringJavaRDD = sc.textFile("./ratings.dat");
```

```
JavaPairRDD<String,String>userIdToLine =  
stringJavaRDD).mapToPair(each
```

```

-> {
String[] split = each.split("::");
return new Tuple2<>(split[0], each);
});

userIdToLine.groupByKey().flatMap((FlatMapFunction<Tuple2<String,
Iterable<String>>, String>) lines -> {
List<String> list = new ArrayList<>();
lines._2.forEach(list::add);

List<String> splittedLine = new ArrayList<>();

int length = list.size();

int train = (9 * length) / 10;

for (; train < length; train++) {
splittedLine.add(list.get(train));
}

return splittedLine.iterator();

}).coalesce(1).saveAsTextFile("./test-ratings.dat");

```

Eğitim verisini oluştururken yazılan Java kod parçasığı ise aşağıdaki gibidir.

```

JavaRDD<String> stringJavaRDD = sc.textFile("./ratings.dat");

JavaPairRDD<String,String>userIdToLine=stringJavaRDD.mapToPair(eac
h->
{
String[] split = each.split("::");

return new Tuple2<>(split[0], each);});

userIdToLine.groupByKey().flatMap((FlatMapFunction<Tuple2<String,
Iterable<String>>, String>) lines -> {
List<String> list = new ArrayList<>();

```

```

lines._2.forEach(list::add);

List<String> splittedLine = new ArrayList<>();

int length = list.size();

int train = (9 * length) / 10;

for (int i = 0; i < train; i++) {

splittedLine.add(list.get(train));

}

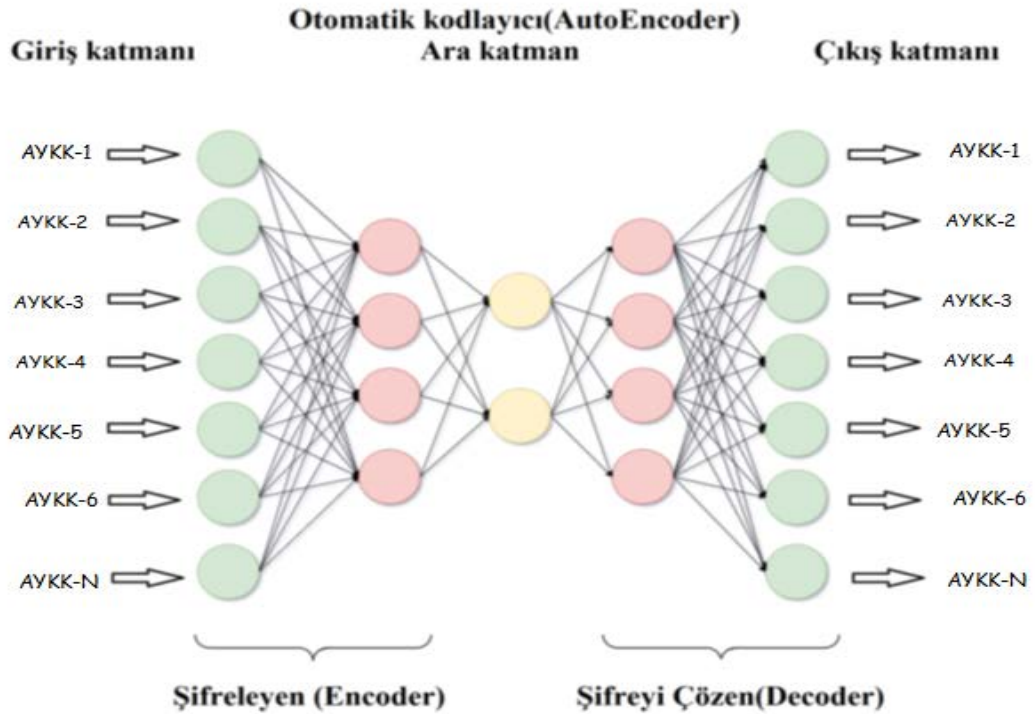
return splittedLine.iterator();

}).coalesce(1).saveAsTextFile("./train-ratings.dat");

```

Bu sayede eğitim seti ve test seti oluşturmuştur.

Çalışmada kullanılan veriler Otomatik Kodlayıcıların eğitimi için uygun hale getirilmelidir. Şekil 7’te Otomatik Kodlayıcıların öneri sistemindeki kullanım yapısı gösterilmiştir.

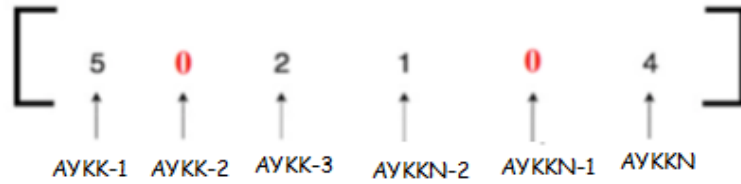


Şekil 7. Öneri sisteminin Otomatik Kodlayıcı ile modellenmesi

Otomatik Kodlayıcı yöntemini öneri sisteminde kullanabilmek için, giriş katmanındaki her bir nöron bir ürünü (ayakkabı) temsil etmelidir. Dolayısıyla

toplam giriş değeri ayakkabı sayısı kadar olmalıdır. Veri seti hazırlanırken her bir kullanıcı için bir eğitim verisi olacaktır. Ayakkabı sayısı 30.000 olduğundan bir eğitim verisi 30.000 rakamdan oluşan bir vektör olacaktır. Derin öğrenme sisteminde eğitim kümesinin ve test kümesinin oluşturulması sırasında; Eğitim kümesinin geçmiş verilerin öğrenildiği yani ağız eğitilmesi için seçilen verilerin bulunduğu küme olarak kabul görmektedir.

Şekil 8’de gösterildiği gibi vektördeki her değer sırası ile bir ayakkabıya verilen puanı temsil etmektedir. Ayakkabıların her vektördeki yeri aynı olmalıdır. Kullanıcıların puan vermediği ayakkabılar için giriş değeri 0 olarak atanacaktır. Öneri sisteminin amacı bu değerleri tahmin etmektir. Veriler test ve eğitim setine bölünürken eğitim ve test seti eşit sayıdadır. Çünkü her kullanıcı hem test hem de eğitim setine dahil edilmiştir. Farklılık ise toplamdaki 0 değerleridir.



Şekil 8. Otomatik Kodlayıcı giriş vektörü

Şekil 8’de gösterilen vektördeki N değeri veri setindeki toplam ayakkabı sayısıdır.

Veriyi vektör haline getiren Python dilinde yazılmış kod parçacığı aşağıda gösterilmiştir.

