



**KTO KARATAY ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS
PROGRAMI**

**DERİN ÖZKODLAYICI AĞLAR İLE ÖNERİ SİSTEMLERİNDE TAHMİN
DOĞRULUĞUNUN ARTIRILMASI**

Akif IŞIK

Yüksek Lisans Tezi

**KONYA
Temmuz 2022**

DERİN ÖZKODLAYICI AĞLAR İLE ÖNERİ SİSTEMLERİNDE TAHMİN
DOĞRULUĞUNUN ARTIRILMASI

Akif IŞIK

KTO Karatay Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Mühendisliği Ana Bilim Dalı
Elektrik ve Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi Ali ÖZTÜRK

Konya
Temmuz 2022

BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini KTO Karatay Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan “Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge” kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve KTO Karatay Üniversitesi Açık Erişim Sisteminde erişime açılır.

Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.¹

Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ... ay ertelenmiştir.²

Tezimle ilgili gizlilik kararı verilmiştir.³⁴

05 Temmuz 2022

Akif IŞIK

¹ MADDE 6(1) Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

² MADDE 6(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

³ MADDE 7(1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

⁴ MADDE 7(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

ETİK BEYAN

KTO Karatay Üniversitesi Lisansüstü Eğitim Enstitüsü Tez/Proje Hazırlama ve Yazım Kurallarına uygun olarak Dr. Öğr. Üyesi. Ali ÖZTÜRK danışmanlığında tarafımdan üretilen bu tez çalışmasında; sunduğum tüm veri, enformasyon, bilgi ve belgeleri bilimsel etik kuralları çerçevesinde elde ettiğimi, tüm değerlendirme, analiz, bulgu ve sonuçları bilimsel usullere uygun olarak sunduğumu, tez çalışmasında yararlandığım kaynakların tümüne bilimsel normlara uygun biçimde atıfta bulunarak kaynak gösterdiğimi, tezimin kaynak gösterilen durumlar dışında özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

05 Temmuz 2022

Akif IŞIK

TEŐEKKÜR

Tez alıřmam boyunca yardımını ve katkısını eksik etmeyen, bilgi ve deneyimi ile her konuda destek olan deęerli danıřman hocam Dr. Öğr. Üyesi Ali ÖZTÜRK'e, teőekkür ederim. Ayrıca Yüksek Lisans eęitim sürecinde yanımda olan ve beni destekleyen deęerli aileme teőekkür ederim.

05 Temmuz 2022

Akif IŐIK

ÖZET

Akif IŞIK

Derin Özkodlayıcı Ağları ile Öneri Sistemlerinde Tahmin Doğruluğunun Artırılması

Yüksek Lisans Tezi

Konya, 2022

Bilgiye erişimin her geçen gün daha fazla hızlanması, internette bulunan veri miktarını da giderek artırmaktadır. Bu büyük veri içerisinde ihtiyaç duyulana kolay erişim de önemli hale gelmektedir. Öneri Sistemleri günümüzde kullanıcıların istedikleri veriye hızlı ve doğru bir şekilde ulaşması için birçok farklı alanda kullanılmaktadır. Bu kapsamda kullanıcıların daha önce yapmış olduğu tercihler göz önünde tutularak gelecekte olası tercihleri tahmin etmeye çalışırlar. Öneri sistemlerinde farklı yöntemler kullanılmakta olup bunların başında İşbirlikçi Filtreleme yöntemi gelmektedir. İşbirlikçi filtreleme, birbirine yakın kullanıcıların benzer ilgi ve beğenilere sahip olacağı yaklaşımı ile çalışır. İşbirlikçi filtreleme yöntemi, Bellek tabanlı ve Model tabanlı olarak sınıflandırılabilir. Bu çalışmada işbirlikçi filtreleme algoritmalarından model bazlı tahmin hesaplama adımı için Derin Özkodlayıcı Ağlar ile Gürültü Giderici Özkodlayıcı Ağlar kullanılarak Öneri Sistemlerinin performansında iyileştirme hedeflenmiştir. Farklı gürültü oranları ile karşılaştırılan ağlardan 0,3 gürültü oranına sahip model en başarılı model olmuştur. Belirlenen parametreler kullanılarak oluşturulan modellerle yapılan testlerde Özkodlayıcı Ağların bir kolu olan Gürültü Giderici Özkodlayıcı Ağların daha başarılı sonuçlar verdiği görülmüştür.

Anahtar Kelimeler

Öneri Sistemleri, işbirlikçi filtreleme, derin özkodlayıcı ağlar, gürültü giderici özkodlayıcı ağlar

ABSTRACT

Akif IŞIK

Increasing Prediction Accuracy in Recommender Systems with Deep Autoencoder

Networks Master's Thesis

Konya, 2022

The fact that access to information becomes more important day by day increases the amount of data available on the Internet. Easy access to the information needed from this big data is also becoming important. Suggestion Systems are used in many different areas today so that users can access the data they want quickly and accurately. In this context, it is tried to predict possible preferences in the future by taking into account the preferences that the users have made before. Different methods are used in recommendation systems, and the first of these is the Collaborative Filtering method. Collaborative filtering works on the principle that close users will have similar interests and likes. Collaborative filtering method can be classified as Memory-based and Model-based. In this study, it is aimed to improve the performance of the Recommender Systems by using Deep Autocoders and Denoising Autoencoders for the model-based prediction calculation step, which is one of the collaborative filtering algorithms. Among the networks compared with different noise ratios, the model with a noise ratio of 0.3 was the most successful model. In the tests made with the models created by using the determined parameters, it has been seen that the Denoising Autoencoders, which is a branch of the Autocoders, give more successful results.

Keywords

Recommendation Systems, Collaborative Filtering, Deep Autoencoders, Denoising Autoencoders

İÇİNDEKİLER

BİLDİRİM	i
ETİK BEYAN.....	ii
TEŞEKKÜR.....	iii
ÖZET.....	v
ABSTRACT	vi
İÇİNDEKİLER	vii
TABLolar DİZİNİ	viii
ŞEKİLLER DİZİNİ.....	ix
KISALTMALAR DİZİNİ.....	x
1. GİRİŞ	1
2. LİTERATÜR ÖZETİ	4
3. MATERYAL VE YÖNTEMLER.....	10
3.1. Kullanılan Veri Seti.....	10
3.2. Öneri Sistemleri ve Kullanılan Yöntemler	12
3.2.1. İşbirlikçi Filtreleme	12
3.2.2. İçerik Tabanlı Öneri Sistemleri	17
3.2.3. Hibrit Sistemler.....	18
3.3. Derin Öğrenme	19
3.3.1. Yapay Sinir Ağları	20
3.3.2. Öz Kodlayıcı Ağlar	21
3.3.3. Gürültü Giderici Öz Kodlayıcılar	23
4. DENEYSEL SONUÇLAR	26
4.1. Deney Metodolojisi ve Değerlendirme Metrikleri	26
4.1.1. Kök Ortalama Kare Hatası.....	27
4.2. Hiper Parametrelerin Özellikleri	27
4.3. Öneri Modelinin Oluşturulması ve Eğitilmesi	29
4.4. Deneysel Sonuçlar	29
5. SONUÇ	40
KAYNAKLAR	43
ÖZGEÇMİŞ	47

TABLolar DİZİNİ

Tablo 1. Film Veri Seti	10
Tablo 2. Film Veri Seti	11
Tablo 3. Puan Veri Seti	11
Tablo 4. Oluşturulan Puan Veri Seti	26
Tablo 5. Deneysel çalışmada kullanılan parametreler	28
Tablo 6. Deneysel çalışmada kullanılan parametreler	30
Tablo 7. Deneysel çalışmada elde edilen sonuçlar (Selu).....	31
Tablo 8. Deneysel çalışmada elde edilen sonuçlar (Elu)	32
Tablo 9. Deneysel çalışmada elde edilen sonuçlar (Relu)	34
Tablo 10. Deneysel çalışmada elde edilen sonuçlar (Selu).....	35
Tablo 11. Deneysel çalışmada elde edilen sonuçlar (elu).....	37
Tablo 12. Deneysel çalışmada elde edilen sonuçlar (Relu)	38
Tablo 13. Farklı modellerin MovieLens 1M veri setinde Test RMSE Karşılaştırması ..	41

ŞEKİLLER DİZİNİ

Şekil 1. Öneri Sistemleri Hiyerarşik Yapısı.....	1
Şekil 2. Yapay Sinir Ağlarının Yapısı.....	20
Şekil 3. Öz Kodlayıcı Ağlarının Temel Yapısı	22
Şekil 4. Gürültü Giderici, Özkodlayıcı Ağlarının Temel Yapısı	25
Şekil 5. Adam Optimizasyon Algoritması (Selu)	31
Şekil 6. Adam Optimizasyon Algoritması (Elu).....	33
Şekil 7. Adam Optimizasyon Algoritması (Relu).....	34
Şekil 8. SGD Optimizasyon Algoritması (Selu)	36
Şekil 9. SGD Optimizasyon Algoritması (elu)	37
Şekil 10. SGD Optimizasyon Algoritması (Relu).....	39

KISALTMALAR DİZİNİ

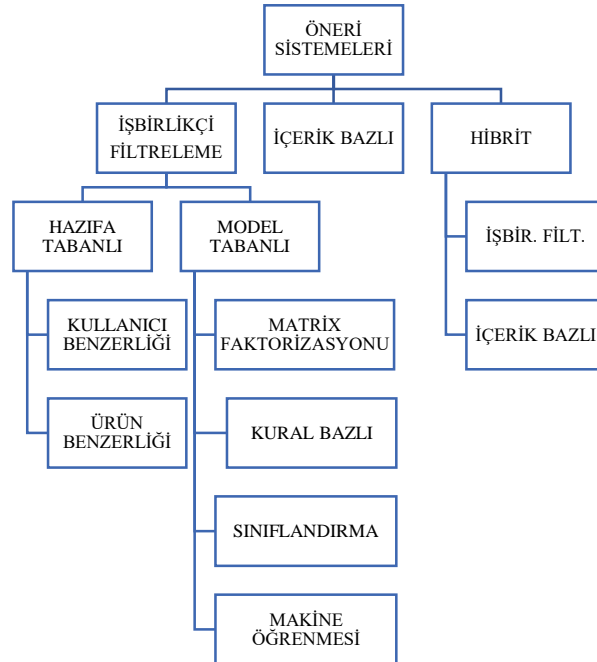
Kısaltma	Açıklama
YSA	Yapay Sinir Ağı
RBM	Restricted Boltzman Machine
SDA	Stacked Denoising AutoEncoder
RNN	Recurrent Neural Network
DBN	Deep Belief Network
TDA	Tekil Değer Ayırıştırma
RMSE	Root Mean Square Error
SGD	Stochastic Gradient Descent
LSTM	Long Short-Term Memory Network
RELU	Rectified Linear Unit

1. GİRİŞ

İnternette verilerin sürekli büyüyen yapısı nedeniyle, kullanıcıların bu büyüyen veri içerisinden istediği bilgiye ulaşması da her geçen gün zorlaşmaktadır. Bundan dolayı kullanıcının istediği veriye ulaşma sorununu gidermek için her kullanıcıya özel kişiselleştirilmiş veri sunulmasıyla mümkündür. Öneri sistemleri bu noktada kullanıcıya kişiselleştirilmiş öneriyi sunmasıyla kısa sürede istenilen ve uygun bilgiye erişimi sağlayan sistemlerdir. Başta e-ticaret, film, şarkı gibi alanlar olmak üzere kullanıcıya kişiselleştirilmiş öneri sunan sistemler son yıllarda her geçen gün artmaktadır.

Öneri sistemleri, yeni kullanıcıların tercihlerini tahmin etmek için kullanıcıların önceki seçimlerini kullanarak bu bilgi karmaşıklığını gidermeye çalışırlar. Bu bağlamda kullanıcıların karar verme sürecini kolaylaştırmak için öneri sistemleri kullanılır.

Kullanıcıların önceki alışkanlıklarını ve davranışlarını inceleyerek, uygun olan ürünleri belirleyen sistemler öneri sistemleridir (Özcan, 2010). Öneri sistemleri büyük miktardaki işlenmemiş veriden istenen bilgiyi seçmek için kullanılırlar. Bu sistemler üç ana başlık altında toplanabilir. Bunlar; işbirlikçi filtreleme, içeriğe dayalı filtreleme ve hibrit yaklaşımlara dayanmaktadır. Öneri sistemlerinin yapısı Şekil 1’de verilmiştir.



Şekil 1. Öneri Sistemleri Hiyerarşik Yapısı

İşbirlikçi filtremenin temelinde birbirine benzeyen kullanıcıların benzer beğeni ve ilgi alanına sahip olacağı düşüncesi bulunmaktadır. Aktif kullanıcı için ona benzeyen kullanıcıların ilgili ürüne yaptıkları değerlendirmelere bakarak aktif kullanıcıya ilgili ürününe vereceği değer hesaplanır.

İçerik bazlı filtreleme yönteminde ise öneride bulunurken içeriklerin arasındaki benzerlik değerlendirilir. Bunu ürün için düşündüğümüzde satın alınan ürünün içerik bilgileri, bakılan ürüne olan içerik benzerliği değerlendirmeye alınır. İki ürün arasındaki benzerliği bulmak için ürünlerin geçmiş kayıtlarına bakılır. Ürünlerin satın alınma zamanı bu ilişkiye örnek verilebilir. Aynı zaman aralığında satın alınan ürünler birbirine benzer ürünler olarak görülür ve iki üründen birini satın alanlara ikinci ürün bir öneri olarak sunulur (Yu & Sun, 2010).

Hibrid modelde ise İşbirlikçi Filtreleme yönteminin ile İçerik bazlı filtreleme yönteminin sağladığı avantajlar birleştirilerek öneri sisteminin başarısını artırmak hedeflenir. Böylece diğer yöntemlerin dezavantajları giderilmeye çalışılır. İşbirlikçi Filtreleme kullanan bir öneri sisteminde, sisteme yeni giren veriler ile ilgili henüz bir bilgi olmadığı için, öneri sistemine dahil olamazlar. Bu soruna soğuk başlangıç sorunu denilmektedir. Bu durum sisteme yeni giren veri için tercih bilgisi dahil olana kadar devam eder. Ancak bu dezavantajı sisteme içerik bazlı öneri metodu dahil edilip Hibrid bir sistem ile giderilebilir.

İşbirlikçi Filtreleme Yönteminde kullanıcıların arasındaki ilişki değerlendirilir yada bir ürün üzerinde diğer kullanıcıların görüşleri değerlendirilerek yakın ilgi alanına sahip veya yakın türdeki içeriğe sahip ürünler gruplandırılır. İşbirlikçi Filtrelemenin temel düşüncesi, “Eğer bir kullanıcı geçmişte komşuları ile aynı fikirde ise gelecekte de aynı fikirde olma olasılığının yüksek olduğudur” (Miller, 2003).

İşbirlikçi Filtreleme yöntemleri iki sınıfa ayrılabilir. Bunlardan ilki hafıza tabanlı işbirlikçi filtreleme, ikincisi ise model tabanlı işbirlikçi filtreleme olarak isimlendirilir. Hafıza tabanlı işbirlikçi filtreleme yönteminde kullanıcıya bir tavsiye sunabilmek için tüm matris üzerinden değerlendirme yapılır (Adomavicius & Tuzhilin, 2005). Model tabanlı işbirlikçi filtrelemede ise elimizde bulunan verileri bir makina öğrenme algoritması ile bir model oluşturulur ve kullanıcıya tavsiye sunulurken tüm matris yerine oluşturulan model üzerinden hesaplamalar yapılmaktadır (Koren & Bell, 2009).

