



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



VERİ SEYREKLİĞİ VE
ÖLÇEKLENEBİLİRLİK PROBLEMLERİNİ
GİDERMEK İÇİN DERİN OTOMATİK
KODLAYICI TABANLI YENİ BİR TAVSİYE
SİSTEMİ MODELİ

Sümeyye Sena ÜNALDI

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Sümeyye Sena ÜNALDI tarafından hazırlanan “Veri Seyrekliđi Ve Ölçeklenebilirlik Problemlerini Gidermek için Derin Otomatik Kodlayıcı Tabanlı Yeni Bir Tavsiye Sistemi Modeli ” adlı tez çalışması 04/07/2022 tarihinde aşağıdaki jüri tarafından oy birliđi ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliđi Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç. Dr. Hüseyin HAKLI

Danışman

Dr. Öğr. Üyesi Ayşe Merve ACILAR

Üye

Dr. Öğr. Üyesi Murat IŞIK

İmza

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../2022 gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. İbrahim KALAYCI
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Sümeyye Sena ÜNALDI

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

VERİ SEYREKLİĞİ VE ÖLÇEKLENEBİLİRLİK PROBLEMLERİNİ GİDERMEK İÇİN DERİN OTOMATİK KODLAYICI TABANLI YENİ BİR TAVSİYE SİSTEMİ MODELİ

Sümeyye Sena ÜNALDI

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ayşe Merve ACILAR

2022, 55 Sayfa

Jüri

Dr. Öğr. Üyesi Ayşe Merve ACILAR

Doç. Dr. Hüseyin HAKLI

Dr. Öğr. Üyesi Murat IŞIK

Tavsiye sistemleri kullanıcının karar vermesini kolaylaştırmak amacıyla oluşturulan yapılardır. Bu yapıda kullanıcının geçmiş etkinlikleri, benzer kullanıcılar veya benzer öğeler gibi bilgiler kullanılarak mevcut kullanıcıya uygun ürün tavsiyesi sunulur. İnternet ve e-ticaret sitelerinin kullanımının artması ile tavsiye sistemleri popüler bir çalışma alanı haline gelmiştir. Son zamanlarda tavsiye sistemlerinin geliştirilmesi için, diğer bir popüler çalışma alanı olan derin öğrenme metotlarından Otomatik Kodlayıcıların kullanımı yaygınlaşmıştır. Günümüzde büyümeye devam eden veri miktarı işlenmeye ihtiyaç duymaktadır. Otomatik Kodlayıcılar giriş verisinden gizli özellikleri modelleyerek bu ihtiyacı karşılar. Zor ve karmaşık problemlerin çözümünü mümkün kılan Otomatik Kodlayıcıların simetrik ağ yapısı oy tahmini için kullanılmaktadır. Tavsiye sistemlerinde kullanılan bu mimarilerin girişi, ürüne verilen oyları içeren tek boyutlu bir vektör şeklinde olmaktadır. Ancak bilgisayarlı görme alanında, iki boyutlu resim bilgilerini işleyen başarılı Derin Otomatik Kodlayıcı mimarileri bulunmaktadır. Özellikle gürültülü/eksik resimler için bu model tercih edilmekte, resmin daha net ve anlamlı şeklinin elde edilmesi için kullanılmaktadır. Bu motivasyondan yola çıkılarak, ürünlerin oylarını içeren vektöre, her bir ürüne ait tür bilgisi de eklenerek iki boyutlu bir giriş matrisi elde edilmiş ve bu matrisi giriş olarak kullanan yeni bir Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec - Deep AutoEncoder based Recommender System) modeli sunulmuştur. Bilgimiz dahilinde iki boyutlu giriş alan derin otomatik kodlayıcı tabanlı bir tavsiye sistemi literatürde bulunmamaktadır. Sunulan model MovieLens100K ve MovieLens1M veri kümeleri üzerinde test edilmiştir. DAERec yöntemi her iki veri setinde de kesinlik, geri çağırma ve F1 skoru değerlendirme kriterleri için 0,5'in üzerinde sonuçlar elde etmiştir. MovieLens100k veri kümesi üzerinde en iyi ortalama mutlak hata değeri 0,635, MovieLens1m üzerinde 0,656'dır. Literatürde bulunan on bir farklı yöntemle karşılaştırılan önerilen DAERec yöntemi, diğer yöntemlerin sonuçlarına kıyasla önemli bir fark ile en isabetli tavsiyeleri üretmiştir.