```
egitim = pd.read_csv('./train-ratings.dat', sep='::', header=None,  
engine='python', encoding='latin-1')  
  
import numpy as np  
  
import pandas as pd  
  
egitim_dizi = np.array(egitim, dtype='int')  
  
kullanici_sayisi = int(max(egitim_dizi[:, 0]))  
  
ayakkabi_sayisi = int(max(egitim_dizi[:, 1]))  
  
egitim_veri = []  
  
for kullanici_id in range(1, kullanici_sayisi + 1):
```

```
ayakkabi_id = egitim_dizi[:, 1][egitim_dizi[:, 0] == kullanıcı_id]
puan_id = egitim_dizi[:, 2][egitim_dizi[:, 0] == kullanıcı_id]
puanlar = np.zeros(ayakkabi_sayisi)
puanlar[ayakkabi_id - 1] = puan_id
egitim_veri.append(puanlar)
egitim_veri = np.array(egitim_veri)
```

## 2. Otomatik kodlayıcı modelinin oluşturulması

Tezde oluşturulan modeller ve test çalışması yapılan optimizasyon algoritmaları Python programlama dilinde TensorFlow kullanılarak yazılmıştır. TensorFlow kütüphanesi araştırmacılara makine öğrenmesi uygulamalarını kurabilecekleri ortam sağlayan açık kaynak kod platformudur. Şekil 8’de gösterildiği gibi vektör haline getirilen veri TensorFlow’da kullanılabilecek şekilde hazır hale getirilmiştir. TensorFlow’u kullanabilmek için aşağıdaki kod parçacığı kullanılmıştır.

```
import tensorflow as tf
```

Giriş değeri 30.000 uzunluğunda bir vektörden oluşmaktadır. Ara katmanlar ve bu katmanlardaki aktivasyon fonksiyonları deneysel olarak belirlenmiştir. Örneğin, ara katmanlardaki nöron sayısının artması başarı oranını artırmamış ve eğitim süresinin artmasına neden olmuştur. Aktivasyon fonksiyonu olarak relu, tanh ve leakly relu fonksiyonları denenmiş ve bu fonksiyonların başarısının çok daha düşük olduğu gözlemlenmiştir. Ara katmanlarda aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Deneysel olarak belirlenen Otomatik Kodlayıcının katmanları sırası ile 80, 40, 80 nörondan oluşmaktadır. Çıkış katman sayısı ise ayakkabi sayısı olan 30.000 uzunluğundadır. Bu işlemlerin yapılabilmesi için ağırlıkların ve bias sabitinin oluşturulması gerekmektedir. Yazılan kod parçacığı aşağıdaki gibidir.

```
W_1 = tf.get_variable(name='agirlik_1', shape=(3952, 80))
W_2 = tf.get_variable(name='agirlik_2', shape=(80, 40))
W_3 = tf.get_variable(name='agirlik_3', shape=(40, 80))
W_4 = tf.get_variable(name='agirlik_4', shape=(80, 3952))
```

```
b_1=tf.get_variable(name='b1',shape=(80), initializer=tf.zeros_initializer())
```

```
b_2=tf.get_variable(name='b2',shape=(40), initializer=tf.zeros_initializer())
```

```
b_3=tf.get_variable(name='b3',shape=(80), initializer=tf.zeros_initializer())
```

### 3. Öneri sistemi modelinin eğitilmesi

Öneri sisteminin eğitilmesi için kullanılan işlem adımları aşağıdaki gibidir.

1. Eğitim verisi okunur.
2. Okunan veri her kullanıcı için vektör haline getirilir. Verilmeyen puanlar 0 olarak yazılır.
3. Ağırlıklar ve bias sabitleri rastgele değerler ile oluşturulur.
4. Eğitim verisi Otomatik Kodlayıcıyı ileri yönlü olarak eğitir. Yazılan kod parçacığı aşağıdaki gibidir.

```
a1 = tf.nn.sigmoid(tf.nn.bias_add(tf.matmul(x, W_1), b_1))
```

```
a2 = tf.nn.sigmoid(tf.nn.bias_add(tf.matmul(a1, W_2), b_2))
```

```
a3 = tf.nn.sigmoid(tf.nn.bias_add(tf.matmul(a2, W_3), b_3))
```

```
cikti = tf.matmul(a3, W_4)
```

5. Giriş katmanında 0 olan verilerin modelin öğrenmesine bir etkisinin olmaması için çıktı değerindeki bu veriler 0 yapılır ve maliyet fonksiyonu hesaplanır.
6. Geri yayılım algoritması ve seçilen optimizasyon algoritması ile ağırlıklar güncellenir.
7. Bu işlem seçilen devir sayısı kadar 4. maddeki işleme gidilerek tekrar yapılır.

Yukarıda belirtilen algoritma ile araştırmacı tarafından hazırlanan veri seti için uygun olan optimizasyon algoritması bulunmaya çalışılmıştır. Uygun olan optimizasyon algoritmasının bulunması YSA'nın eğitim zamanının kısalmasını ve daha başarılı bir öneri sistemi tasarlanmasını sağlayacaktır.

Gradyan Alçalma – Gradient Descent (GA), Hızlı Gradyan Alçalma - Gradient Descent

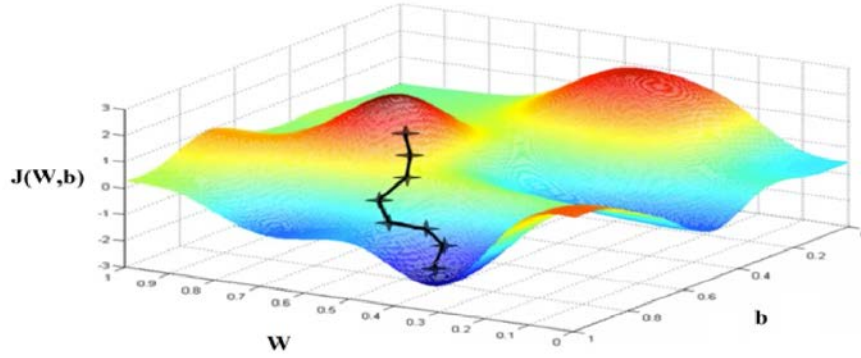


With Momentum (HGA), RMSProp ve Adam (Adaptive Momentum Optimization) algoritmaları veri seti üzerinde denenmiştir.

Modellerin eğitimi sırasında TensorFlow platformunun sağladığı hazır fonksiyonlar kullanılmıştır. Bu sayede geri yayılım algoritmasının yazımı, gradyanların hesaplanması ve optimizasyon algoritmalarının çalışması gibi işlemler TensorFlow platformunda otomatik olarak gerçekleştirilmiştir.

#### 4. Optimizasyon algoritmaları

YSA’larda test hatasını en aza indirmek için, daha önceden rastgele oluşturulan parametre değerlerinin en iyi olması için optimizasyon algoritmaları kullanılmaktadır.



Şekil 9. W ve b parametresine göre maliyet fonksiyonunun değişimi

Şekil 9’da parametre değerlerine göre oluşan maliyet fonksiyonu grafiksel olarak gösterilmiştir. Bütün optimizasyon algoritmalarının amacı bu parametreleri değiştirerek minimum hatanın olduğu parametre kümesini bulmaktır.

Kırmızı alandan başlayan maliyet fonksiyonu her güncelleme ile birlikte azalmaktadır. Her güncellemedeki değişim öğrenme oranı ile belirlenir. Öğrenme oranının artırılması sağa sola salınımları artırmaktadır. Dolayısıyla Şekil 9’da gösterilen diğer minimum noktaların bulunmasına neden olabilir. Küçük öğrenme oranı ise öğrenmeyi geciktirmektedir. Öğrenme oranı veri setine göre deneysel olarak belirlenebilen bir parametredir.

TensorFlow platformunda kullanılan optimizasyon algoritmaları ve sınıfları sırası ile aşağıdaki gibidir.

- GA algoritması ve Gradient Descent Optimizer sınıfı kullanılmıştır.

- HGA algoritması ve MomentumOptimizer sınıfı kullanılmıştır.
- RmsProp algoritması ve RMSPropOptimizer sınıfı kullanılmıştır.
- Adam algoritması ve AdamOptimizer sınıfı kullanılmıştır [30].

Örnek olarak AdamOptimizer'ın TensorFlow platformunda kullanımı aşağıda gösterilmiştir.