Bu çalışma 4 ayrı başlık altında toplanmıştır. Bunlar; Giriş, Literatür araştırması, Materyal ve Yöntemler ile Deneysel sonuçlar bölümleridir. Giriş bölümünde Öneri Sistemlerinin ne olduğu anlatıldıktan sonra, öneri sistemlerinin hangi ana başlıklar altında toplandığı, İşbirlikçi Filtreleme, İçerik bazlı ve Hibrit sistemlerinin ne olduğundan bahsedilmiştir. Bu çalışmanın içeriği ve amacı hakkında bilgi verilmiştir. İkinci bölümde Literatür çalışması yapılmıştır. Genel olarak öneri sistemlerinin ortaya çıkmasından, öneri üretmede kullanılan tekniklerin gelişiminden bahsedilmiştir. Üçüncü bölümde kullanılan veri setinin açıklanması, derin özkodlayıcı ağlardan Gürültü Giderici Özkodlayıcının ne olduğu anlatılmıştır. Dördüncü bölümde çalışmaya ait deneysel sonuçlar açıklanmıştır. Gürültü Giderici Özkodlayıcı kullanarak sistem performansı artıracak optimizasyon algoritmaları karşılaştırılmıştır.

2. LİTERATÜR ÖZETİ

Öneri sistemleri üzerine literatür incelemesi yapıldığında dünyada ve ülkemizde pek çok farklı yöntem için çalışmaların yapıldığı görülmektedir. Bu yöntemler İçerik Bazlı Filtreleme, İşbirlikçi Filtreleme ve Hibrit Model Yöntemleri olarak üç ana başlık altında değerlendirilebilir.

Literatürde İşbirlikçi Filtreleme yöntemi ilk kez 1992 yılında Tapestry ismiyle Goldberg tarafından ortaya konulan çalışmada öneri sisteminde kullanılmış ve İşbirlikçi Filtreleme metodunun öneri sistemlerinde kullanılmasının başlangıcıdır (Goldberg et al., 1992).

1998 yılında yapılan çalışmada Breese ve arkadaşları tarafından işbirlikçi filtreleme öneri sistemleri, model ve hafıza tabanlı olmak üzere iki ayrı başlıkta incelenmiştir. İlgili çalışmada Bayesian ağlarının daha iyi sonuçlar verdiği görülmüştür (Breese et al., 2013).

İşbirlikçi Filtreleme sisteminin hafıza tabanlı ve model tabanlı olarak ikiye ayrıldığından bahsedilmiştir. Hafıza tabanlı yaklaşımlarda iki ayrı başlıkta incelenmektedir. Bunlar kullanıcı ve ürün tabanlı yaklaşımlar olarak adlandırılır. Kullanıcı tabanlı yaklaşımlarda aktif kullanıcı için yapılacak önerilerde aktif kullanıcıya benzeyen komşu diğer kullanıcılar belirlenip seçilerek öneri sunulur. Ürün tabanlı sistemlerde ise aktif kullanıcının ilgilendiği ürün baz alınarak bu ürüne benzeyen komşu diğer ürünler belirlenip seçilerek öneriler sunulur. Ürün tabanlı modelin temelinde ürünler arasındaki benzerliğin belirlenip bulunan ilişki kullanılarak öneriler sunulmadır. Burada ürünlere kullanıcılar tarafından verilen puanlar kullanılabilir. Bir benzerlik hesaplama metodu ile verilen puanlar kullanılarak bütün ürünler arasındaki birbiri ile olan benzerlikleri saptanır. Birbirine benzerlikleri belirlenen ürünlerden aktif kullanıcının daha önce incelemeyeği ve ilgisini çekeceği düşünülen ürünler öneri olarak sunulur. Hafız tabanlı işbirlikçi filtreleme yöntemi için literatürde birçok çalışma yapılmıştır. Hem kullanıcı tabanlı işbirlikçi filtreleme yaklaşımı için hem de ürün tabanlı işbirlikçi filtreleme yaklaşımı için yapılan çalışmalarda en yaygın olarak kullanılan algoritma k en yakın komşu algoritması olmuştur (Sarwar et al., 2000).

Ürün tabanlı işbirlikçi filtreleme sistemlerinde benzerlik hesaplama ve öneri sunmak için daha önce ürünlere verilen puanların bulunduğu iki boyutlu ürün kullanıcı matrisi kullanılır. Oluşturulan matrisin özelliği ise bir kullanıcının bir ürün için yalnızca bir değerlendirmesinin bulunmasıdır. Bu nedenle bu ve benzeri sistemler tek bir kriteri

üzerinden benzerlik kurmaya çalışırlar. Halbuki bir ürünün birden fazla özelliği bulunur ve bunlar ayrı ayrıda değerlendirilebilir. Farklı kriterlere yapılan değerlendirmeler ilgili ürün için daha detaylı bilgiye sahip olunmasını sağlar. Bunu farkederek Adomavicius ve Kwon, birden çok kriterin kullanılmasının tek bir kriteri baz alan sistemlere kıyasla daha iyi sonuçlar alınacağını düşünmüş ve ispat etmişlerdir. Örnek olarak bir film önerisi için genel tek bir değerlendirmeyi esas almak yerine oyunculuk, senaryo ve görsel efektler gibi birden fazla kriteri değerlendirerek film ile ilgili daha derin bir bilgiye sahip olunur. Adomavicius ve Kwon tek kriterli hafıza tabanlı yaklaşımı çok kriterli hafıza tabanlı sistemlere uygulamışlardır (Adomavicius & Kwon, 2009). Hafıza tabanlı işbirlikçi filtreleme yönteminden farklı olarak birden fazla kriter olduğundan benzerlik hesaplamaları için yeni metotlar geliştirmişlerdir. Geleneksel benzerlik tabanlı yaklaşımlar kullanılabilmesi gibi çok kriterli veriler çok boyutlu uzaydaki veriler olarak ele alınarak uzaklık metrikleri sayesinde aralarındaki benzerliklerin hesaplanabileceğini göstermişlerdir.

Çok büyük sayıda verinin kullanıldığı öneri sistemlerinde ise daha çok Model tabanlı İşbirlikçi Filtreleme yöntemi tercih edilmektedir. Bu yöntemde kullanıcıların daha önceki tercihlerinden bir model kurulur. Kurulan model ile yeni gelen kullanıcılar için öneri üretilir. İlk zamanlarda Model Tabanlı İşbirlikçi Filtreleme yönteminde model kurmak için Kural tabanlı yaklaşımlar, Kümeleme yöntemleri, Bayes Ağları sıkça kullanılmıştır. 1998 yılında Breese ve arkadaşları yaptıkları çalışmada model tabanlı işbirlikçi filtreleme yaklaşımını kullanmışlardır (Breese et al., 2013). Bu çalışmada benzer tercihleri yapan kullanıcılar bir demetleme modeli ile aynı grup içerisinde toplanır. Belirlenen sınıflar içerisindeki herhangi bir aktif kullanıcının değerlendirmeleri bağımsız kabul edilerek Naive Bayes formülüne göre çıkartılır. Lyle H. Ungar ve Dean P. Foster ise ürünleri göz önünde tutarak kullanıcıların gruplandırılması algoritmalarını İşbirlikçi Filtreleme alanında yaptıkları çalışmalarda kullanmışlardır. Gruplandırma aşamasında K Means Kümelemesini ve Gibbs örnekleme yöntemlerini farklı varyasyonlar ile denemişlerdir. Ortaya çıkan modelde farklı farklı özelliğe sahip ürünlerin gruplandırılması başarı ile yapılmıştır (Ungar & Foster, 1998). Bir diğer çalışmada David N. Pennock tarafından 2000 yılında yapılmıştır. Çalışmada yeni bir yaklaşım hedeflenmiş ve kişiliği tanıma ismiyle yine İşbirlikçi Filtreleme temelli sistem geliştirilmiştir. Geliştirilen yöntemde

gruplandırma için Pearson Benzerlik yönteminde kullanılan yeni bir ölçeklendirme metodu eklenmiştir (Pennock et al., 2013)

Baltunas ve arkadaşları 2009 yılında filtreleme yönteminde alternatif yeni iki ayrı yöntem önermişlerdir. Bunlardan ilki ön filtreleme yöntemiyle kullanıcılardan gelen geribildirimleri kullanarak geliştirilen filtreleme sistemidir. Sitemin temelindeki amaç CARS olarak bilinen içeriğe bağlı öneri sistemlerinin gelen geri bildirimler sayesinde zaman içerisinde kendini iyileştirmesidir. Tasarlanan sistemde kullanıcılara ait profiller zamana göre alt profiller oluşturularak öneri üretme sırasında oluşturulan alt profiller göz önüne alınır. İkinci yöntemde ise kullanıcı değerlendirmelerini baz alan, İşbirlikçi filtreleme yönteminin geliştirilmesine katkı sunmak amacıyla kullanıcıların değerlendirmelerini yaparken tek başına yapılan bu değerlendirmelerin yeterli olmayacağını, bu nedenle daha önce toplanmış olan kullanıcının bağlamsal verilerinin dahil edilmesi gerektiği ortaya koyulmuştur. Bu nedenle değerlendirilmesi istenen bir ürün için farklı özniteliklerde de farklı değerlendirmelerin toplanması ile daha başarılı sonuçların alınacağı ifade edilmiştir (Baltrunas & Amatriain, 2009). Bir diğer çalışmada Rafter, İşbirlikçi Filtreleme yönteminde geçmişten itibaren günümüze kadar kullanılan methodlar hakkında geniş kapsamlı bir araştırma yapmış ve öneri üretmek için işbirlikçi filtreleme yönteminde kullanıcıdan etkileşimli olarak bilgi alan bir metodu geliştirmiştir (Rafter, 2010).

Wang ve arkadaşları yaptıkları çalışmada İşbirlikçi Filtreleme sisteminin geliştirilmesi amacıyla ürün tanımlamada kullanılan etiketleri değerlendirmeye almışlardır. İşbirlikçi filtreleme metodunda öneri doğruluğunu etkileyen ana unsurlardan biride verilen oyların yeterli sayıda olmasıdır. Bu sorunu gidermek için ürünleri tanımlamakta kullanılan etiketlerin sisteme dahil edilebileceğini göstermişlerdir (Wang et al., 2010).

Kişiselleştirilmiş tavsiye sunmayı hedefleyen öneri sistemleri, kullanıcılara ilgilendikleri sayfalara yakın önerilerde bulunmak için de işbirlikçi filtreleme sistemleri sıkça kullanır. Lian tarafından 2011 yılında yapılan çalışmada bu konu ele alınmıştır. Kullanıcıların sayfa ziyaret ağılıkları ve sayfada geçirdikleri süre dikkate alınmıştır. Kullanılan kümelere yönteminde bu metrikler hesaba katılmıştır (Lian, 2011).

Model tabanlı İşbirlikçi filtreleme yöntemi kullanan Ruslan Salakhutdinov tarafından 2007 yılında yapılan çalışmada Netflix firmasının yayınladığı veri seti kullanılmıştır.

Çalışmada model oluşturmak için yapay sinir ağı türü olan RBM'leri (Restricted Boltzman Machine) kullanmıştır. Sonuçları Netflix'in kullanmış olduğu öneri sisteminden yaklaşık olarak %6 daha iyi sonuç verdiğini göstermişlerdir (Salakhutdinov et al., 2007).

İşbirlikçi filtreleme ile ilgili yeni bir yaklaşım Hasan Bulut ve Musa Milli tarafından 2014 yılında yapılan çalışmada sunulmuştur. Bu yaklaşımda oluşturulan sistemin son aşaması olan tahmin hesaplama adımında iki farklı yöntem önerilmiştir. İlk yöntemde tahmin hesabı aşamasında aktif kullanıcı ile komşu kullanıcılar arasında değerlendirilen pozitif benzerlik yönteminin tersi yorumlanmıştır. Pozitif benzerlik yönteminde aktif kullanıcı ile komşu kullanıcılar ne kadar bir birine benzerlik gösteriyorsa, bu düşüncenin tersi olarak aktif kullanıcılar ile de negatif benzerliğe sahip kullanıcılar o kadar uzaktır. Bu sebeple negatif benzerliğe sahip kullanıcıların tahmin hesaplama adımında yer almasını önermiştir. İkinci yöntemde de kullanıcıların ürün değerlendirmelerinde verdikleri puanların standart sapmalarının hesaplamaya dahil edilmesidir (Bulut & Milli, 2016).

Model tabanlı yaklaşımların artmasıyla 2015 yılında Suvash Sedhain özkodlayıcıları kullanmıştır. Bu çalışma Netflix veri seti ve MovieLens veri seti ile yapılmıştır. Alınan sonuçlar daha önce RBM ve Matris çarpanlarına ayırma metodları ile yapılan çalışmalardan daha iyi olduğunu göstermiştir (Sedhain et al., 2015).

Öneri sistemleri kendine e-ticaret alanında da yaygın bir kullanım alanı bulmuştur. Zhou ve arkadaşları tarafından 2018 yılında yapılan bir çalışmada mikro davranışların öneri sistemi üzerindeki etkilerini incelemişlerdir. Deneysel çalışmalarda mikro davranışların öneri sistemi üzerinde olumlu etkileri olduğunu göstermişlerdir (Zhou et al., 2018).

Veri problemlerinde sıkça kullanılan ve başarılı sonuçlar alınan Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Network) 2015 yılında Balazs Hidasi tarafından yapılan çalışmada kullanılmıştır. Kullanıcıların ziyaret verilerini zaman bağılı olarak sıralamış, iki ayrı veri setinde yaptığı deneylerde elde ettiği verileri paylaşmıştır (Hidasi et al., 2015).

2017 yılda ise Bahriye Akay tarafından yapılan çalışmada Derin İnanç ağları (Deep Belief Network) kullanmıştır. MovieLens veri setini kullandığı çalışmada farklı optimizasyon algoritmaları ile test yapmıştır. Elde ettiği sonuçlara göre Adamax algoritmasının aşırı öğrenmeye neden olduğunu tespit etmiştir (Akay et al., 2017).

RNN'nin başka bir çeşidi olan LSTM'ler ile Yu Zhu tarafından 2017 yılında yapılan çalışmada Uzun kısa süreli Bellek (LSTM)'lere zaman aralıkları ilave ederek farklı üç model çalışmıştır. Önerdiği yöntemlerin uyguladığı iki ayrı veri setinde de öneri sisteminin başarısını artırdığını göstermiştir (Zhu et al., 2017).

Kullanıcı davranışlarının etkisi konusunda çalışma yapan Daniel Sánchez 2017 yılında yine RNN'leri kullanmıştır. Kullanıcıların ürün satın alma, görüntüleme gibi bilgilerini sıralı bir veriye çevirmiştir. İki ayrı veri setinde RNN'leri kullanarak aktif kullanıcının davranışını tahmin etmeye çalışmıştır (Santolaya, 2017).

Bir diğer çalışmada araştırmacılar Jaccard benzerlik metodu için yeni bir yöntem önermişlerdir. Bu yöntemde aktif kullanıcının minimum oylanmamış ortak ürünleri ile en yakın komşularının da maksimum ortak ürünlerinin sayısına öncelik vermiştir (Bag et al., 2019).