```
egitim = tf.train.AdamOptimizer(0.05).minimize(hata)
```

AdamOptimizer sınıfı parametre olarak öğrenme oranı alır.

Tasarlanan öneri sistemindeki Otomatik Kodlayıcıda denenen optimizasyon algoritmaları ve sonuçları aşağıda sırası ile açıklanmıştır.

Gradyan alçalma algoritması

Makine öğrenmesinde en yaygın kullanılan optimizasyon algoritması GA algoritmasıdır. GA'da maliyet fonksiyonunun minimum olması için ağırlıklar ve sabitler güncellenir. Maliyet fonksiyonunda belirtilen W ve b değerlerinin gradyanları hesaplanır.

Hesaplanan gradyan değerlerine göre W ve b değerleri güncellenir.

Değerler güncellenirken rastgele bir öğrenme oranı belirlenir. Formülde gösterilen öğrenme oranıdır.

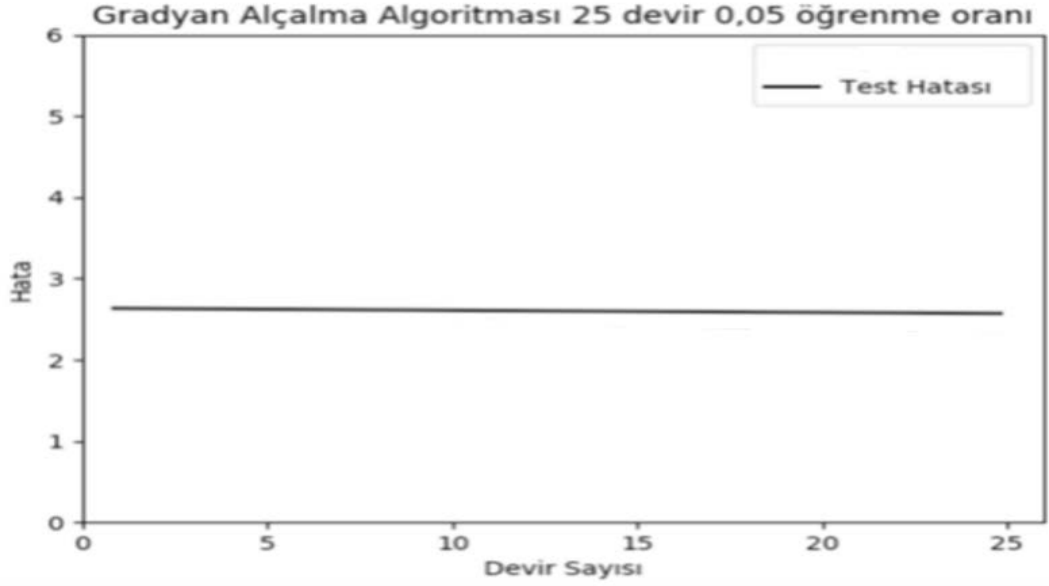
Tasarlanan öneri sistemi GA algoritması ile üç farklı şekilde test edilmiş ve sonuçları aşağıda verilmiştir.

Öğrenme oranı 0,05 seçilmiştir. Eğitim toplam 25 devirde tamamlanmış her devir sonunda test verisi tasarlanan modelde test edilmiştir.

Çizelge 7. GA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,942
5	2,876
10	2,874
15	2,826
20	2,653
25	2,524

Çizelge 7'de görüldüğü gibi 2,942 puan ile başlayan test hatası 2,524 puanda son bulmuştur.



Şekil 10. GA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

GA algoritması kullanılarak 0,05 öğrenme oranı ve 25 devirde Otomatik Kodlayıcının öğrenemediği gözlemlenmiştir.

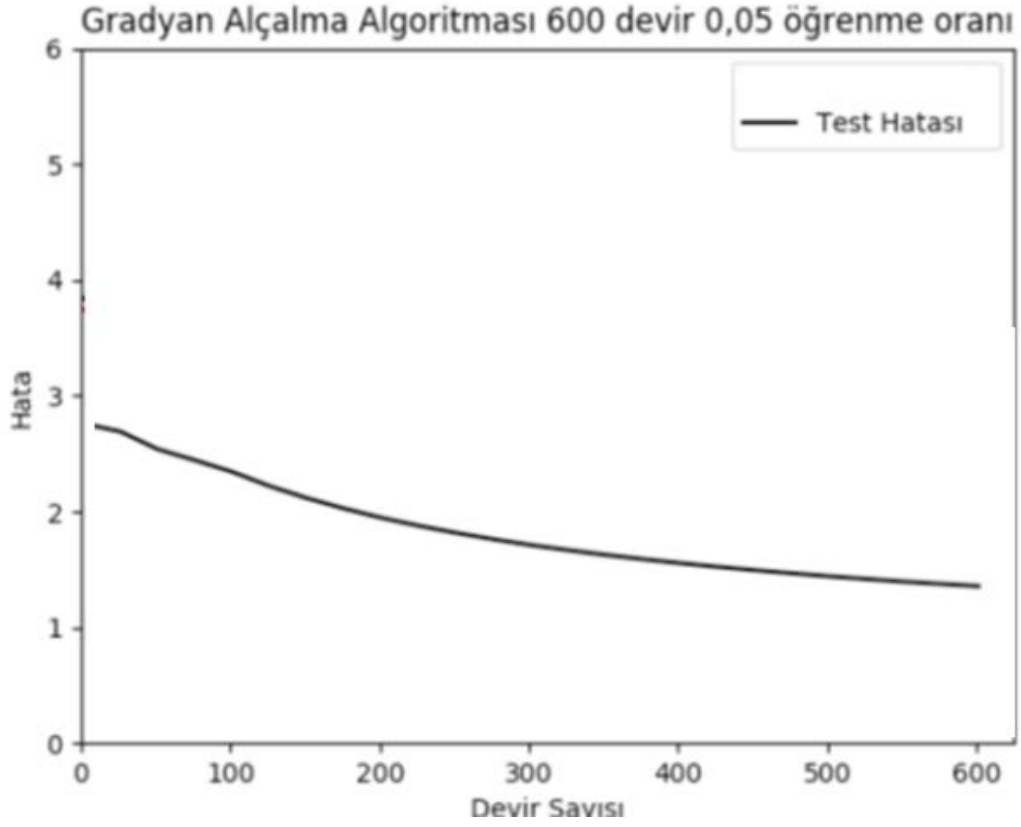
İkinci denemede öğrenme oranı değiştirilmemiş devir sayısı 600'e çıkarılmıştır.

Denemeler de devir sayısını daha fazla artırmanın öğrenmeye çok fazla etkisi olmadığı ve eğitim zamanını artırdığı görülmüştür. Örneğin, 25 devir ile 600 devirlik bir eğitim zamanı arasında 8,5 kat gibi bir zaman farkı vardır.

Çizelge 8. GA Algoritması 0,05 öğrenme oranı ve 600 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,945
300	2,848
500	2,476
600	2,467

Çizelge 8'de görüldüğü gibi 2,945 puan ile başlayan test hatası 2,467 puanda son bulmuştur.



Şekil 11. GA Algoritması 0,05 öğrenme oranı ve 600 devir ile test hatası

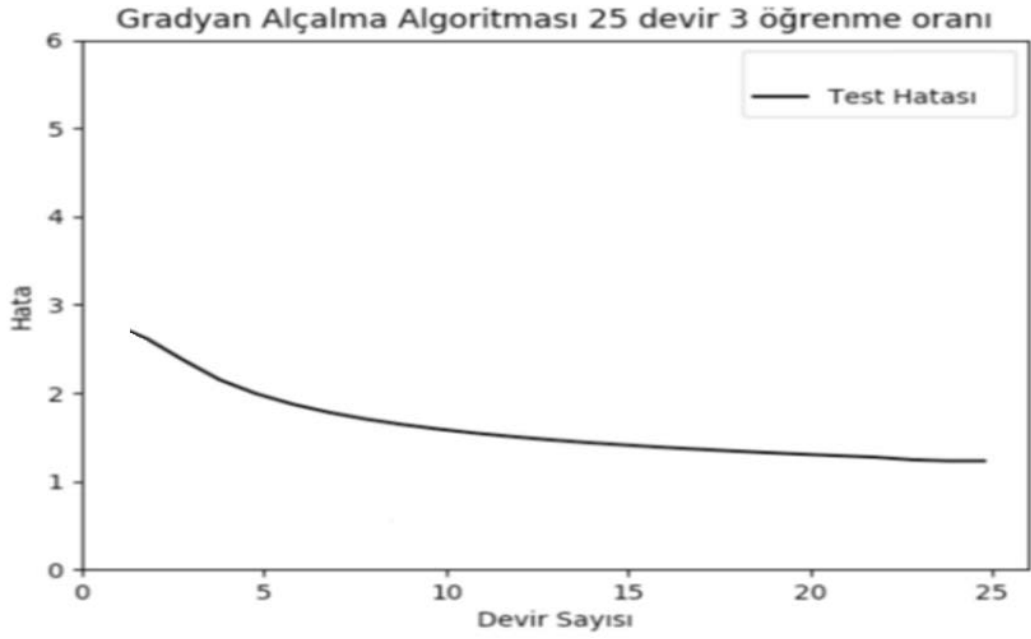
Şekil 11’de görüldüğü gibi yaklaşık 300 devirden sonra Otomatik Kodlayıcının öğrenme hızı azalmaya başlamıştır.

Üçüncü deneme ise 25 devir ve 3 öğrenme oranı ile yapılmıştır.

Çizelge 9. GA Algoritması 3 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,696
5	2,594
10	2,495
15	2,318
20	2,215
25	2,133

Çizelge 9’da görüldüğü gibi 2,696 puan ile başlayan test hatası 2,133 puanda son bulmuştur.



Şekil 12. GA Algoritması 3 öğrenme oranı ve 25 devir ile test hatası

Şekil 12’de görüldüğü gibi 10’ununcu devirden sonra Otomatik Kodlayıcının öğrenmesi yavaşlamıştır.

Hızlı gradyan alçalma algoritması

Bu algoritmada bir önceki devirden hesaplanan gradyanlar hesaba katılır.

Matematiksel olarak yukarıda formülleri verilmiştir. 0 ile 1 arasında belirlenen değeri hesaplanan gradyanların bir önce hesaplanan gradyanlara ne kadar yakın olacağını belirler.

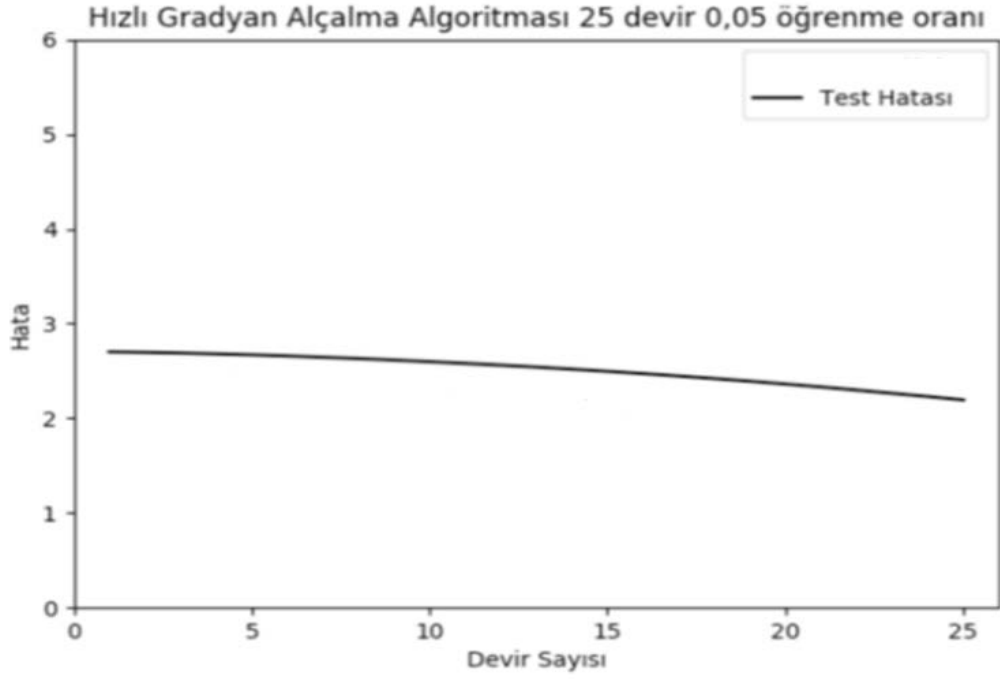
Tasarlanan öneri sistemi HGA algoritması ile üç farklı şekilde test edilmiş ve sonuçları aşağıda verilmiştir.

HGA algoritmasında yapılan denemede ilk olarak öğrenme oranı 0,05 seçilmiştir. Eğitim toplam 25 devirde tamamlanmış, her devir sonunda test ve eğitim verisi tasarlanan modelde test edilmiştir.

Çizelge 10. HGA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,846
5	2,815
10	2,742
15	2,640
20	2,506
25	2,338

Çizelge 10’da görüldüğü gibi 2,846 puan ile başlayan test hatası 2,338 puanda son bulmuştur.



Şekil 13. HGA Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

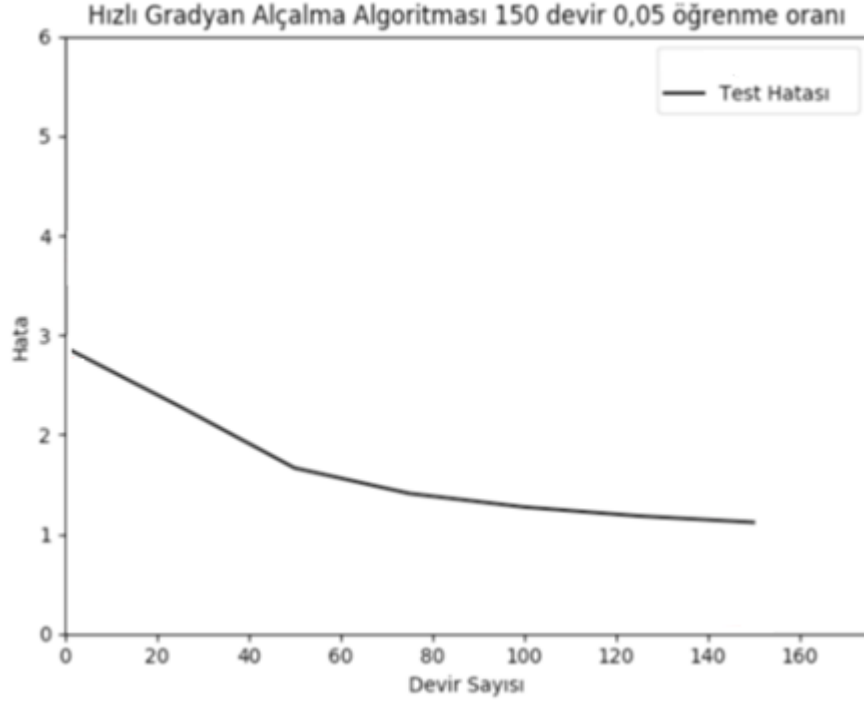
Şekil 13’de görüldüğü gibi 25 devir ve 0,05 öğrenme oranında Otomatik Kodlayıcının öğrenmesi istenilen seviyeye gelmemiştir.

HGA Algoritması üzerinde yapılan ikinci denemede devir sayısı artırılmıştır.

Çizelge 11. HGA Algoritması 0,05 öğrenme oranı ve 150 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,835
25	2,264
50	1,646
75	1,390
100	1,253
125	1,164
150	1,080

Çizelge 11’de görüldüğü gibi 2,835 puan ile başlayan test hatası 1,080 puanda son bulmuştur.



Şekil 14.HGA Algoritması 0,05 öğrenme oranı ve 150 devir ile test hatası

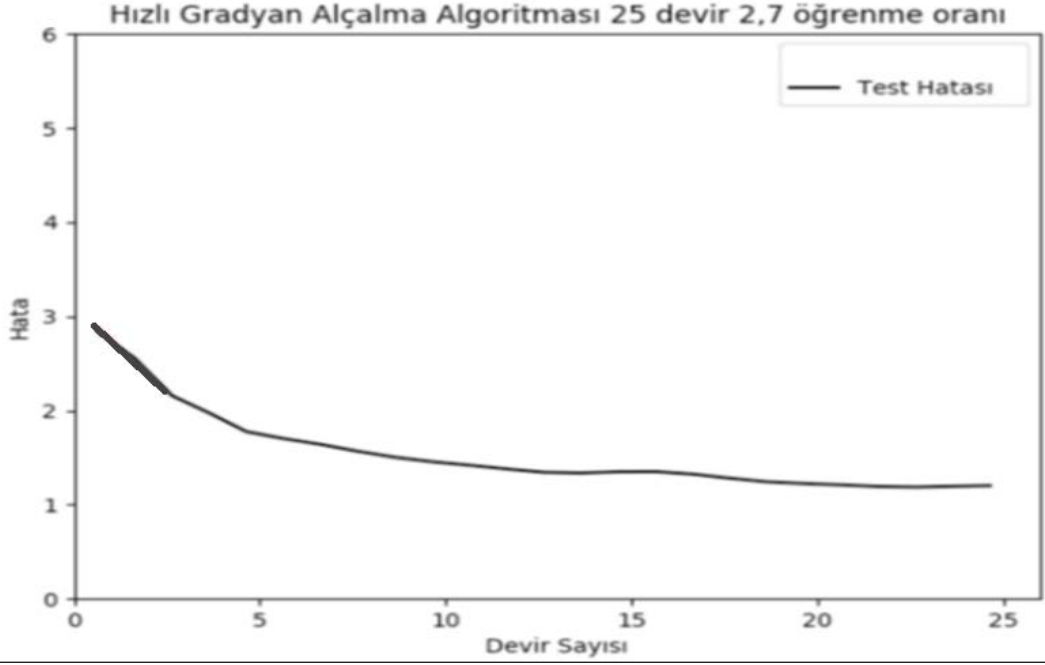
Şekil 14’de ise 50’nci ve 80’inci devirlerden sonra Otomatik Kodlayıcının öğrenme hızı yavaşlamıştır. Yapılan denemelerde 150 devirin artırılması öğrenmeyi artırdığı görülmüştür. Fakat eğitim zamanını artırmıştır. Örneğin, devir sayısının 250’ye çıkarılması hatayı indirmiş, eğitim zamanını 1,7 kat artırmıştır.

Yapılan son denemede ise öğrenme oranı 2,7’ye çıkarılmış devir sayısı ise 25 olarak belirlenmiştir.

Çizelge 12. HGA Algoritması 2,7 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,711
5	2,648
10	2,328
15	1,221
20	1,096
25	2,010

Çizelge 12’da görüldüğü gibi 2,711 puan ile başlayan test hatası 2,010 puanda son bulmuştur.



Şekil 15.HGA Algoritması 2,7 öğrenme oranı ve 25 devir ile test hatası

Şekil 15’de görüldüğü gibi Otomatik Kodlayıcının HGA algoritması ile öğrenme hızı

15’inci devirden sonra düşmüştür. Ayrıca öğrenme oranını arttırmanın test hatasına etkisinin yüksek olmadığı görülmüştür.