MovieLens veri seti ile yapılan başka bir çalışmada Ahuja ve arkadaşları K en yakın komşu algoritması ile KMeans Clustering algoritmalarını kullanmışlardır. Elde ettikleri sonuçların var olan sonuçlardan daha iyi değer verdiğini göstermişlerdir (Ahuja et al., 2019).

TDA (Tekil Değer Ayırıştırma) algoritmasını kullanan Patra ve Ganguly k en yakın komşu algoritması ile elde ettikleri sonuçları iyileştirmeye çalışmışlardır. Tekil Değer Ayırıştırma filmlere ait yorumlarda yer alan bazı gizli kalıp ifadelerin boyut azaltma tekniği ile bulunmasını sağlamıştır. Dezavantajı ise hesaplama ve kaynak kullanımını açısından zayıf kalmasıdır. MovieLens 1M veri seti ile yapılan çalışmada Tekil Değer Ayırıştırma konusunda birkaç farklı ön işleme algoritması arasında kıyaslama yapılmıştır. Diğer taraftan Tekil Değer Ayırıştırma algoritması seyreklik sorununa sahip öneri sitemlerinde bir çözüm olarak sıkça kullanılmaktadır. K en yakın komşu algoritmasının da beraberinde kullanılması bu sorunda daha iyi sonuçlar alınmasını amaçlamıştır. K en yakın komşu algoritması aktif kullanıcıya en yakın k tane komşusunu bulup, eksik değerlendirme tahmini için kullanılır. Eksik değerler için yakın komşuların ortalaması hesaplanarak yapıldığında, benzerlik hesabı için Öklid hesabı kullanıldığında k en yakın komşu tabanlı Tekil Değer Ayırıştırma yöntemi en iyi sonucu vermiştir (Patra & Ganguly, 2019).

E-Ticaret ile ilgili başka bir çalışma Waleed Abdullah tarafından yapılan ürün öneri sistemidir. Veri seti olarak 1023 ürün bulunan ve yaklaşık yarım milyonluk satış verisi

bulunan deęerlendirmeler kullanılmıřtır. alıřmada İřbirliki filtreleme, FP- Byme ve Apriori algoritmaları birlikte kullanılmıř olup sık rnt madencilięi konusunda kıyaslamalar yapılmıřtır. Benzerlik hesaplama yntemi iinse Jaccard, Pearson ve Kosins benzerlięi kullanılmıřtır. Elde edilen sonularda FP-Byme algoritması ile Jaccard benzerlięinin en iyi deęerleri verdięi grlmřtr (Abdullah, 2019).

ok kriterli rn neri siteleri konusunda 2020 yılında Sadıkoęlu ve Okkaloęlu tarafından yapılan alıřmada iřbirliki filtreleme ynteminde farklı aęırlıklandırma metotları kullanarak sistemin doęruluęunu artırmaya alıřmıřlardır. Benzerlik hesabı iin kullanılan nem aęırlıklandırma metodunu kullanıcılar yerine rnler arasında kullanmıřlardır. Deney sonularında nerilen yntemin mevcut deęerlerin zerinde bir performansa sahip olduęunu gstermiřlerdir (Sadıkoęlu & Okkaloęlu, 2020).

3. MATERYAL VE YÖNTEMLER

3.1. Kullanılan Veri Seti

Bu çalışmada MovieLens veri seti kullanılmıştır (Movielens, 2021). Movielens veri kümesi Minnesota Üniversitesinde GroupLens araştırma projeleri kapsamında oluşturulmuş bir veri kümesidir. GroupLens araştırma projesi, Minnesota Üniversitesi Bilgisayar Bilimleri ve Mühendisliği Bölümü üyelerinden oluşan bir grup tarafından yürütülmüştür. Grup üyelerinin GroupLens Projesi dışında İşbirlikçi Filtreleme ve öneri sistemleri alanında geliştirilmiş pek çok projesi bulunmaktadır.

Veri kümesi araştırmacıların kullanımına açık olarak oluşturulmuş ve farklı büyüklükte veri kümeleri bulunmaktadır. Ayrıca kullanıcılara ait meslek, yaş, bölge, cinsiyet gibi temel bilgileri de içermektedir. Çalışmada kullanılan veri seti ise 1 Milyon oya sahip veri kümesidir. Bu versiyonunda 6040 kullanıcı tarafından 3952 tane filme verilmiş 1 Milyon oy bulunmaktadır. Veri kümesi dört kısımdan oluşur.

README: Veri kümesi ile ilgili genel bilgilendirmeler bulunan dosyadır.

movies.dat: Filmlerin id'leri, isimleri ve hangi türe ait olduğu gösteren verilerden oluşur. Tablo 1'de örnek kısım gösterilmiştir.

Tablo 1. Film Veri Seti

ID	İSİM	TÜR
1	Toy Story (1995)	Animation Children's Comedy
2	Jumanji (1995)	Adventure Children's Fantasy
3	Grumpier Old Men (1995)	Comedy Romance
4	Waiting to Exhale (1995)	Comedy Drama
5	Father of the Bride Part II (1995)	Comedy

users.dat: Kullanıcıların id, cinsiyet, yaş, meslek ve posta kodu verileri bulunmaktadır. Tablo 2'de ise bu dosyaya ait örnek veriler gösterilmiştir. Meslek sütununda bulunan rakamların hangi meslek grubuna ait olduğu README dosyasında yer almaktadır.

Tablo 2. Film Veri Seti

ID	CİNSİYET	YAŞ	MESLEK	POSTA KODU
1	F	1	10	48067
2	M	56	16	70072
3	M	25	15	55117
4	M	45	7	2460
5	M	25	20	55455

ratings.dat: Bir kullanıcının yaptığı değerlendirmede bir filme kaç puan verdiği ve ne zaman verdiği bilgisinden oluşmaktadır. Dosyada toplam 3952 farklı film ve 6040 farklı kullanıcı bulunmaktadır. Kullanıcıların filmlere için yaptığı değerlendirmelerde puanlar 1 ile 5 arasında değişmektedir. Tablo 3’de örnek veriler gösterilmiştir. Veri seti bu şekilde toplam 1.000.209 adet puan verisine sahiptir. Zaman sütununda bulunan zaman bilgisi kullanıcının puanı verdiği zamanın 1 Ocak 1970’ten itibaren geçen sürenin saniye cinsinden değeridir.

Tablo 3. Puan Veri Seti

KULLANICI ID	FILM ID	PUAN	ZAMAN
1	1193	5	978300760
1	661	3	978302109
1	914	3	978301968
1	3408	4	978300275
1	2355	5	978824291

Veri kümelerinin kurulacak modelde kullanılabilmesi için farklı ön işlemlere tabi tutulması gerekmektedir. Çalışmada veri analizlerinin Python yazılım dilinin pandas kütüphanesinde yapılması için dataframe veri yapısına dönüştürülmesi gerekmektedir. Dataframe pandas kütüphanesi içerisinde hızlı ve doğru veri analizleri yapmaya olanak sağlayan çok boyutlu veri yapılarıdır. Ayrıca dataframe içerisinde tutulan veriler yine Python ortamında oldukça kolay bir şekilde “to_csv()” fonksiyonu ile csv dosyalarına aktarılır. Google Colab ortamında Python programlama dili ile kullanılacak verilerin bulunduğu dat dosyaları csv dosyalarına dönüştürülmüştür.

Eğitim veri seti, modelini eğitmek için kullanılırken, test veri seti doğrulama amaçlı kullanılmaktadır.

3.2. Öneri Sistemleri ve Kullanılan Yöntemler

Öneri sistemleri, kullanıcıların kişisel zevklerine uygun ürünlere veya hizmetlere kolay ve hızlı bir şekilde ulaşabilmelerine olanak tanır (Schafer et al., 2001). Çevrimiçi ürün ve hizmet sağlayıcıları öneri sistemlerinden yararlanarak kullanıcılarına kişiselleştirilmiş ürünler veya hizmetler sağlarlar. Böylelikle kullanıcılarına birçok karar verme sürecinde yardımcı olmak için öneriler üretir. Tavsiye sistemlerinin yardımıyla, kullanıcıların film, kitap, müzik, yemek, otel ve restoran gibi uygun ürün ve hizmetlere erişme olasılıkları daha yüksekken ürün ve hizmet sağlayıcıları da maddi kazanç elde ederler.

Öneri sistemlerinde kullanıcılar hakkında bilgi toplanması iki farklı yöntem ile yapılmaktadır. Bunlara açık ve kapalı bildirimler denir. Açık bildirimler bir kullanıcının bir ürün, film ya da müzik olsun onun hakkındaki tercihlerinin yer aldığı (yorum, puanlama, vb) bilgilerden oluşmaktadır. Kapalı bildirim ise kullanıcıların herhangi bir sayfa ya da uygulama üzerindeki geçirdikleri zaman süresince buldukları davranış veya yaptıkları işlemler ile dolaylı olarak elde edilen verilerden oluşmaktadır. Her iki yöntem ile elde edilen veriler kullanılarak öneri sistemleri tarafından kullanıcılara tavsiyede bulunulur.

Öneri sistemleri ortaya çıktığı ilk zamanlarda çok basit yapılar için kullanılırken günümüzde çok daha büyük veriler ve daha karmaşık yapıların çözümünde kullanılmaktadır.

Tavsiye yaklaşımına dayalı olarak, öneri sistemleri üç ana kategoride sınıflandırılır (Adomavicius & Tuzhilin, 2005). İşbirlikçi Filtreleme öneri sistemleri, benzer zevklere sahip diğer kullanıcıların eğilimlerine göre kullanıcılarına öneriler üretir. İçerik tabanlı öneri sistemleri, öğelerin tanımlayıcı karakter özelliklerini kullanarak, kullanıcının geçmişte beğendikleriyle yeni öğelerin benzerliklerine dayalı olarak öneriler üretir. Hibrit tavsiye sistemleri, işbirlikçi filtreleme, içerik tabanlı filtreleme ve bu ikisinin farklı şekillerde bir araya getirildikleri hibrit filtrelemelerdir.

3.2.1. İşbirlikçi Filtreleme

İşbirlikçi filtreleme yöntemi öneri sistemlerinde en yaygın kullanılan yöntemlerden birisidir. Bu teknikte aktif kullanıcı için tahminler üretilirken aktif kullanıcı ile benzer davranış eğilimi gösteren diğer kullanıcıların tercihleri göz önünde bulundurulur.

“Geçmişte benzer tercih eğilimi gösteren kişiler gelecekte de benzer tercih eğilimi içerisindeyler.” varsayımına dayanmaktadır. İşbirlikçi filtreleme yöntemini kullanan sistemler ürünlerin içerik özellikleri yerine kullanıcıların oy bazlı profilleri tahmin üretme işleminin temelini oluşturur. Bu sistemin temel dezavantajı olarak hakkında değerlendirme bulunmayan ürünleri kullanıcılarına önermemeleri söylenebilir.

İşbirlikçi filtreleme sistemleri model tabanlı ve hafıza tabanlı olmak üzere ikiye ayrılmaktadır (Fathan et al., 2018). Hafıza tabanlı yaklaşımlar tahminler üretirken tüm kullanıcı-ürün matrisi üzerinde işlem yaparlar. Hafıza tabanlı yaklaşımlar ise kullanıcı tabanlı ve ürün tabanlı işbirlikçi filtreleme olarak iki başlıkta değerlendirilir. Kullanıcı tabanlı işbirlikçi filtrelemede benzerlik hesaplaması matrisin satırları yani kullanıcılar arasında yapılırken, ürün tabanlı işbirlikçi filtrelemede matrisin sütunları yani ürünler arasında yapılır (Sarwar et al., 2001). Model tabanlı yaklaşımlarda ise çeşitli veri madenciliği ve makine öğrenmesi teknikleri ile kullanıcı-ürün matrisi üzerinde çevrimdışı bir model oluşturulur. Çevrimiçi sistemlerde daha önceden eğitilen bu model kullanılarak tahminler üretilir (Rokach et al., 2001). Model tabanlı filtreleme yaklaşımı, geçmiş derecelendirmeye dayalı olarak derecelendirme yapmamış kullanıcıların verecekleri dereceleri tahmin etmektedir. Hafıza tabanlı filtreleme yaklaşımı, öğelerin geçmiş davranışlarından ve kullanıcılardan derecelendirmeye yönelik benzerliklerini keşfederek çalışmaktadır.

İşbirlikçi Filtreleme yönteminde kullanıcıların veya ürünlerin birbirlerine benzerliklerini hesaplarken kullanılabilir 4 farklı yöntem bulunmaktadır. Bu yöntemler Pearson Benzerliği, Serman Benzerliği, Kosinüs Benzerliği ve Ayarlanabilir Vektör Benzerliği'dir. Bununla birlikte kullanılan başlıca makine öğrenmesi algoritmaları şunlardır; Birliktelik kuralları, Kümeleme, Karar Ağaçları, Boyutsallık Azaltma Tekniği ve Yapay Sinir Ağlarıdır. İşbirlikçi filtreleme yöntemi kullanan öneri sistemleri metod olarak kullanıcı bazlı ve öğe bazlı olmak üzere ikiye ayrılırlar.

Kullanıcı bazlı filtreleme (User-to-user filtering): Bu yöntemde benzer kullanıcılar bulunarak aynı sınıf içerisinde toplanmaya çalışılır ve bu benzer kullanıcılar beğendikleri ürünler kullanılarak ürün önerisinde bulunulur. En etkili yöntemlerin başında gelir ancak zaman ve kaynak açısından zayıftır. Bu sebeple büyük veriler üzerinde bu yöntemi kullanmak zordur. Tahmin aşamasında Kosinüs benzerliğini ya da Pearson

Korelasyonunu kullanırlar. Bu işlem 3 aşamada gerçekleşir. İlk olarak aktif kullanıcıya benzeyen kullanıcıların bulunması ve bu kullanıcıların beğendiği ürünleri bularak aktif kullanıcıya bunların belirlenen sıralama ile sunulmasıdır. Bu yöntem aynı zamanda Hafıza tabanlı Filtreleme yada k en yakın komşu yöntemi olarak adlandırılmaktadır.

Öge bazlı filtreleme (Item-to-item filtering): Önceki algoritmaya çok benzerdir fakat kullanıcılar arasında benzerlik bulmak yerine ürün benzerliğine odaklanır. Bu algoritma ile birlikte herhangi bir ögeyi satın alan kullanıcıya, benzer öğeleri kolaylıkla önerebiliriz. Kullanıcı bazlı filtremeye kıyasla daha az kaynak ve zaman gerektirmektedir. Öge-öge filtreleme algoritması kullanıcı-öge puanlarından elde edilen ürünlerin benzerliklerini analiz eder (Barkan et al., 2016).

Öge bazlı bir model geliştirmek için kullanılabilecek 2 yöntem vardır. Bunlar olasılık yaklaşımı veya derecelendirme tahminidir. Modelleme süreci, sınıflandırma, kümeleme ve kural tabanlı yaklaşım gibi makine öğrenmesi teknikleri ile gerçekleştirilir (Aditya et al., 2016). Literatürde yapılan çalışmalarda öge bazlı filtreleme yönteminin kullanıcı bazlı filtremeye göre daha başarılı tahminlerde bulunduğu belirtilmiştir. Bu yöntem model tabanlı filtreleme yöntemi olarak da adlandırılmaktadır.