RmsProp algoritması

RmsProp algoritmasında bir önceki devirden hesaplanan gradyanların kareleri hesaba katılır. Aşağıdaki formülde görülen değeri ile bir önceki iterasyondaki değere ne kadar yakın olacağı belirlenir.

W ve b değeri ise aşağıdaki formüllerde gösterildiği gibi hesaplanır

Tasarlanan öneri sistemi RmsProp algoritması ile üç farklı şekilde test edilmiş ve sonuçları aşağıda verilmiştir.

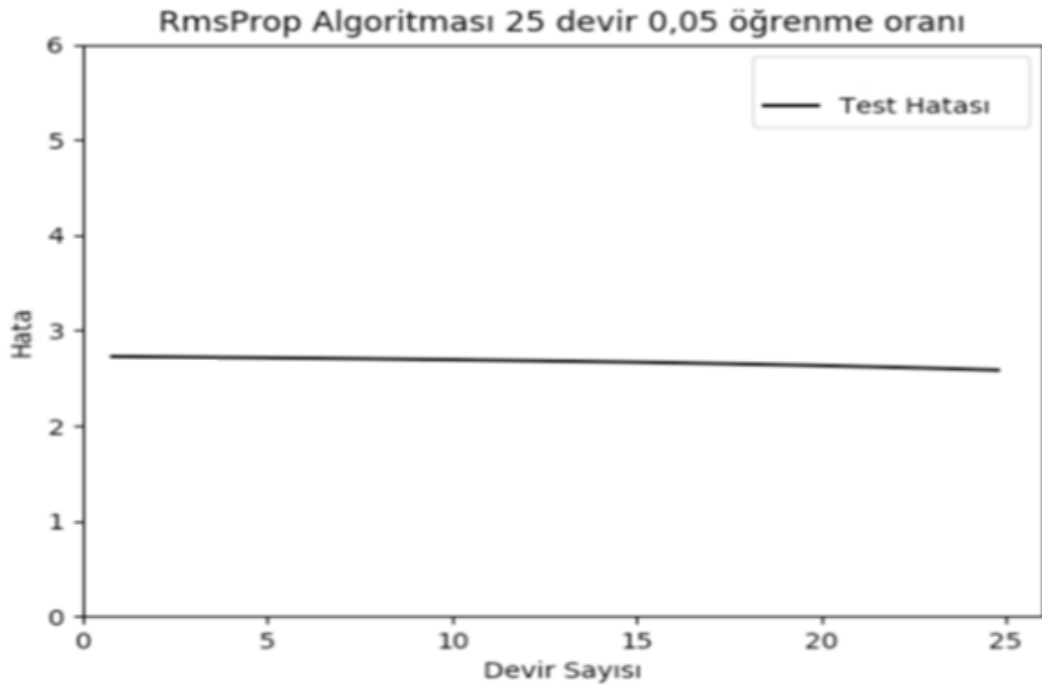
Yapılan ilk denemede öğrenme oranı 0,05 seçilmiş, eğitim ise 25 devirde tamamlanmış, her devir sonunda test ve eğitim verisi tasarlanan model ile test edilmiştir.



Çizelge 13. RmsProp Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,581
5	2,426
10	2,814
15	2,780
20	2,774
25	2,703

Çizelge 13’de görüldüğü gibi 2,581 puan ile başlayan test hatası 2,703 puanda son bulmuştur.



Şekil 16. RmsProp Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

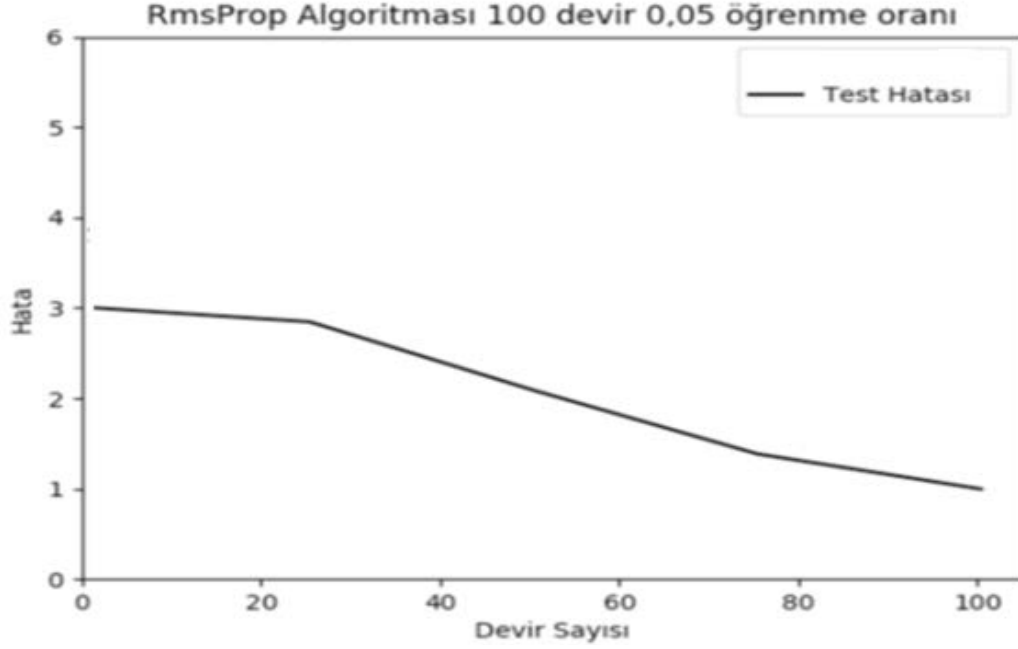
Şekil 16’de görüldüğü gibi 25 devir ve 0,05 öğrenme oranında, Otomatik Kodlayıcı RmsProp algoritması ile istenilen seviyeye gelmemiştir.

RmsProp algoritması üzerinde yapılan ikinci denemede devir sayısı artırılmıştır.

Çizelge 14. RmsProp Algoritması 0,05 öğrenme oranı ve 100 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,860
25	2,712
50	1,940
75	1,236
100	1,827

Çizelge 14’de görüldüğü gibi 2,860 puan ile başlayan test hatası 1,827 puanda son bulmuştur. Devir sayısının daha fazla arttırılması RmsProp algoritmasının öğrenmesinde yüksek bir artış sağlamamıştır.



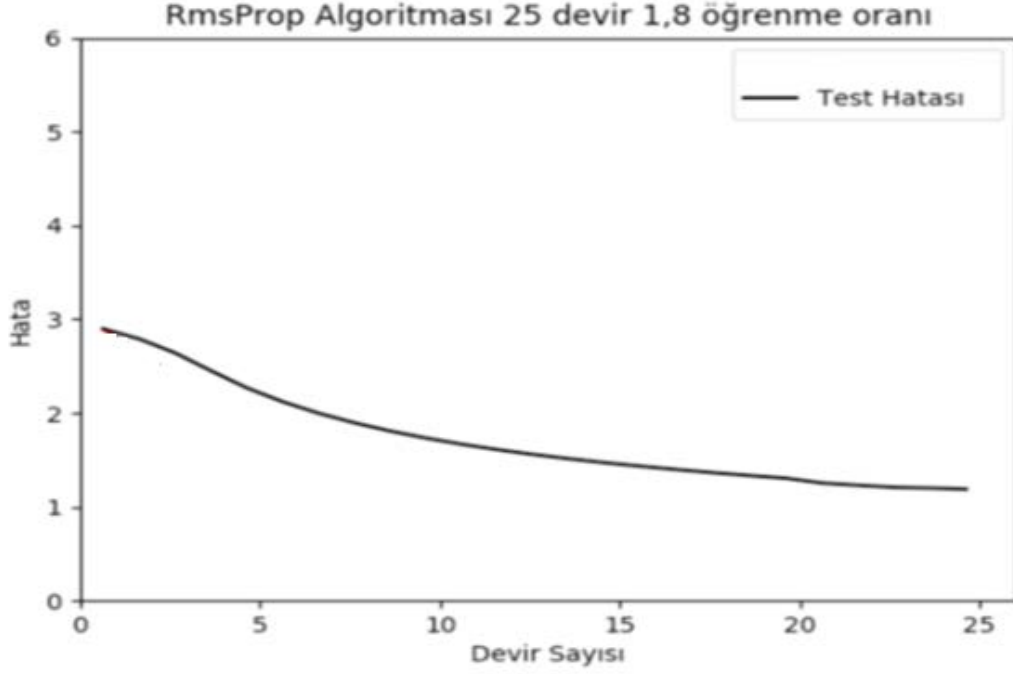
Şekil 17. RmsProp Algoritması 0,05 öğrenme oranı ve 100 devir ile test hatası

Şekil 17’de görüldüğü gibi 80’inci devirden sonra Otomatik Kodlayıcının öğrenme hızı düşmüştür. Yapılan son denemede ise öğrenme oranı 1,8’e çıkarılmış, devir sayısı ise 25 olarak belirlenmiştir.

Çizelge 15. RmsProp Algoritması 1,8 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,714
5	2,120
10	1,598
15	1,325
20	1,161
25	1,040

Çizelge 15’de görüldüğü gibi 2,714 puan ile başlayan test hatası 1,040 puanda son bulmuştur.



Şekil 18. RmsProp Algoritması 1,8 öğrenme oranı ve 25 devir ile test hatası

Şekil 18’de görüldüğü gibi 15’inci devirden sonra öğrenme Otomatik Kodlayıcının öğrenme hızı yavaşlamıştır.

#### Adam Algoritması

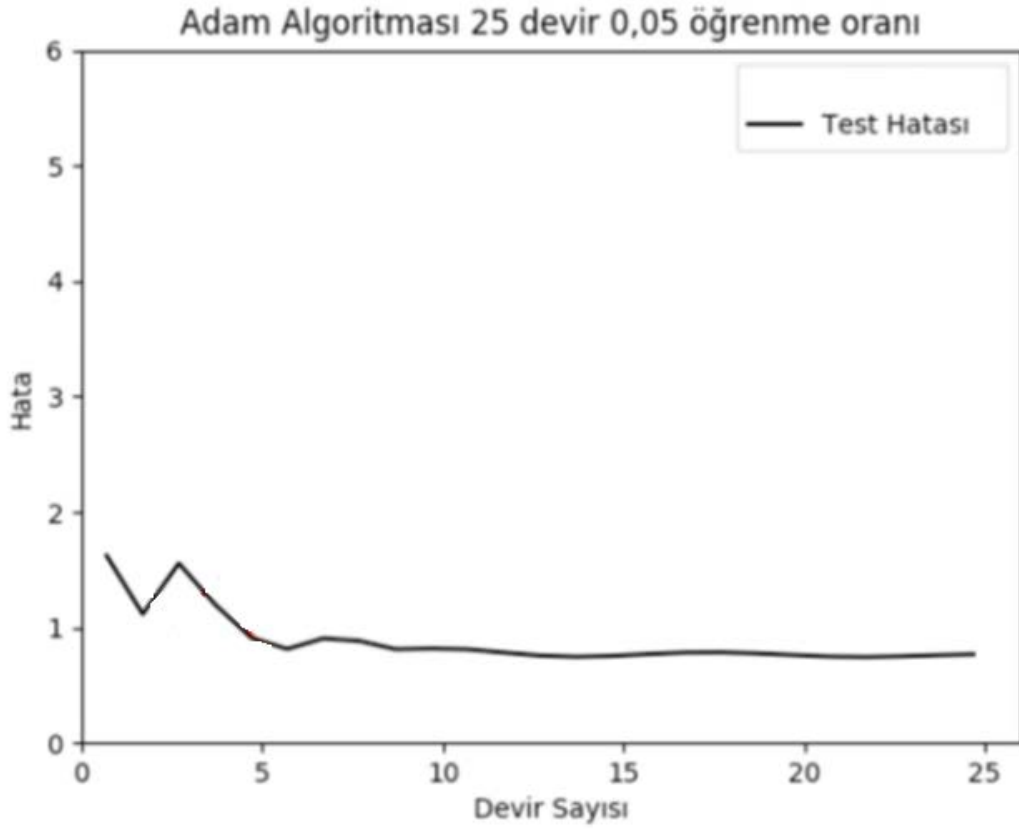
Tasarlanan öneri sistemi Adam algoritması ile 0,05 öğrenme oranı ve 25 devirde yapılan test sonuçları aşağıda verilmiştir.

Çizelge 16’de görüldüğü gibi 2,125 puan ile başlayan test hatası ise 1,013 puan olarak değişiklik göstermiştir.

Çizelge 16. Adam Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

Devir Sayısı	Test Hatası
1	2,125
5	1,508
10	1,312
15	1,249
20	1,208
25	1,013

Şekil 19’da görüldüğü gibi 10’uncu devirden sonra Otomatik Kodlayıcının öğrenmesi yavaşlamıştır. Adam algoritması üzerinde yapılan farklı öğrenme oranları test hatası fazla etkilememiştir. Ayrıca devir sayısının artırılması da test hatası fazla etkilememiştir.



Şekil 19. Adam Algoritması 0,05 öğrenme oranı ve 25 devir ile test hatası

## 5. Giriş veri miktarının sistem başarısına etkisi

Araştırmacı tarafından hazırlanan veri setine göre en uygun optimizasyon algoritmasının Adam algoritması olduğu görülmüştür. Derin öğrenme yöntemlerinde veri miktarı arttıkça başarının arttığı görülmektedir. Artan veri miktarının tasarlanan modeldeki etkisini incelemek için beş ayrı deney yapılmıştır. Optimizasyon algoritması olarak ise Adam algoritması kullanılmıştır.

Öneri sistemleri üzerine çalışma yapan Strömqvist, matrisi çarpanlara ayırma yönteminde matristeki 0 olan kullanıcıların vermediği puanların, öneri sistemi üzerindeki performansını incelemiş ve matristeki boşluğun azalmasının öneri sisteminin performansını artırdığını görmüştür.

## V.TARTIŞMA VE SONUÇ

Öneri sistemi modelini oluştururken araştırmacı tarafından hazırlanan veri setine en uygun optimizasyon algoritmasının 25 devir 0,05 öğrenme oranı ve 1,013 test hatası ile Adam algoritmasının en uygun olduğu bulunmuştur. Bu durumun önemi, oluşturulan modelin eğitim sürecini kısaltması ve çok daha başarılı olan bir model oluşturulmasına imkân sağlayacaktır.

Veri sayısındaki yükseliş ve düşüşün YSA'ya olan etkisine bakıldığında, çok daha fazla sayıda veri kullanılarak eğitilen YSA'ların kullanıcı kişiler ile ürünler arasında mevcut olan ilişkiyi çok daha iyi bir şekilde algılamaya başladığı neticesine varılmıştır.

Öneri sistemlerine ait başarı düzeyini yükseltmek için yalnızca bir modelden faydalanmaktansa hibrid yaklaşımlar kullanılabilir. Böylece oluşturulan öneri sisteminde var olan bir takım dezavantajlı durumlar da giderilebilir. Oluşturulan öneri sistemi içerisinde, sistem içerisine yeni dâhil olan ürünler meydana getirilen vektörlerde bulunamayacağından, öneri sistemine giremezler. Nitekim bu durum soğuk başlangıç durumunu meydana getirecektir. Söz edilen durum sistem içerisine yeni bir ürün dâhil olan yeni bir ürün puanlanıncaya kadar sürer. Bahsedilen problem içerik temelli öneri tekniği aracılığıyla çözülebilir. Ortaya konan tez çalışmasında tavsiye edilen öneri sistemi ile içerik bazlı öneri tekniği bir araya getirilip hibrid bir tavsiye sistemi yaratılabilir.