3.2.1.1. Benzerlik Yöntemleri

İşbirlikçi Filtreleme yönteminde benzer kullanıcıları bulmak için kullanıcılar arasındaki benzerlik hesaplanır. Benzerlikler hesaplanırken farklı yöntemler kullanılmaktadır. Bu yöntemlerin başında Jaccard benzerliği, Kosinüs benzerliği, Pearson Korelasyon benzerliği gibi yöntemler gelir. Kullanıcı-Ürün matrisi üzerinde oluşturulmuş kullanıcı vektörleri sayesinde kullanıcılar arasındaki, ürün vektörleri sayesinde de ürünler arasındaki benzerlik hesaplamaları yapılmaktadır (Koren, 2009).

a. Kosinüs Benzerliği

Kosinüs benzerliğinde her bir kullanıcı daha önceki değerlendirmelerinin bir vektörü olarak düşünülür. İki vektör arasındaki benzerlik, iki vektörün arasındaki açının kosinüs değerine bakılarak ölçülür. Kosinüs benzerlik denklemi iki vektör x ve y arasındaki ölçümü şu şekilde tanımlanmaktadır.

$$\cos(\theta) = \frac{x \cdot y^*}{\|x\| \|y\|} \quad (1)$$

Kosinüs benzerliği ile elde edilen sonuç her zaman -1 ile +1 arasında yer almaktadır. Ürün veya kullanıcı, vektör olarak tanımlanırsa vektörler arası korelasyondan yola çıkarak vektörler arasındaki açının kosinüs değeri benzerlik ölçütü olarak kabul edilir.

Vektör boyutundan etkilenmemesi, kosinüs benzerliğinin güçlü bir özelliğidir. Vektör Benzerliği daha çok Nesne Tabanlı Filtreleme Yöntemleri'nde yaygın olarak kullanılmaktadır. Literatürde yapılan çalışmalar incelendiğinde Nesne Tabanlı Filtreleme Yöntemleri'nde daha iyi sonuçlar verdiği gözlemlenmektedir.

b. Jaccard benzerliği

Jaccard benzerliği, iki nesne arasındaki benzerliği hesaplamak için kullanılan yaygın bir yakınlık ölçümüdür. Bu benzerlikte her iki kümedeki gözlem sayısı her iki kümedeki eleman sayısına bölünerek hesaplanır. Başka bir ifade ile Jaccard benzerliği, iki kümenin birleşiminin boyutuna bölünen kesişimin boyutu olarak hesaplanabilir. Denklem 2 ile matematiksel formülü gösterilmiştir.

$$j(x, y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (2)$$

Bu denklem sonucunda, asimetric ikili özellikler arasındaki benzerlik ölçümü elde edilebilmektedir. Bir nesnenin özellikleri ikili bir formatta incelenirken karar vericinin kolaylıkla sonuca erişmesinde uygulanabilecek bir uzaklık ölçüm mekanizması sunmaktadır.

c. Pearson korelasyonu

Pearson Benzerliği, İşbirlikçi Filtreleme Yöntemi'nde en sık olarak kullanılan benzerlik bağıntısıdır. Bu benzerlikte iki kullanıcının arasındaki benzerliğini hesaplamak için nesnelere vermiş oldukları puanları karşılaştırır. Benzerlik değeri -1 ile +1 aralığında değişkenlik gösterir. Benzerlik değerinin yüksekliği de iki kullanıcının birbirlerine yakın olduğunu ifade eder. Eğer benzerlik değeri sıfır ise bu iki kullanıcı ya da nesnenin arasında hiçbir benzerlik olmadığı anlaşılır.

$$\rho(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

Denklem 3'te verilen formülde x ve y kullanıcıları arasındaki benzerliğin için x_i ; x kullanıcısının i nesnesine vermiş olduğu puanı, y_i ; y kullanıcısının i nesnesine vermiş olduğu puanı, n değeri x ve y kullanıcıların ortak oylamış oldukları nesnelerin sayısını ve sırasıyla \bar{x} , \bar{y} değerleri ise x ve y kullanıcılarının oylamış oldukları nesnelerin ortalama puan değerini gösterir.

Kimi kullanıcılar ise sürekli düşük değerlendirme yapma eğilimindedirler. Pearson korelasyonu benzerliği bu tür olumsuzlukları en aza indirebilmek için kullanıcıların kendi değerlendirme ortalamalarını alıp bunları hesaba katarak daha doğru sonuçlar elde etmeyi amaçlamaktadır.

3.2.1.2. Komşu Seçimi

Benzerlik hesaplaması belirlenen yöntem ile yapıldıktan sonraki aşama komşuların seçimidir. Benzerlik oranlarına göre belli sayıda veya oranda kullanıcı seçilmektedir. Bu işleme komşuluk seçimi denilmektedir. Komşuluk seçiminde kullanılan iki yöntem vardır. Birinci yöntem komşu seçerken öğeye olan benzerlik oranı belli bir eşik değerin üstünde olanların komşu seçilmesi (Bağıntı Eşiği yöntemi), ikinci yöntem ise öğeye en yakın k adet komşu (K En Yakın Komşu Yöntemi) seçilmesidir (Gong, 2009).

Bağıntı Eşiği Yöntemi'nde bir sınır değeri belirlenir ve sadece bu sınır değerinden daha yüksek benzerliğe sahip kullanıcılar dikkate alınarak hesaplamalar yapılır (Shardanand & Maes, 1995). Bağıntı Eşiği Yöntemi, benzerlikleri düşük olan kullanıcıları eleyerek gürültü değerini en aza indirmeyi hedeflemektedir. Ancak eşik değerinin çok yüksek seçilmesi durumunda yeterli sayıda kullanıcı işleme alınamayacağından doğru sonuçlar elde edilemeyecektir.

K En Yakın Komşu Yöntemi'nde ise kullanıcılar arasındaki uzaklıkların hesaplanması gerekmektedir. Uzaklıklar hesaplandıktan sonra en küçük uzaklığa sahip k adet komşu seçilir. Uzaklıkların hesaplanması aşamasında herhangi bir uzaklık hesaplama yöntemi kullanılabilir. En yaygın olarak kullanılan uzaklık hesaplama yöntemi Öklid Uzaklığıdır.

3.2.1.3. Tahmin Hesaplama

İşbirlikçi filtrelemede komşular seçildikten sonra son aşamada artık aktif kullanıcıya ürün önerme yapılmaktadır. Bu aşamada aktif kullanıcının değerlendirmesi olmayan ürünler için yapabileceği değerlendirmeler tahmin edilmeye çalışılır. Literatürde birçok tahmin hesaplama yöntemi bulunmakla beraber, en yaygın olarak kullanılan dört adet tahmin hesaplama yöntemi Basit Ortalama, Ağırlıklandırılmış Ortalama, Ayarlanabilir Ağırlıklandırılmış Ortalama ve Standart Sapmalı Ayarlanabilir Ağırlıklandırılmış Ortalama yöntemleridir (Milli, 2013).

Ayarlanabilir Ağırlıklandırılmış Ortalama yönteminde bulunan benzerlik sonuçlarıyla birlikte ağırlık vektörleri oluşturulur. Ağırlık vektörleri oluşturulduktan sonra kullanıcılara tercih edebilecekleri ürünleri önermek için benzerlik puanlarının kullanıldığı denklem 4 ile gösterilen, ayarlanabilir ağırlıklandırılmış ortalama sayesinde, önerilerde bulunma işlemi gerçekleştirilebilir.

$$R_{x,i} = \bar{R}_x + \frac{\sum_{u \in U} (R_{y,i} - \bar{R}_y) \cdot sim(x, y)}{\sum_{y \in U} sim(x, y)} \quad (4)$$

Denklemden $R_{x,i}$ x kullanıcısının i öğesine verdiği değer tahminini \bar{R}_x ve \bar{R}_y sırasıyla x ve y kullanıcısının ortalamasını, $sim(x, y)$ x kullanıcısı ile y kullanıcısı arasındaki benzerlik oranını; U kümesi ise x kullanıcısına yakın olan komşuları ifade etmektedir.

3.2.2. İçerik Tabanlı Öneri Sistemleri

İçerik tabanlı öneri sistemlerinde kullanıcıların geçmişte tercih etmiş oldukları ürünlere içerik olarak benzerlik gösteren diğer ürünler önerilir. İçerik tabanlı öneri sistemleri, öğelerin tanımlayıcı niteliklerine ve kullanıcı profillerine dayalı öneriler üretir (Van Meteren & Van Someren, 2000). Burada kullanıcıların ve öğelerin özellikleri önemlidir. İçerik tabanlı öneri sistemlerinin amacı, her kullanıcı ve her öğe için bir profil oluşturmaktır. Örneğin, film önerisi yapılırken, filmler arasındaki benzerliği bulmak için

yönetmen, aktörler, film uzunluğu, tür vb. özellikler kullanılır. İçerik tabanlı filtrelemenin temel mantığı, kullanıcı bir ürünü beğendiyse bu ürüne benzer başka bir ürünü de beğeneceği düşüncesi yatmaktadır. Bu sistemin avantajı ise özellikle sisteme yeni eklenen ve hakkında hiç değerlendirme bulunmayan ürünleri içerik bilgilerini kullanarak kullanıcılara önermede oldukça etkilidir. Ancak bu yöntem içerik olarak kullanıcılarının geçmişte tercih ettikleri ürünlere benzeyen önerilerde bulduklarından kullanıcılarını şaşırtma konusunda zayıf kalırlar (Barkan et al., 2016).

İçerik tabanlı öneri siteminde öneriler bir kullanıcıya özgü olduğundan dolayı, içerik tabanlı model diğer kullanıcılar hakkında herhangi bir veriye ihtiyaç duymaz. Bu, çok sayıda kullanıcıyı ölçeklendirmeyi oldukça kolaylaştırır. Bununla beraber sistemde bulunan az sayıda kullanıcı ile bile güvenilir sonuçlar ortaya çıkabilir. Yeni ürünler için de puanlama, satın alma olmadan da öneriler sunulabilir, bu şekilde soğuk başlangıç problemine çözüm olabilir. Model, bir kullanıcının belirli ilgi alanlarını yakalayabilir ve çok az sayıda kullanıcının ilgilendiği uygun öğeleri önerebilir.

Sistemin negatif yönleri ise ürüne ait yeterli bilgi yok ise, öneri sistemi geçerliliğini yitirebilir. Tavsiyeler yalnızca kullanıcının belirttiği yani sisteme söylediği ilgi alanlarıyla sınırlı kalır. Bu sebeple sistemin yapılabileceği öneriler sınırlı kalır. Ayrıca kullanıcı için sağlam bir profil oluşturmak için yeterli bilgi olmadığı zamanlarda, tavsiye sistemi doğru şekilde çalışmaz.

3.2.3. Hibrit Sistemler

Hibrit öneri sistemleri İşbirlikçi filtreleme sistemleriyle İçerik tabanlı filtreleme sistemlerinin avantajlı olan yönlerini birleştirerek geliştirilmiş öneri sistemleridir. Hem İşbirlikçi filtreleme sistemlerin hem de içerik tabanlı sistemlerin kendine özgü güçlü ve zayıf yönleri bulunmaktadır. Hibrit öneri sistemleri ise, her iki sisteminde artılarından yararlanarak her bir yöntemin dezavantajından kaçınmak için işbirlikçi filtreleme ve içerik tabanlı yöntemleri birleştirir. Bu birleştirme, bir işbirlikçi filtreleme öneri sisteminde herhangi bir kullanıcı değerlendirmesi olmadan yeni bir öğenin içerik tabanlı tanımlayıcı bilgilerini kullanarak öneri üretmesi şeklinde olacaktır (Tran & Cohen, 2000).

3.3. Derin Öğrenme

Derin öğrenme, tipik olarak yapay sinir ağlarını kullanarak birkaç temsil katmanını öğrenmeye dayanan bir makine öğrenimi alanıdır. Günümüzde derin öğrenme yaklaşımları, bilgisayarla görme, doğal dil işleme ve konuşma tanıma gibi birçok soruna yeni çözümler üretir (Deng & Yu, 2014). Sinir ağları ve derin modellerin arkasındaki bilim 50 yılı aşkın süredir var olmasına rağmen, derin öğrenme teknikleri son on yılda çokça kullanılmaya başladı. Makine öğrenmesi tekniği olarak derin öğrenmeyi teşvik eden ana faktörler büyük verilere sahip olunması ve hesaplama gücündeki gereken işlem gücünün sağlanmasıdır.

Derin öğrenme teknikleri çok katmanlı yapay sinir ağlarını kullanarak öğrenmeye dayanır. Derin öğrenme teknikleri ile heterojen, karmaşık ve büyük boyutlu veriden lineer olmayan, kompleks ve gizli özellikler çıkarılabilir. Son zamanlarda büyük veri depolama olanaklarının artmasıyla ve hızlı işlem yapabilen güçlü işlem birimlerinin tasarlanmasıyla birlikte derin öğrenme teknikleri de popüler hale gelmeye başlamış ve doğal dil işleme, örüntü tanıma, imge işleme gibi pek çok alanda sıklıkla kullanılmıştır (LeCun et al., 2015).

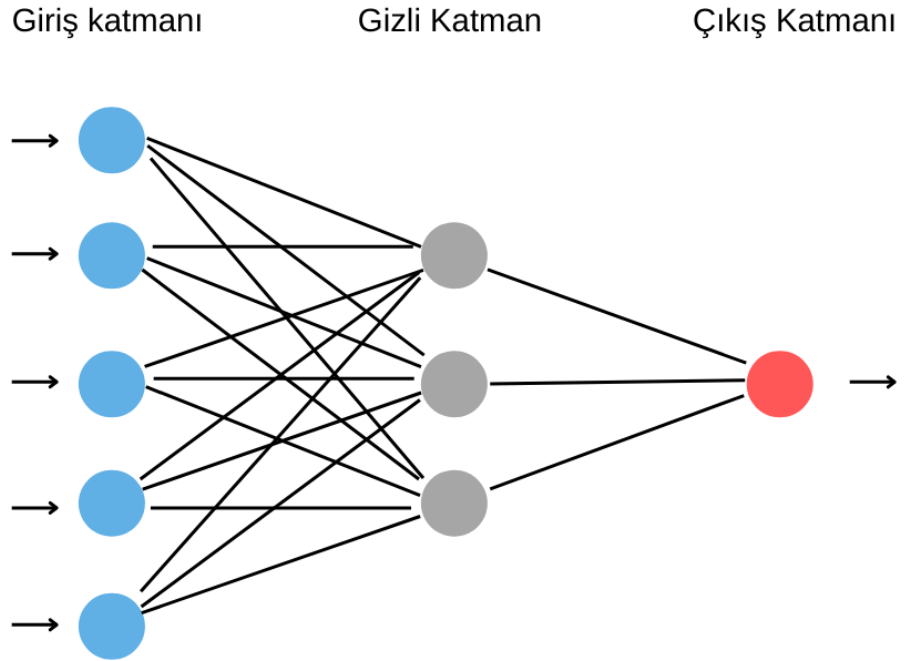
Derin öğrenme tekniklerinin bilgisayar bilimlerindeki bu başarısı araştırmacıları bu teknikleri öneri sistemlerinde de kullanmaya yönlendirmiştir. Öneri sistemlerinde yaygın olarak kullanılan derin öğrenme modellerini arasında özkodlayıcılar, konvolüsyonel sinir ağları (KSA), kısıtlı Boltzmann makineleri (KBM) ve yinelemeli sinir ağları (YSA) gösterilebilir.