Veri seti içerisinde yer alan ratings.dat dosyası kapsamında yer alan zaman bilgisi, dizayn edilen öneri sistemi içerisinde kullanılmamıştır. Kullanıcı olan kişinin hangi ürünü ne zaman puanladığını gösteren bu bilgi başka çalışmalar için kullanılabilir. Söz edilen veriler kronolojik olarak sıralanıp kullanıcıların daha sonra ürünü görme talep edebileceği ürün, derin öğrenme tekniklerinden bir tanesi olan RNN'ler vasıtasıyla ön görülebilir

## VI. KAYNAKLAR

### KİTAPLAR

ALTAŞ, D. (2013). **İstatistiksel Analiz**, 1. Baskı-Şubat,

CANBAY, P. (2021). **Sağlıkta Yapay Zeka: Makine Öğrenmesi Yöntemleri Ve Uygulamaları.**

FAUSETT, L. (1994). **Fundamentals of Neural Networks: Architectures, Algorithms and Applications**, Prentice Hall.

HAİR, J.F., BLACK, W.C., BABİN, B.J., ANDERSON, R.E. (2009). **Multivariate Data Analysis**, 7th Ed., PrenticeHall, London, UK

KOHONEN, T. (1987). **Int. Conf. on AI**, State o the Art in Neural Computing, pp. 1-79, 1-90

MOCKENHAUPT, A. (2021). **Maschinelles Lernen.** Digitalisierung und Künstliche Intelligenz in der Produktion. Wiesbaden: Springer Vieweg.

ÖZTEMEL, E. (2003). **Yapay Sinir Ağları**, Papatya Yayıncılık, İstanbul.

ŞEN, Z. (2004). **Yapay Sinir Ağları İlkeleri**, Su Vakfı Yayınları, İstanbul.

TABACHNICK, B.G. ve FIDELL, L.S. (2007). **Using Multivariate Statistics** (5th ed.). Pearson Education, Inc. / Allyn and Bacon.

ZHU, Y., LI, H., LIAO, Y., WANG, B., GUAN, Z., LIU, H., CAI, D. (2017, Ağustos). **What to Do Next: Modeling User Behaviors by Time-LSTM.** In IJCAI (pp. 3602-3608).

### MAKALELER

AKAY, B., KAYNAR O., DEMİRKOPARAN F. (2017) Derin Öğrenme Tabanlı Onerici Sistemler. **UBMK.**

- BLOEMER, M. M. J., BRIJS, T., VANHOOF, K. ve SWINNEN, G. (2003);  
“Comparing Complete And Partial Classification For Identifying  
Customers At Risk”, **International Journal Of Research In  
Marketing**, Vol:20, Issue:2, s. 117-131.
- GÖKAY, E. G. ve TAŞKIN, Ç. (2005). “Araç Rotalama Problemlerinin İki  
Aşamalı Çözümünde Genetik Algoritma Kullanımı”. **Gazi  
Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, 7.1: 1-17.
- KAELBLING, L. P., LITTMAN, M. L., ve MOORE, A. W. (1996).  
“Reinforcement learning: A survey.” **Journal of artificial  
intelligence research**, 4, 237-285.
- KEITH D. F. (2019). “A brief history of machine learning. dataversity web  
page,” **Homepage, Data education, Smart Data News**, Articles, &  
Education.
- ÖZGÜR, S. B. “Algoritmalar, Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme  
ve Uygulamaları: Beşeri Fayda Üretiminin Yazılımlar Tarafından  
Karşılanması.” **Ekonomi ve Yönetim Araştırmaları Dergisi**, 10(1),  
1-29.

## **ELEKTRONİK KAYNAKLAR**

- HİDASİ, B., KARATZOGLOU, A., BALTRUNAS, L., TİKK, D. (2015).  
Session-based recommendations with recurrent neural networks. arXiv  
preprint arXiv:1511.06939.
- SALAKHUTDİNOV, R., MNİH, A., HİNTON, G. (2007, Haziran). Restricted  
Boltzmann machines for collaborative filtering. In Proceedings of the  
24th international conference on Machine learning (pp. 791-798).  
ACM.
- SEDHAIN, S., MENON, A. K., SANNER, S., XİE, L. (2015, Mayıs). Autorec:  
Autoencoders meet collaborative filtering. In Proceedings of the 24th  
International Conference on World Wide Web (pp. 111-112). ACM

STRUB, F. ve MARY, J. (2015, Kasım). Collaborative filtering with stacked denoising autoencoders and sparse inputs. In NIPS workshop on machine learning for eCommerce.

## **TEZLER**

ÇOBAN, Ö. (2016). Metin sınıflandırma teknikleri ile türkçe twitter duygu analizi. Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü.

SANTOLAYA, D.S. (2017) Using Recurrent Neural Networks to Predict Customer Behavior From Interaction Data. Yüksek Lisans Tezi. Amsterdam Üniversitesi, Amsterdam(7 Temmuz 2017)

SARICAOĞLU, C. (2019). Sözcüksel Analiz Kullanarak Kötü Niyetli Url'leri Derin Öğrenme Teknikleri ile Tespit Etme. Gazi Üniversitesi Yüksek Lisans Tezi.



## **ÖZGEÇMİŞ**

**ALİ SARIKAYA**

### **EĞİTİM**

**Yüksek Lisans – (2022) - İstanbul Aydın Üniversitesi - Bilgisayar Mühendisliği** Anabilim dalı – Bilgisayar Mühendisliği (Tezli) Bölümü (Türkçe)

**Lisans – (2011-2014) Anadolu Üniversitesi (Açık Öğretim) İşletme**

**Önlisans— (2009-2011) İstanbul Aydın Üniversitesi – Bilgisayar Programcılığı (İkinci Öğretim)**

### **İŞ TECRÜBESİ**

**Greyder Ayakkabı – E-Ticaret ve Dijital Pazarlama Müdürü (2021-)**

**Jumbo (Karaca Grup) – E-Ticaret Satış Müdürü (2021-2021)**

**Gülaylar Group – E-Ticaret Müdürü (2018-2021)**

**Gülaylar Group – Kıdemli Yazılım Uzmanı (2016-2018)**

**Tekzen – Yazılım Uzmanı (2010-2016)**

### **YETKİNLİKLER**

#### **Yabancı Dil**

Türkçe – Anadil

İngilizce—Orta

#### **Bilgisayar Bilgisi**

Microsoft Visual Studio, Sql Server Manegment Studio, Photoshop, Ofis programları, C#, Asp.Net Web Form, Asp.Net MVC, MsSQL, JQuery / JSON, WCF / SOAP, Web API / REST / RESTful, CSS / CSS3, Java / Javascript, Html / Html5

### **Sertifikalar**

BT AKADEMİ – ASP. NET ile Web Programlama (2012)

BT AKADEMİ – JQuery ve JQuery UI Eğitimi (2013)