Derin öğrenme teknikleri, ham veriden gizli ve karmaşık özellikleri çıkarabilme ve farklı kaynaklardan bilgileri bir araya getirebilme yetilerinden dolayı geleneksel İşbirlikçi Filtreleme sistemlerinde üretilen tahminlerin doğruluğunu iyileştirmek için sıklıkla kullanılmıştır. Öneri sistemlerinde, derin öğrenme yaklaşımları, kullanıcıların ve ürünlerin gizli ve kompleks özelliklerini elde etmek, hem kullanıcı hem de ürün tabanlı öneri sistemlerinin birleşik modelini oluşturmak ve içerik, yorum gibi ekstra verilerden özellik çıkarımı yaparak bu bilgileri öneri sistemlerine entegre etmek için yaygın olarak kullanılırlar (Salakhutdinov et al., 2007).

3.3.1. Yapay Sinir Ağları

Yapay Sinir Ağı, biyolojik sinir ağlarının yapısı örnek alınıp işlevlerini taklit ederek çalışan matematiksel bir modeldir. Bütün yapay sinir ağlarının temel yapısı yapay bir nörondan oluşur, bu da basit bir matematiksel modeldir.

Yapay sinir ağlarının biyolojik sinir ağlarının sinir hücrelerine benzer şekilde yapay sinir hücreleri vardır. Yapay sinir hücrelerinin proses elamanları olarak adlandırılan 5 temel elemanı bulunmaktadır. Bunlar; girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılardır (Öztemel, 2006).



Şekil 2. Yapay Sinir Ağlarının Yapısı

Sisteme verilen bir veri setinden öğrenebilme yeteneği Yapay sinir ağlarının en önemli özelliğidir. Burada öğrenme işlemi girişler ile başlar. Dış dünyadan veya bir önceki katmandan alınan bilgiler giriş olarak yapay sinir hücrelerine gönderilir. Yapay sinir ağlarında bilgi, ağdaki sinirlerin bağlantılarının ağırlıklarında tutulur. Bu nedenle ağırlıkların nasıl belirleneceği önemlidir. Bilgi tüm ağda saklandığı için bir düğümün sahip olduğu ağırlık değeri tek başına bir şey ifade etmez. Tüm ağdaki ağırlıklar optimal değerler almalıdır. Bu ağırlıklara ulaşılabilmesi için yapılan işleme “ağın eğitilmesi”

denir. Ağırlıklar, girişlerin yapay sinir hücreleri üzerindeki etkisini belirleyen ve öğrenme işleminin gerçekleşmesini sağlayan katsayılardır.

$$y = f\left(\sum_{i=1}^n w_i x_i\right) \quad (5)$$

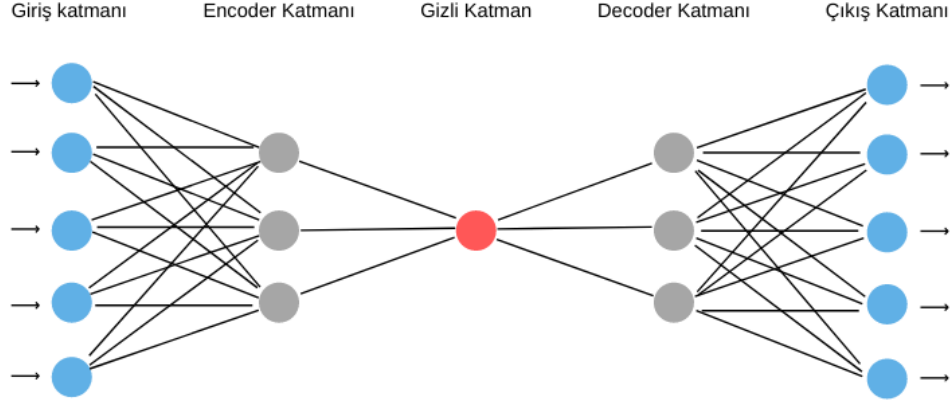
Yapay sinir hücresine gelen net girişin hesaplanması toplama işlemi olarak adlandırılır. Denklem 5 te verilen fonksiyonda yapay sinir hücresine ait toplama işleminin sonucu, x_i yapay sinir hücresine ait girişler, w_i yapay sinir hücresine ait giriş ağırlığını ifade etmektedir.

Toplam işleminden sonra, elde edilen değere “aktivasyon fonksiyonu” olarak adlandırılan bir fonksiyon uygulamaktadır. Aktivasyon fonksiyonu genellikle daha önce hesaplanan toplam değeri bir fonksiyon ile 0 ile 1 arasında bir sayıya çevirmek için kullanılır. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. Yapay sinir ağlarının bir özelliği olan “doğrusal olmama” aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir.

3.3.2. Öz Kodlayıcı Ağlar

Öz kodlayıcı yapay sinir ağı modellerinden biri olup denetimsiz bir makine öğrenmesi yöntemidir. Giriş değerleri ile çıkışta elde edilen değerlerin birbirine yakın olması amaçlanır. Klasik bir yapay sinir ağı modeli girdi katmanı, gizli katman ve çıktı katmanı olmak üzere üç katmandan meydana gelmektedir. Girdi katmanındaki nöron sayısını veri setindeki özellik sayısı kadardır. Çıktı katmanındaki nöron sayısını ise elde edilmesi istenen sınıf sayısı kadardır. Gizli katman sayısı ve bu katmandaki nöron sayıları ise sabit değildir ve genel olarak deneme yolu ile bulunmaktadır. Klasik yapay sinir ağları çok karmaşık olmayan veri setlerinde oldukça iyi sonuçlar vermektedir ancak daha karmaşık veri setleri için yeterli başarı oranı elde edilmemektedir. Öz kodlayıcılar ise daha karmaşık veri setlerinde başarı oranını artırmak için benzer bir ağ yapısını iyileştirerek kullanmıştır. Öz kodlayıcı modelinde girdi katmanındaki nöron sayısı genellikle gizli katmandaki nöron sayısından daha fazladır. Bu modeli ileri beslemeli yapay sinir ağlarından ayıran en önemli ikinci özellik ise giriş veri setiyle çıkış veri setinin aynı

olması dolayısıyla çıktı katmanındaki nöron sayısının girdi katmanındaki nöron sayısına eşit olmasıdır (Ng, 2011).



Şekil 3. Öz Kodlayıcı Ağlarının Temel Yapısı

Öz kodlayıcı sınıflandırma işlemi yapmaz. Temel amacı n boyutlu bir özellik vektörünü daha küçük bir boyutlu vektöre en az kayıp ile düşürmektir. Öz kodlayıcı yapısı temel olarak çok katmanlı perceptron mimarisine benzemektedir. Öz kodlayıcılar kendi içerisinde iki farklı yapı bulundurlar. Bunlar kodlayıcı (encoder) ve çözücü (decoder) diye adlandırılır. Kodlayıcı çok boyutlu veriyi az boyuta düşürmemizi sağlar. Çözücü ise sıkıştırılmış veriyi boyutunu artırarak girişi yeniden elde etmeye çalışır. Matematiksel olarak ifade etmek gerekirse N boyutlu bir özellik vektörünü daha küçük bir boyutlu vektöre indirgeme aşamasında girdi katmanında tüm özellikler okunur. Daha sonra bu bilgiler Denklem 6'de gösterildiği gibi gizli katmana aktarılır. Denklem 6'da x_j girdi katmanındaki j 'inci nöronun değerini, y_i gizli katmandaki i 'inci nörona aktarılan değeri, n girdi katmanındaki nöron sayısını, w_{ji} girdi katmanındaki j 'inci nörondan gizli katmandaki i 'inci nörona giden ağırlığı, f ise aktivasyon fonksiyonunu temsil etmektedir.

$$y_i = f\left(\sum_{j=1}^n w_{ij} x_j\right) \quad (6)$$

Elde edilen değerler Denklem 7'de gösterildiği gibi çıktı katmanına aktarılarak, son değerler hesaplanır. Denklem 7'de x_j' çıktı katmanındaki j 'inci nöronu, y_i gizli

katmanındaki i 'inci nöronu, w_{ij} gizli katmanındaki i 'inci nöronun çıktısı katmanındaki j 'inci nörona giden ağırlığı, m gizli katmandaki nöron sayısını f ise aktivasyon fonksiyonunu temsil etmektedir.

$$x'_j = f\left(\sum_{i=1}^m w_{ij} y_i\right) \quad (7)$$

Özkodlayıcı modelinde temel amaç, ilk iki aşama sonrasında üretilen x'_j değerinin, girdi katmanındaki x_j değerine yakın bir değer elde etmektir. Bu iki değer birbirine yakın çıkması için ağırlıklar geri yayılım algoritması yardımıyla hesaplanarak sürekli güncellenir. Geri yayılım algoritması denklem 8'te verildiği gibi iki değer arasındaki farkın karesini minimize etmektedir.

$$\min \sum_{i=1}^n (x'_j - x_j)^2 \quad (8)$$

Diğer derin öğrenme mimarilerinde olduğu gibi özkodlayıcı modelinde de karşılaşılan en önemli sorunlardan birisi aşırı öğrenmesidir. Aşırı öğrenme durumu, bir modelin eğitim seti için büyük başarı oranı vermesine rağmen eğitim verileri dışındaki yeni bir veri için düşük bir başarı oranı elde etmesidir. Bu problemin önüne geçmek için çeşitli yöntemler bulunmaktadır. Bu yöntemlerden ilki kullanılan döngü sayısının sınırlanmasıdır. Öğrenme aşamasında döngü sayısı arttıkça ağırlıklar eğitim veri setini ezberleyecek şekilde sıfır hatayla öğrenmektedir. Giriş veri setinde gürültü olması durumunda gürültünün de gerçek veri gibi modele dâhil edilmesine neden olacaktır. Bir diğer yöntem ise, eğitim verisinin bir kısmını doğrulama verisi olarak kullanıp eğitim verilerindeki hata oranı azalırken benzer şekilde doğrulama verisindeki hata oranının azalıp azalmadığını kontrol ederek eğitimin erken sonlandırılmasını sağlamaktır.

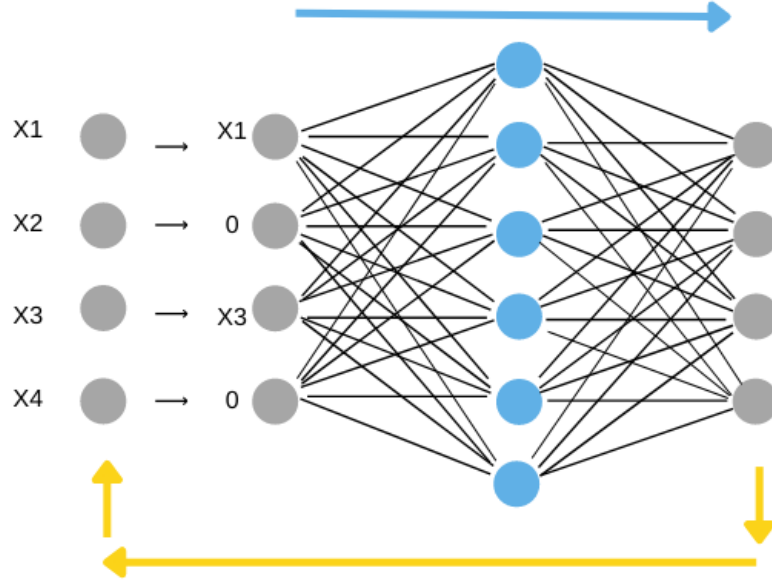
3.3.3. Gürültü Giderici Öz Kodlayıcılar

Gürültü Giderici Öz Kodlayıcılar, gürültülü bir girdiden güçlü bir şekilde elde edilebilen ve orijinal girdiyi yeniden inşa etmek için faydalı olacak olan bir yöntemdir. Öz

kodlacılara giriş olarak gürültülü veri verilerek gürültü giderme için eğitilebilir. Veri setine yapay olarak gürültü eklersek ve hatayı hesaplarken gürültüsüz veri ile farkını kullanırsak model sadece girişi tekrar üretmeyi değil, gürültüsünü de gidermeyi öğrenecektir. Gürültü giderici özkodlayıcıların, bildiğimiz özkodlayıcılardan tek farkı girdi olarak verdiğimiz veride rastgele eklenmiş gürültülerin de varolmasıdır. Yani model, girdi verilerinden çıktı verilerini oluştururken yaptığı yeniden inşa işleminde girdinin kendisini görmek yerine gürültü katılmış veriyi görmektedir.

Bir özkodlayıcı için eğitim sırasında her bir x girdisi için tüm gizli katmanlarda ki aktivasyonları hesaplamak üzere ileri beslemeli bir geçiş yapılmaktadır. Daha sonra, çıktı katmanında, çıktı ve girdi x arasındaki fark hata karesi kullanılarak ölçülür ve hata büyük olduğunda, hatanın ağ boyunca yeniden yayılması ve geri yayılım algoritması ile x' çıktısını elde etmek için ağırlık güncellemeleri yapılarak azaltılır. Bu yöntemde ise eğitimin her döngüsünde kod çözücü tarafından çıkarılan gürültülü veri ile orijinal veri arasındaki bir kaybı hesaplayacak ve ayrıca yeniden yapılandırılmış görüntü ile orijinal gürültüsüz görüntü arasındaki bu kaybı veya farkı en aza indirmeye çalışacaktır (Sharma, 2018).

Ancak, gizli katmandaki düğümlerin sayısı girdi katmanındakilerden daha fazla olduğunda, sinir ağı kimlik fonksiyonu olarak adlandırılır. Bu da girdinin çıktıya direkt kopyalandığı, eşit olduğu anlamına gelir. Gürültü giderici özkodlayıcılar, girdi verilerinin bazılarını rastgele olarak sıfıra getirerek yani buradaki verileri bozarak bu sorunu çözer. Genel olarak, sıfıra ayarlanan girdi düğümlerinin yüzdesi yaklaşık olarak %50'dir. Bir gürültü giderici özkodlayıcının temel mimarisi Şekil 4'te gösterilmektedir.



Şekil 4. Gürültü Giderici, Özkodlayıcı Ağlarının Temel Yapısı

Burada dikkat edilmesi gereken bir nokta, kayıp fonksiyonunu hesaplariken, çıktı değerlerini bozduğumuz girişle değil orijinal girişle karşılaştırmak önemlidir. Bu şekilde, öznelik özellikleri yerine kimlik fonksiyonunu öğrenme riski ortadan kaldırılır.

Gürültü giderici özkodlayıcılar, verilerden daha sağlam bir eşleme öğrenme motivasyonu ile verilerin bozuk bir versiyonundan gelen girişi yeniden yapılandırır.

Gürültü Giderici Özkodlayıcı, ağın kimlik işlevini öğrenmesini önlemek için özkodlayıcı üzerinde yapılan bir değişikliktir. Spesifik olarak, eğer özkodlayıcı çok büyükse, o zaman sadece verileri öğrenebilir, böylece çıktı girdiye eşittir ve herhangi bir yararlı temsil öğrenme veya boyutsallık azaltma gerçekleştirmez. Gürültüsüz özkodlayıcılar, giriş verilerini bilerek bozarak, gürültü ekleyerek veya giriş değerlerinin bazılarını maskeleyerek bu sorunu çözer.

4. DENEYSEL SONUÇLAR

4.1. Deney Metodolojisi ve Değerlendirme Metrikleri

Tezde oluşturulan modeller ve test çalışması yapılan optimizasyon Google Colaboratory veya kısa adıyla Colab bulut sistemi kullanılmıştır. Deneysel analizler Python yazılım dilinde Pandas ve Keras kütüphanesi kullanılarak gerçekleştirilmiştir (Google Colab, 2022). Keras kütüphanesi araştırmacılara makine öğrenmesi uygulamalarını kurabilecekleri ortam sağlayan açık kaynak kod platformudur.

Ham veri dosyası, pasdas kütüphanesi kullanılarak içe aktarılır ve başlıkları olan bir CSV'ye dönüştülür. Sıfır tabanlı dizin için tüm kullanıcı ve film kimliğinden 1 çıkarılacaktır. Derecelendirme verileri için ön işleme yapılır, benzer ön işleme, kullanıcı verilerine ve film verilerine de uygulanır. Elde edilen csv dosyalarından derecelendirme dosyasının örneği Tablo 4'te verilmiştir.

Tablo 4. Oluşturulan Puan Veri Seti

KULLANICI ID	FILM ID	DERECELENDİRME	ZAMAN DAMGASI	KULLANICI YENİ ID	FILM YENİ ID
1	1193	5	978300760	0	1192
1	661	3	978302109	0	660
1	914	3	978301968	0	913
1	3408	4	978300275	0	3407
1	2355	5	978824291	0	2354

Oluşturulan modelde giriş değeri 3.952 uzunluğunda bir vektörden oluşmaktadır. Gizli katmanlar ile bu katmanlardaki aktivasyon fonksiyonları yapılan testler ile denenmiştir. Gizli katmanlardaki nöron sayısının artması başarı oranını artırmıştır. Aktivasyon fonksiyonu olarak elu, selu ve relu fonksiyonları denenmiştir. Özkodlayıcının katmanları sırası ile 80, 40, 80 nörondan oluşmaktadır. Çıkış katman sayısı ise film sayısı olan 3.952 uzunluğundadır.

Öneri sistemlerinde yapılan çalışmaların kalitesi doğruluk, hız ve kişiselleştirme seviyesine göre belirlenir. Bu kriterlerden en önemlisi doğruluktur (Milli, 2013).

Doğruluğu ölçmek için literatürde birçok yöntem önerilmiştir. Tez kapsamında bu yöntemlerden Kök Ortalama Kare Hatası yöntemi kullanılmıştır.

4.1.1. Kök Ortalama Kare Hatası

Özkodlayıcılarda modelin performansı genellikle maliyet fonksiyonu olarak Kök Ortalama Kare Hatası (RMSE) ile ölçülür. Denkem 8’de gösterilmiştir.

$$RMSE = \frac{1}{m} \sqrt{\sum_{i=1}^m (r_i - y_i)^2} \quad (8)$$

Burada r_i gerçek derecelendirmeyi, y_i tahmin değerini, m ise toplam sayıyı ifade etmektedir. RMSE yönteminin amacı özkodlayıcının tahmin ettiği değerler ile gerçek değerler arasındaki uzaklığın bulunmasıdır. Denklemden hata farklarının karesi alınıp negatiflikten kurtulduktan sonra aritmetik ortalama işlemi yapılmaktadır. Daha sonra bulunan hata karelerinin ortalaması alınmaktadır. Bu iki değer birbirine yakın çıkması için ağırlıklar geri yayılım algoritması yardımıyla hesaplanarak sürekli güncellenir. Geri yayılım algoritması Denklem 8’de verildiği gibi iki değer arasındaki farkın karesini minimize etmektedir. Sonuç ne kadar az ise modelin başarısı o kadar yüksek demektir.

4.2. Hiper Parametrelerin Özellikleri

Veri seti üzerinde yapılan hazırlık aşaması ve değerlendirme kriterinin belirlenmesinden sonra modeli oluşturmak için veri setine uygun hiper parametreler belirlenir. Derin öğrenme modelleri tasarlanırken problemi iyi temsil edecek, yapılacak deneylere uygun katman sayısı, veri boyutu (Batch size), öğrenme oranı (learning rate), eğitim tur sayısı (Epoch size) gibi hiper parametreler seçilmelidir. Bu modelde kullanılacak değerler Tablo 5’ te belirlenmiş olup iki farklı optimizasyon algoritması kullanılmıştır. Bu amaçla, optimizasyon yöntemi SGD ve Adam, aktivasyon fonksiyonu Selu, Elu ve Relu, nöron sayılarının kombinasyonları kullanılarak 12 farklı model oluşturulmuştur. Oluşturulan model kombinasyonlarının performansları karşılaştırılarak, en iyi model belirlenmiştir. Sonuçlara göre; modellerin performanslarının parametrelere bağlı olarak değişkenlik gösterdiği görülmektedir.

Sonuç olarak derin öğrenme çalışmalarında model oluşturulurken; optimizasyon algoritmalarının, aktivasyon fonksiyonlarının ve nöron sayılarının farklı seçeneklerine göre model performanslarını denemek mümkündür. Ayrıca oluşturulan modelde, optimizasyon yöntemlerinin farklı parametrelerinin kombinasyonlarıyla çalışıldığında, veri setine daha uygun mimari elde edilmektedir.

Tablo 5. Deneysel çalışmada kullanılan parametreler

PARAMETRE	DEĞER
Parti boyutu (batch size)	256
Döngü (epoch) sayısı	500
Öğrenme oranı (learning rate)	0,001
Optimizasyon algoritması	Adam, SGD
Aktivasyon fonksiyonu (activation function)	Selu, elu, relu

Modeli eğitmek ve test etmek için rastgele %90–%10 eğitim ve test verilerine bölünür ve eğitim setinin %10'unu doğrulama için ayrılmıştır. Burada dikkat edilmesi gereken nokta bir kullanıcının değerlendirmelerinin tamamının bir eğitim veya test setine bölünmesi ve yanlılığa neden olmasıdır. Örneğin, A kullanıcısının tüm değerlendirmeleri eğitim setine konursa, test süresi boyunca bu kullanıcı için test verisi olmaz. Bu kullanıcı için test RMSE değeri 0 olacaktır. Diğer taraftan, tüm değerlendirmeler test setine konursa, eğitim süresi boyunca bu kullanıcı için değerlendirme olmaz ve bu kullanıcı için RMSE değerinin daha yüksek olmasına neden olur.

Bu sebeple eğitim ve test verileri her kullanıcının değerlendirmesinin de uygun oranda yer alması sağlanmıştır. Oluşturulan veri setlerinde 395 tane test verisi ve 3556 tane de eğitim verisi olmuştur. Bu işlem sklearn kütüphanesinde `train_test_split` fonksiyonu sayesinde yapılmıştır. Verileri istediğimiz oranda eğitim ve test verisi olarak rastgele bir şekilde dağıtmıştır.

Böldüğümüz verileri yükleyip, python kütüphanelerini ve belirlediğimiz hiper parametreleri içeriye aktardıktan sonra derin öğrenme kütüphanesi tensorflow kullanılarak deneyler yapılmıştır.

4.3. Öneri Modelinin Oluşturulması ve Eğitilmesi

Tezde oluşturulan modeller ve test edilen optimizasyon algoritmaları Python programlama dilinde Keras kullanılarak yazılmıştır. Keras kütüphanesi makine öğrenmesi algoritmalarının uygulanmasını sağlayan açık kaynak kod platformudur.

Öneri sisteminin eğitilmesi için kullanılan işlem adımları aşağıdaki gibidir.

- Eğitim verisi okunur
- Hiper parametreler belirlenir.
- Giriş verilerine gürültü eklenir
- Model oluşturulup eğitim süreci başlar

Her eğitim örneği önce kasıtlı olarak gürültü eklenerek bozular. Ağ, farklı oranlarda gürültüyü işlemek için eğitilmiştir. Bozuk girişler rastgele seçilir ve eğitim için ağa beslenir. Kayıp fonksiyonu, eğitim veri setindeki tüm örneklerin ortalama kare hatası olarak hesaplanır ve ağ bu hatayı mümkün olduğunca azaltmaya çalışır. Eklenecek gürültü, koddaki kodlama kod çözme katmanı işlevlerine gürültü girişi değiştirilerek buna göre seçilebilir. Daha iyi eğitim sonuçları için gürültü seviyeleri de değiştirilmektedir.

Yapılan testlerde ayrıca en uygun optimizasyon algoritmasının hangisi olacağı değerlendirilir. Deneylede adam ve SGD algoritmaları denenerek karşılaştırılması yapılmıştır.

4.4. Deneysel Sonuçlar

Gürültü Giderici Özkodlayıcılar girişi kısmen bozulmuş bir veriyi yeniden yapılandırılmaları için eğitilir ve sonuçlar bozulmamış veriler ile karşılaştırılır. Bu nedenle, gürültü giderici özkodlayıcı bozuk, rastgele seçilen eksik örüntü alt setleri için, bozulmamış değerlerden bozuk değerleri tahmin etmeye çalışmaktadır.

Derin öğrenme tabanlı işbirlikçi filtreleme yöntemi için kullanılan en iyi mimariyi bulmak amacıyla modelde denenen katman, düğüm sayısı ve hiper parametre değerleri ile alınan sonuçları verilmiştir.

Model mimarisinin belirlenmesinde ilk aşama katman ve düğüm sayılarının belirlenmesidir. Özkodlayıcıların yapısal karakteristiği gereği giriş ve çıkış düğüm sayıları aynı olmalıdır. Kodlayıcı katmanın düğüm sayıları [256, 512, 1024] düğümleri

içerisinden her bir düğüm denenerek karar verilecektir Yapılan deneysel çalışmalarda farklı katman ve düğüm sayılarının modele etkisi incelenmiştir. Sonraki aşamada ise hiper parametrelerin seçilmesidir. Belirlenen katman ve düğüm sayıları ile hiper parametre değerleri değiştirilerek modelin en iyi çalıştığı parametreler bulunur.

İlk olarak dört farklı katman modelde kullanılan parametreler Tablo 6’da değerleri ile birlikte verilmiştir.

Tablo 6. Deneysel çalışmada kullanılan parametreler

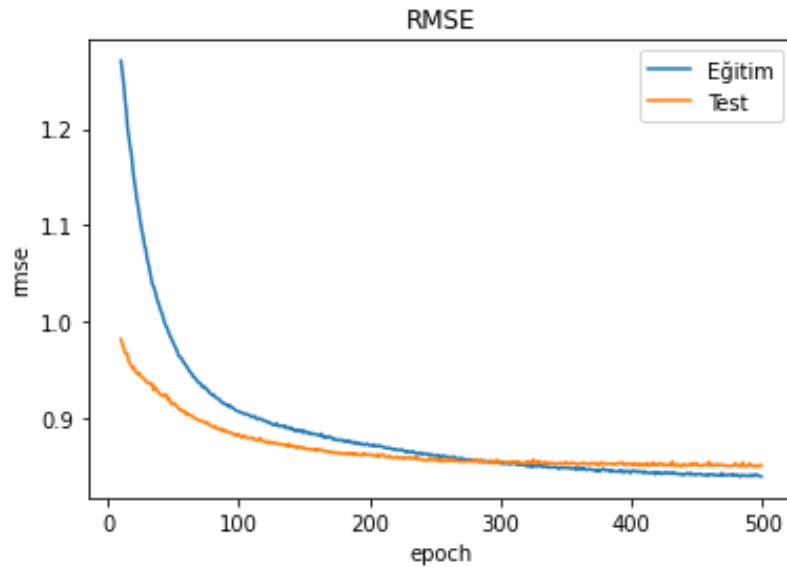
PARAMETRE	DEĞER
Parti boyutu (batch size)	256
Döngü (epoch) sayısı	500
Öğrenme oranı (learning rate)	0,0001
Optimizasyon algoritması	Adam
Aktivasyon fonksiyonu (activation function)	Selu, elu, relu
Noise	0, 0,1 - 0,3 - 0,5

Ağlarda yapılan deneylerde aktivasyon fonksiyonu olarak sırasıyla Selu, elu, relu fonksiyonu kullanılmıştır. Bu yöntemde öncelikle kullanıcılar için gizli katmanları oluşturulmuş ve bunlar bir araya getirilmiştir. Bu aşamada orijinal giriş verisine farklı oranlarda gürültü eklenmiştir. Ardından modele enkoder kısmı ve sonra gelen katmanda düğüm seyreltme uygulanmıştır. Son olarak dekoder katmanı ile veri tekrar oluşturularak çıktı katmanı oluşturulmuştur. Kayıp fonksiyonu için RMSE kayıp yöntemi ve optimizasyon algoritması olarak adam kullanılmıştır. Öğrenme oranı 0,0001 seçilmiştir. Eğitim toplam 500 devirde tamamlanmış her devir sonunda ve test verisi tasarlanan modelde test edilmiştir.

Tablo 7. Deneysel çalışmada elde edilen sonuçlar (Selu)

MODEL	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	0,8588	0,8101
[256, 512, 256], Noise:0,3	0,8567	0,8371
[256, 512, 256], Noise:0,5	0,8654	0,862
[256, 512, 256]	0,8612	0,7921
[512, 256, 512], Noise:0,1	0,8559	0,8138
[512, 256, 512], Noise:0,3	0,8534	0,8392
[512, 256, 512], Noise:0,5	0,861	0,8636
[512, 256, 512]	0,8596	0,7944
[512, 512, 1024, 512, 512], Noise:0,1	0,8838	0,7021
[512, 512, 1024, 512, 512], Noise:0,3	0,8753	0,7647
[512, 512, 1024, 512, 512], Noise:0,5	0,8769	0,8101
[512, 512, 1024, 512, 512]	0,9035	0,6625
[1024, 512, 256, 512, 1024], Noise:0,1	0,8615	0,7729
[1024, 512, 256, 512, 1024], Noise:0,3	0,8579	0,811
[1024, 512, 256, 512, 1024], Noise:0,5	0,8665	0,8437
[1024, 512, 256, 512, 1024]	0,8739	0,7458

Oluşturulan modellerde gürültü değeri artırılarak sonuçlar incelenmiştir ve gürültü olmaksızın derin ağ ile kıyaslanmıştır. Tablo 7’de görüldüğü gibi test hatası 0,8534 puan ile [512, 256, 512] ağı 0,3 gürültü oranında en iyi sonuç alınmıştır.



Şekil 5. Adam Optimizasyon Algoritması (Selu)

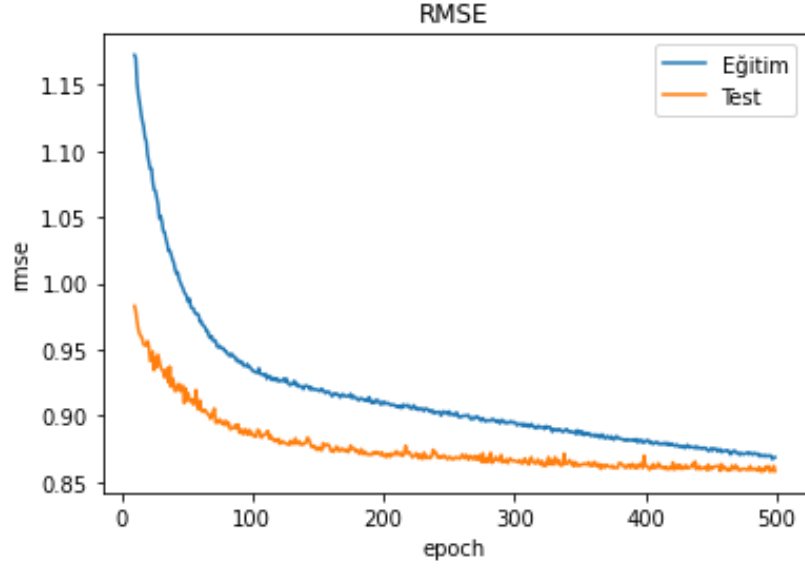
Şekil 5'te [512, 256, 512] modelinde Adam Algoritması, selu fonksiyonu ile 0,3 noise oranı en iyi sonucu vermiştir, eğitim ve test hatası grafikte olarak görülmektedir.

Sonraki test için elu fonksiyonu kullanılarak modele etkisi incelenmiştir. Eğitim toplam 500 devirde tamamlanmış her devir sonunda test verisi tasarlanan modelde test edilmiştir.

Tablo 8. Deneysel çalışmada elde edilen sonuçlar (Elu)

MODEL (Elu)	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	0,8639	0,8753
[256, 512, 256], Noise:0,3	0,8686	0,8922
[256, 512, 256], Noise:0,5	0,8853	0,9069
[256, 512, 256]	0,8624	0,8688
[512, 256, 512], Noise:0,1	0,8608	0,8696
[512, 256, 512], Noise:0,3	0,8672	0,8845
[512, 256, 512], Noise:0,5	0,8839	0,9022
[512, 256, 512]	0,8609	0,8613
[512, 512, 1024, 512, 512], Noise:0,1	0,8727	0,8034
[512, 512, 1024, 512, 512], Noise:0,3	0,8683	0,8281
[512, 512, 1024, 512, 512], Noise:0,5	0,9019	0,8511
[512, 512, 1024, 512, 512]	0,88	0,7829
[1024, 512, 256, 512, 1024], Noise:0,1	0,8637	0,8304
[1024, 512, 256, 512, 1024], Noise:0,3	0,8717	0,847
[1024, 512, 256, 512, 1024], Noise:0,5	0,8843	0,865
[1024, 512, 256, 512, 1024]	0,8692	0,8105

Oluşturulan modellerde gürültü değeri artırılarak sonuçlar incelenmiştir ve gürültü olmaksızın derin ağ ile kıyaslanmıştır. Tablo 8'de görüldüğü gibi test hatası 0,8608 puan ile [512, 256, 512] ağı 0,1 gürültü oranında en iyi sonuç alınmıştır.



Şekil 6. Adam Optimizasyon Algoritması (Elu)

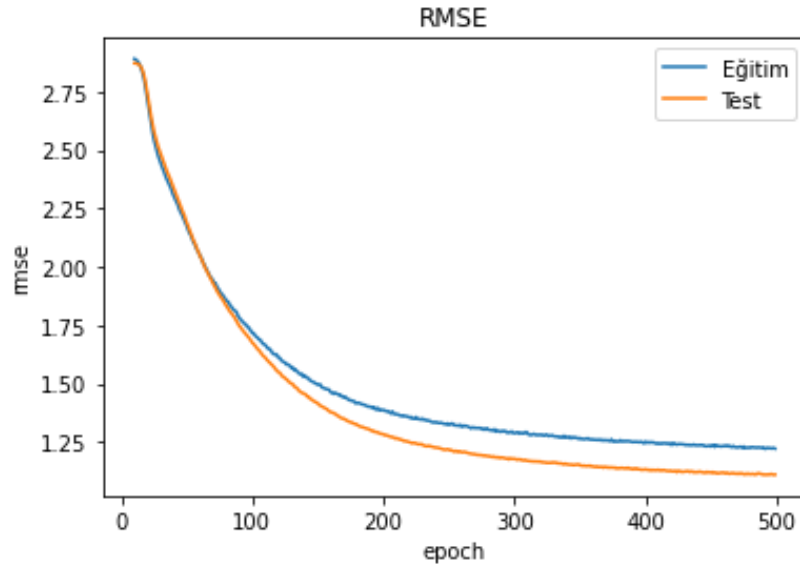
Tablo 8’de [512, 256, 512] modelinde noise oranı 0,1 ile en iyi sonucu vermiştir, eğitim ve test hatası Şekil 6’da grafik olarak görülmektedir.

Sonraki test için relu fonksiyonu kullanılarak modele etkisi incelenmiştir. Eğitim toplam 500 devirde tamamlanmış her devir sonunda test verisi tasarlanan modelde test edilmiştir.

Tablo 9. Deneysel çalışmada elde edilen sonuçlar (Relu)

MODEL (Relu)	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	0,8738	0,888
[256, 512, 256], Noise:0,3	0,9059	0,8992
[256, 512, 256], Noise:0,5	0,9883	0,9114
[256, 512, 256]	0,8687	0,8847
[512, 256, 512], Noise:0,1	1,1354	0,8773
[512, 256, 512], Noise:0,3	1,2471	0,8902
[512, 256, 512], Noise:0,5	1,4359	0,906
[512, 256, 512]	1,2534	0,8678
[512, 512, 1024, 512, 512], Noise:0,1	0,8751	0,8564
[512, 512, 1024, 512, 512], Noise:0,3	0,9041	0,8622
[512, 512, 1024, 512, 512], Noise:0,5	0,9986	0,8716
[512, 512, 1024, 512, 512]	0,8737	0,8509
[1024, 512, 256, 512, 1024], Noise:0,1	2,6092	0,8593
[1024, 512, 256, 512, 1024], Noise:0,3	2,5424	0,8679
[1024, 512, 256, 512, 1024], Noise:0,5	2,5875	0,8787
[1024, 512, 256, 512, 1024]	2,6671	0,8528

Oluşturulan modelde relu aktivasyon fonksiyonu gürültü değeri artırılarak sonuçlar incelenmiştir. Tablo 9’da görüldüğü gibi test hatası 0,8687 puan ile [256, 512, 216] ağı en iyi sonuç alınmıştır. Eğitim ve test hatası Şekil 7’de grafik olarak görülmektedir.



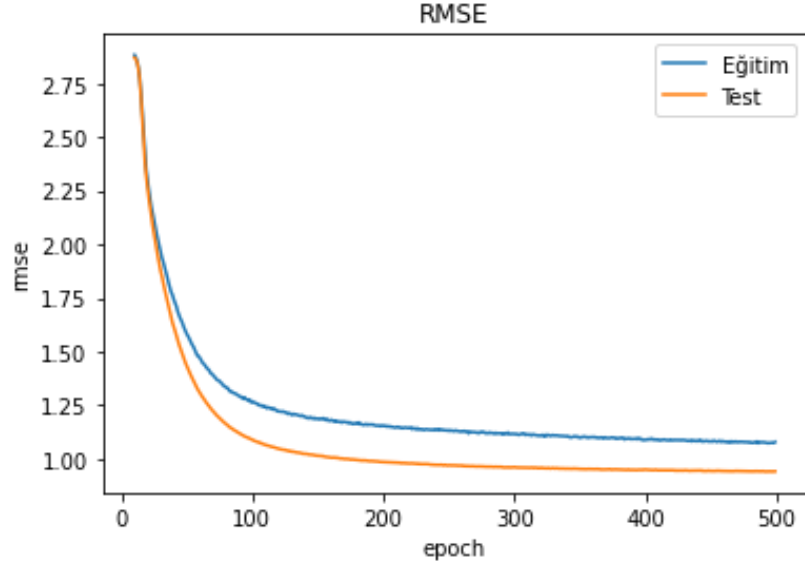
Şekil 7. Adam Optimizasyon Algoritması (Relu)

Optimizasyon algoritması değiştirilerek SGD kullanılan testlerdeki sonuçların detayları şu şekildedir;

Tablo 10. Deneysel çalışmada elde edilen sonuçlar (Selu)

MODEL (Selu)	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	0,975	1,1439
[256, 512, 256], Noise:0,3	0,9747	1,171
[256, 512, 256], Noise:0,5	0,9756	1,1905
[256, 512, 256]	0,9755	1,1484
[512, 256, 512], Noise:0,1	0,9799	1,1946
[512, 256, 512], Noise:0,3	0,9832	1,2273
[512, 256, 512], Noise:0,5	0,9848	1,2502
[512, 256, 512]	0,9813	1,1937
[512, 512, 1024, 512, 512], Noise:0,1	0,9511	1,0432
[512, 512, 1024, 512, 512], Noise:0,3	0,9482	1,0488
[512, 512, 1024, 512, 512], Noise:0,5	0,9578	1,0642
[512, 512, 1024, 512, 512]	0,9478	1,034
[1024, 512, 256, 512, 1024], Noise:0,1	0,9476	1,0628
[1024, 512, 256, 512, 1024], Noise:0,3	0,9478	1,072
[1024, 512, 256, 512, 1024], Noise:0,5	0,9545	1,0852
[1024, 512, 256, 512, 1024]	0,9487	1,0636

Oluşturulan modellerde SGD optimizasyon algoritması ile Selu aktivasyon fonksiyonu, gürültü değeri artırılarak sonuçlar incelenmiştir ve gürültü olmaksızın derin ağ ile kıyaslanmıştır. Tablo 10’de görüldüğü gibi test hatası 0,9476 puan ile [1024, 512, 256, 512, 1024] ağı 0,1 gürültü oranında en iyi sonuç alınmıştır.



Şekil 8. SGD Optimizasyon Algoritması (Selu)

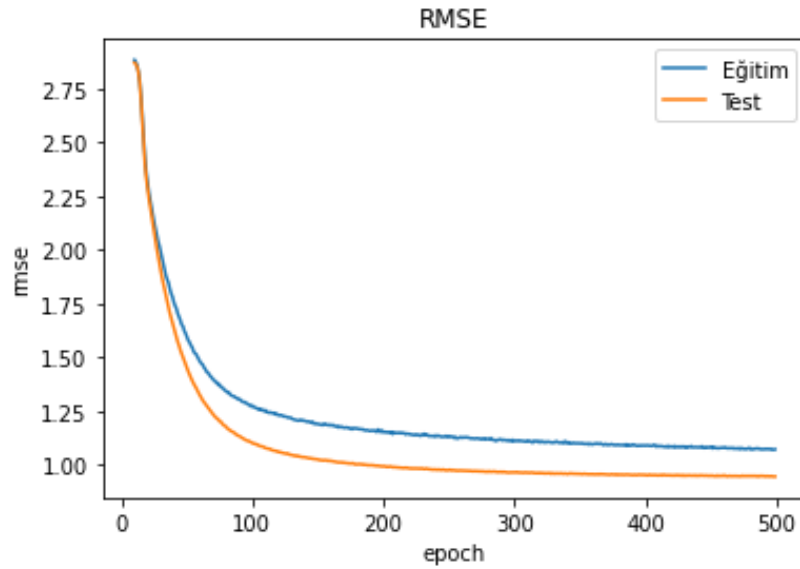
Tablo 10’da [1024, 512, 256, 512, 1024] modeli 0,1 noise oranı en iyi sonucu vermiştir, eğitim ve test hatası Şekil 8’de grafik olarak görülmektedir.

Sonraki test için elu fonksiyonu kullanılarak modele etkisi incelenmiştir. Eğitim toplam 500 devirde tamamlanmış her devir sonunda test verisi tasarlanan modelde test edilmiştir.

Tablo 11. Deneysel çalışmada elde edilen sonuçlar (elu)

MODEL (Elu)	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	0,9764	1,1139
[256, 512, 256], Noise:0,3	0,9877	1,1347
[256, 512, 256], Noise:0,5	1,0335	1,1514
[256, 512, 256]	0,9807	1,1255
[512, 256, 512], Noise:0,1	0,985	1,1654
[512, 256, 512], Noise:0,3	1,0206	1,181
[512, 256, 512], Noise:0,5	1,135	1,1954
[512, 256, 512]	0,9839	1,1666
[512, 512, 1024, 512, 512], Noise:0,1	0,9494	1,0344
[512, 512, 1024, 512, 512], Noise:0,3	0,9576	1,0316
[512, 512, 1024, 512, 512], Noise:0,5	0,9935	1,0467
[512, 512, 1024, 512, 512]	0,9486	1,0278
[1024, 512, 256, 512, 1024], Noise:0,1	0,9456	1,0742
[1024, 512, 256, 512, 1024], Noise:0,3	0,9706	1,0806
[1024, 512, 256, 512, 1024], Noise:0,5	1,039	1,078
[1024, 512, 256, 512, 1024]	0,9471	1,0756

Oluşturulan modelde SGD optimizasyon algoritması ile Elu aktivasyon fonksiyonu, gürültü değeri artırılarak sonuçlar incelenmiştir ve gürültü olmaksızın derin ağ ile kıyaslanmıştır. Tablo 11’de görüldüğü gibi test hatası 0,9456 puan ile [1024, 512, 256, 512, 1024] ağ 0,1 gürültü oranında en iyi sonuç alınmıştır.



Şekil 9. SGD Optimizasyon Algoritması (elu)

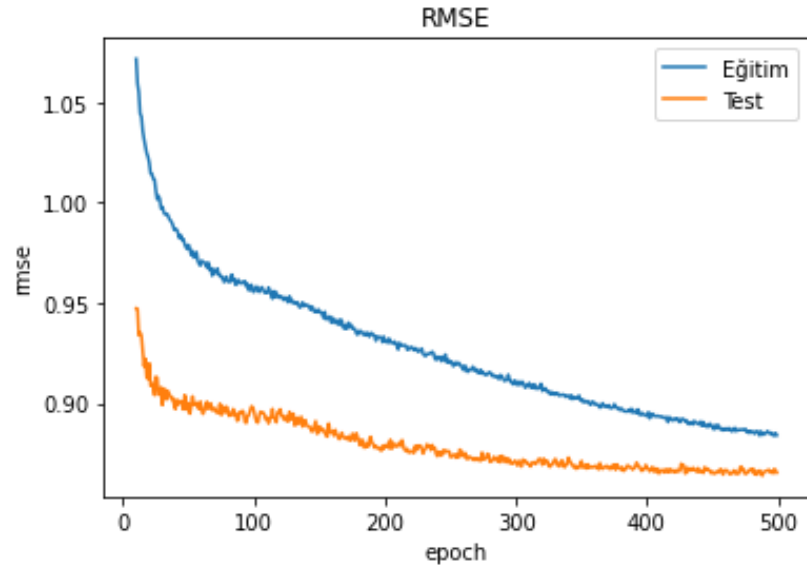
Tablo 11’de SGD Algoritması, elu fonksiyonu ile 0,1 noise oranı en iyi sonucu vermiştir, eğitim ve test hatası Şekil 9’da grafik olarak görülmektedir.

Sonraki test için relu fonksiyonu kullanılarak modele etkisi incelenmiştir. Eğitim toplam 500 devirde tamamlanmış her devir sonunda test verisi tasarlanan modelde test edilmiştir.

Tablo 12. Deneysel çalışmada elde edilen sonuçlar (Relu)

MODEL (Relu)	TEST RMSE	TRAIN RMSE
[256, 512, 256], Noise:0,1	1,2018	1,257
[256, 512, 256], Noise:0,3	1,5153	1,3043
[256, 512, 256], Noise:0,5	1,9218	1,2181
[256, 512, 256]	1,1125	1,2241
[512, 256, 512], Noise:0,1	1,2585	1,1988
[512, 256, 512], Noise:0,3	1,6374	1,1997
[512, 256, 512], Noise:0,5	2,1478	1,2114
[512, 256, 512]	1,1421	1,2061
[512, 512, 1024, 512, 512], Noise:0,1	1,4588	1,4376
[512, 512, 1024, 512, 512], Noise:0,3	1,7441	1,4518
[512, 512, 1024, 512, 512], Noise:0,5	2,1277	1,4584
[512, 512, 1024, 512, 512]	1,36	1,4253
[1024, 512, 256, 512, 1024], Noise:0,1	1,9028	1,317
[1024, 512, 256, 512, 1024], Noise:0,3	2,1543	1,3243
[1024, 512, 256, 512, 1024], Noise:0,5	2,5501	1,2923
[1024, 512, 256, 512, 1024]	1,6682	1,3145

Oluşturulan modelde relu aktivasyon fonksiyonu gürültü değeri artırılarak sonuçlar incelenmiştir. Tablo 12’de görüldüğü gibi test hatası 1,1125 puan ile [256, 512, 216] ağı en iyi sonuç alınmıştır. Eğitim ve test hatası Şekil 10’da grafik olarak görülmektedir.



Şekil 10. SGD Optimizasyon Algoritması (Relu)

5. SONUÇ

İnternette çok fazla verinin içinde kullanıcılar ihtiyaçlarını nerede ve en kısa sürede bulmakta oldukça zorlanmaktadır. Bu sorunu gidermek ihtiyaçtan çok zorunluluk haline geldiğinden öneri sistemleri bir çok farklı alanda kullanılmaktadır. Giderek artan kullanım nedeniyle bir çok araştırmada öneri sistemlerinin verimliliğini artırmak amaçlanmaktadır. Öneri sistemleri temel olarak üç kategoride incelenmektedir. Bunlar İşbirlikçi Filtreleme, İçerik Bazlı Filtreleme ve Hibrit sistemler olarak sınıflandırılmaktadır. İşbirlikçi Filtreleme ise Hafıza tabanlı ve Model tabanlı olmak üzere ikiye ayrılır.

Son yıllarda Model bazlı işbirlikçi filtreleme yöntemleri öneri sistemlerinde sıkça kullanılmaktadır. Bu çalışmada Özkodlayıcı Ağların bir türü olarak Gürültü Giderici Özkodlayıcı Ağlar kullanılmıştır. Özkodlayıcı Ağlar girişte verilen değerler de dekoder ve enkoder yapıları sayesinde veriyi yeniden yapılandırarak çıkışta verirler. Gürültü Giderici Özkodlayıcı Ağlar ise girişte verilen veriyi belirlenen oranda bozarak çıkışta yeniden yapılandırır. Bu giriş olarak gürültülü veri verilerek gürültü giderme için eğitilebilir. Veri setine yapay olarak gürültü eklersek ve hatayı hesaplarken gürültüsüz veri ile farkını kullanırsak model sadece girişi tekrar üretmeyi değil, gürültüsünü de gidermeyi öğrenecektir. Gürültü giderici özkodlayıcıların, bildiğimiz özkodlayıcılardan tek farkı girdi olarak verdiğimiz veride rastgele eklenmiş gürültülerin de varolmasıdır. Yani model, girdi verilerinden çıktı verilerini oluştururken yaptığı yeniden inşa işleminde girdinin kendisini görmek yerine gürültü katılmış veriyi görmektedir.

Veri seti olarak kullanılan MovieLens 1M çok sayıda değişken ve veri içeren veri setidir. Model bazlı öneri sistemlerinde yapılan çalışmalarda veri analizin ayrıntılı olarak yapılması için büyük verilere ihtiyaç vardır. Çok sayıda kullanıcı değerlendirmesi bulunan veri seti sayesinde kapsamlı bir model oluşturmaya olanak sağlamıştır. Veri seti modelde kullanılmadan önce bir takım ön işleme aşamasından geçirilerek Eğitim ve Test verisi olarak ikiye ayrılmıştır. Eğitim veri seti, modeli eğitmek için kullanılırken, test veri seti doğrulama amaçlı kullanılmaktadır.

Gürültü Giderici Özkodlayıcı Ağ için kullanılacak modeller belirlenmiş ve dört farklı düğüm sayısına sahip model oluşturulmuştur. Düğüm noktaları [256, 512, 1024] düğüm sayısında seçilmiş, katman sayılarına göre ayrılmıştır. Model başarısını ölçmek için

RMSE değeri incelenmiştir. Gürültü Giderici Ağlar için her bir model için gürültü değeri artırılarak sonuçlar incelenmiştir. Gürültü değerleri 0,1, 0,3 ve 0,5 oranları için en iyi sonuç aranmıştır. Burada önemli nokta çıkışta elde edilen verinin bozulmamış ham veri ile kıyaslanarak modelin bozulan verileride öğrenmesi amaçlanmaktadır.

Yapılan testlerde iki ayrı optimizasyon algoritması kullanılmıştır. Bunlar Adam, SGD optimizasyon algoritmalarıdır. Aktivasyon fonksiyonu olarak ise üç ayrı fonksiyon sırasıyla Selu, Elu ve Relu aktivasyon fonksiyonları kullanılmıştır. Oluşturulan modellerde dört farklı katman sayısı ile Gürültü Giderici Öz Kodlayıcı ağlar oluşturulmuştur.

Sonuçların başarısını incelemek için belirlenen her model için gürültü oranları ile herhangi bir gürültü eklenmeksizin özkodlayıcı ağın sonuçları kıyaslanmıştır. Kıyaslamada test verilerine ait RMSE değerleri tablo halinde çıkarılmıştır. Farklı modellerde alınan verilen deneysel sonuçlar kısmında detaylı olarak incelenmiştir.

Yapılan test sonuçları için en başarılı model [512, 256, 512] katman sayısının bulunduğu, aktivasyon fonksiyonunun SeLU olduğu ve optimizasyon yöntemi olarak da Adam optimizasyon algoritmasının kullanıldığı model RMSE si 0,8534 değeri olmuştur. Modele ait gürültü oranı ise 0,3 değerinde gerçekleşmiştir. Hem Selu aktivasyon fonksiyonu hemde Elu aktivasyon fonksiyonunda, tüm optimizasyon algoritmaları için Gürültü giderici Özkodlayıcı ağların, Özkodlayıcı ağlara göre daha başarılı olduğu görülmüştür.

Tablo 13. Bazı Farklı modellerin MovieLens 1M veri setinde Test RMSE Karşılaştırması

METHOD	TEST RMSE
I-CFN(Berg et al., 2017)	0,8321
GHRSDarban & Valipour, 2022)	0,833
BST(Chen et al., 2019)	0,8401
NNMF(Dziugaite & Roy, 2015)	0,843
Gürültü Giderici Öz Kodlayıcı	0,8534
IGMC(Zhang & Chen, 2019)	0,857
U-CFN(Strub et al., 2016)	0,8574
Factorized EAE(Hartford et al., 2018)	0,86
Factorization with dictionary learning(Mensch et al., 2016)	0,866

Tablo 13 de bazı farklı modellerin MovieLens 1M veri setinde Test RMSE Karşılaştırması verilmiştir. Literatürde MovieLens 1M veri setinde yapılan çalışmalarda önerilen yöntemin bazı geleneksel yöntemlerden daha iyi olduğu bazı yöntemler ile de yarışabildiği görülmüştür (Papers with code, 2022). Literatüre göre MovieLens 1M veri seti üzerinde en düşük RMSE değeri 0,8227 olarak GLocal-K modeli ile elde edilmiştir (Han et al., 2021).

En düşük başarı oranına sahip model ise Relu aktivasyon fonksiyonunun kullanıldığı modeller olmuştur. Ayrıca tüm sonuçlar incelendiğinde, Adam optimizasyon yönteminin diğer SGD yönteme göre, SeLU aktivasyon fonksiyonunun ise Elu fonksiyonuna göre daha başarılı olduğu görülmektedir. Tüm modellerde Relu aktivasyon fonksiyonu geride kalmıştır.

Öneri sistemlerinde Gürültü Giderici Özkodlayıcı Ağların kullanımının doğru öneriler sunmak için etkili bir yöntem olduğu görülmüştür. Sonraki yapılacak çalışmalarda Öneri doğruluğunu artırmak için sunulan metoda İçerik Bazlı Filtreleme tekniği eklenerek hibrit bir yaklaşım oluşturulabilir. Hibrit bir yaklaşım öneri doğruluğunu artırmak için etkili bir yöntem olacaktır.

KAYNAKLAR

- Abdullah, W. (2019). E-ticaret için ürün tavsiye sistem geliřtirmesi (Master's thesis, Fen Bilimleri Enstitüsü).
- Aditya, P. H., Budi, I., & Munajat, Q. (2016, October). A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X. In 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 303-308). IEEE.
- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6), 734-749.
- Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3), 48-55.
- Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using K-Means clustering and K-Nearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263-268). IEEE.
- Akay, B., Kaynar, O., & Demirkoparan, F. (2017, October). Deep learning based recommender systems. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 645-648). IEEE.
- Bag, S., Kumar, S. K., & Tiwari, M. K. (2019). An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences*, 483, 53-64.
- Baltrunas, L., & Amatriain, X. (2009, October). Towards time-dependant recommendation based on implicit feedback. In Workshop on context-aware recommender systems (CARS'09) (pp. 25-30).
- Barkan, O., & Koenigstein, N. (2016, September). Item2vec: neural item embedding for collaborative filtering. In 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6). IEEE.
- Berg, R. V. D., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.
- Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*.
- Bulut, H., & Milli, M. (2016). İşbirlikçi filtreleme için yeni tahminleme yöntemleri. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 22(2), 123-128.
- Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019, August). Behavior sequence transformer for e-commerce recommendation in alibaba. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (pp. 1-4).
- Darban, Z. Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*, 200, 116850.

- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3-4), 197-387.
- Dziugaite, G. K., & Roy, D. M. (2015). Neural network matrix factorization. arXiv preprint arXiv:1511.06443.
- Fathan, G., Adji, T. B., & Ferdiana, R. (2018, October). Impact of matrix factorization and regularization hyperparameter on a recommender system for movies. In 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 113-116). IEEE.
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.
- Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *J. Softw.*, 5(7), 745-752.
- Google colab. Google Colab. 26 Haziran 2022 tarihinde erişildi <https://colab.research.google.com/notebooks/intro.ipynb>
- Han, S. C., Lim, T., Long, S., Burgstaller, B., & Poon, J. (2021, October). GLocal-K: Global and Local Kernels for Recommender Systems. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 3063-3067).
- Hartford, J., Graham, D., Leyton-Brown, K., & Ravanbakhsh, S. (2018, July). Deep models of interactions across sets. In International Conference on Machine Learning (pp. 1909-1918). PMLR.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939.
- Koren, Y., & Bell, R. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37
- LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, 521, 436-444.
- Lian, R. (2011, June). The construction of personalized Web page recommendation system in e-commerce. In 2011 International conference on computer science and service system (CSSS) (pp. 2687-2690). IEEE.
- Mensch, A., Mairal, J., Thirion, B., & Varoquaux, G. (2016, June). Dictionary learning for massive matrix factorization. In International Conference on Machine Learning (pp. 1737-1746). PMLR.
- Miller, B. N. (2003). Toward a personal recommender system. University of Minnesota.
- Milli, M., 2013. Film öneri sistemleri için hibrit bir yöntem geliştirilmesi a new hybrid approach for movie recommender systems. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İzmir, 75s.orm.
- Movielens 1M Dataset. GroupLens. 26 Haziran 2022 tarihinde erişildi. <https://grouplens.org/datasets/movielens/1m/>
- Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72(2011), 1-19.

- Özcan, A. (2010). Sosyal Ağ Verisi Kullanılarak Mobil Telefonlar İçin Öneri Altyapısı Tasarlanması (Doctoral dissertation, Fen Bilimleri Enstitüsü).
- Öztemel, E. (2006). Yapay sinir ağları. 2. Baskı. PapatyaYayincilik, İstanbul.
- Papers with code - movielens 1M benchmark (recommendation systems). The latest in Machine Learning. (n.d.). Retrieved July 26, 2022, from <https://paperswithcode.com/sota/collaborative-filtering-on-movielens-1m>.
- Patra, S., & Ganguly, B. (2019). Improvising singular value decomposition by KNN for use in movie recommender systems. *Journal of Operations and Strategic Planning*, 2(1), 22-34.
- Pennock, D. M., Horvitz, E. J., Lawrence, S., & Giles, C. L. (2013). Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. arXiv preprint arXiv:1301.3885.
- Rafter, R. (2010). Evaluation and conversation in collaborative filtering (Doctoral dissertation, University College Dublin).
- Rokach, L., Ricci, F., & Shapira, B. (Eds.). (2015). *Recommender systems handbook*. Springer.
- Sadıkoğlu, E., & Okkalıoğlu, B. D. (2020). Çok Kriterli Ürün-Tabanlı İşbirlikçi Filtrelemede Ağırlıklandırma Yöntemlerini Kullanarak Tahmin Performansının Arttırılması. *Avrupa Bilim ve Teknoloji Dergisi*, 110-121.
- Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning* (pp. 791-798).
- Santolaya, D. S. (2017). Using recurrent neural networks to predict customer behavior from interaction data. University of Amsterdam.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. Minnesota Univ Minneapolis Dept of Computer Science.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295).
- Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery*, 5(1), 115-153.
- Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015, May). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web* (pp. 111-112).
- Sharma, A. (2018). Implementing Autoencoders in Keras. DataCamp.
- Shardanand, U., & Maes, P. (1995, May). Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 210-217).

- Strub, F., Gaudel, R., & Mary, J. (2016, September). Hybrid recommender system based on autoencoders. In Proceedings of the 1st workshop on deep learning for recommender systems (pp. 11-16).
- Tran, T., & Cohen, R. (2000, July). Hybrid recommender systems for electronic commerce. In Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press (Vol. 40).
- Ungar L. H., & Foster D.P. (1998). Clustering methods for collaborative filtering, Workshop on Recommender Systems, 15th National Conference on Artificial Intelligence, Monona Terrace, USA
- Wang, Z., Wang, Y., & Wu, H. (2010). Tags meet ratings: Improving collaborative filtering with tag-based neighborhood method. In Proceedings of the Workshop on Social Recommender Systems (pp. 15-23).
- Van Meteren, R., & Van Someren, M. (2000, May). Using content-based filtering for recommendation. In Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop (Vol. 30, pp. 47-56).
- Yu, X., & Sun, S. (2010, May). Research on personalized recommendation system based on web mining. In 2010 International Conference on E-Business and E-Government (pp. 346-349). IEEE
- Zhou, M., Ding, Z., Tang, J., & Yin, D. (2018, February). Micro behaviors: A new perspective in e-commerce recommender systems. In Proceedings of the eleventh ACM international conference on web search and data mining (pp. 727-735).
- Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., & Cai, D. (2017, August). What to Do Next: Modeling User Behaviors by Time-LSTM. In IJCAI (Vol. 17, pp. 3602-3608)..
- Zhang, M., & Chen, Y. (2019). Inductive matrix completion based on graph neural networks. arXiv preprint arXiv:1904.12058.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Akif IŞIK

EĞİTİM DURUMU

Lisans Öğrenimi : 2008, Selçuk Üniversitesi, Mühendislik Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği

Bildiği Yabancı Diller : İngilizce

İŞ DENEYİMİ

Çalıştığı Kurumlar : 2008-2019 Türk Telekom, Saha Op. Yön.

Uzmanı : 2020-Halen Necmettin Erbakan Üniversitesi, Network

Tarih: 05 Temmuz 2022