Anahtar Kelimeler: Derin Öğrenme, İşbirlikçi Filtreleme, Otomatik Kodlayıcılar, Tavsiye Sistemleri

ABSTRACT

MS THESIS

A NOVEL DEEP AUTOENCODER-BASED RECOMMENDATION SYSTEM MODEL TO RESOLVE THE DATA SPARSITY AND SCALABILITY PROBLEMS

Sümeyye Sena ÜNALDI

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Asst.Prof. Dr. Ayşe Merve ACILAR

2022, 55 Pages

Jury

**Asst. Prof. Dr. Ayşe Merve ACILAR
Assoc. Prof. Dr. Hüseyin HAKLI
Asst. Prof. Dr. Murat IŞIK**

Recommendation systems are structures created to facilitate user decision-making. In this structure, a suitable product recommendation is offered to the current user by using information such as the user's past activities, similar users or similar items. With the increased use of the Internet and e-commerce sites, recommendation systems have become a popular field of study. Recently, the use of Autoencoders, another popular area of study, of deep learning methods, has become widespread for the development of recommender systems. The amount of data that continues to grow today needs processing. Autoencoders meet this need by modeling hidden features from the input data. The symmetrical network structure of Autoencoders, which makes it possible to solve difficult and complex problems, is used for vote estimation. The input of these architectures used in recommendation systems is in the form of a one-dimensional vector containing the votes cast for the product. However, in the field of computer vision there are successful Deep Auto-Encoder architectures that process two-dimensional picture information. This model is preferred especially for noisy/incomplete pictures, and it is used to obtain a clearer and more meaningful form of the picture. Based on this motivation, a two-dimensional input matrix was obtained by adding the type information of each product to the vector containing the votes of the products, and a new Deep AutoEncoder Based Recommender System (DAERec) model was presented using this matrix as an input. . To the best of our knowledge, a deep automatic encoder based recommendation system that takes two-dimensional input is not available in the literature. The presented model has been tested on the MovieLens100K and MovieLens1M datasets. The DAERec method achieved results above 0,5 for precision, recall and F1 score evaluation criteria in both datasets. The best average absolute error value on the Movielens100k dataset is 0,635, on the Movielens1m it is 0,656. The proposed DAERec method, which was compared with eleven different methods in the literature, produced the most accurate recommendations with a significant difference compared to the results of other methods.

Keywords: Autoencoders, Collaborative Filtering, Deep Learning, Recommender Systems

ÖNSÖZ

Tez çalışmam boyunca değerli katkılarını, yönlendirici desteğini ve anlayışını hiçbir zaman esirgemeyen danışmanım Sayın Dr. Öğr. Üyesi Ayşe Merve ACILAR'a;
Manevi destekleri ile her zaman bana moral veren sevgili aileme, arkadaşlarıma ve iş arkadaşlarıma;
İçtenlikle teşekkür eder ve minnetlerimi sunarım.

Sümeyye Sena ÜNALDI
KONYA-2022



İÇİNDEKİLER

ÖZET	i
ABSTRACT.....	ii
ÖNSÖZ	iii
ŞEKİLLER LİSTESİ	vi
ÇİZELGELER LİSTESİ	viii
SİMGELER VE KISALTMALAR	ix
1.GİRİŞ	1
1.1.Tezin Amacı ve Önemi	3
1.2. Kaynak Araştırması	4
1.3.Tezin Organizasyonu	5
2. TAVSİYE SİSTEMLERİ.....	6
2.1. İşbirlikçi Filtreleme Tabanlı Yöntemler	6
2.2. İçerik Tabanlı Yöntemler	7
2.3. Hibrit Yöntemler.....	8
3. MATERYAL VE YÖNTEM.....	9
3.1. Veri Kümeleri	9
3.2. Yapay Sinir Ağları (YSA)	9
3.3. Derin Otomatik Kodlayıcılar	10
3.3.1.Optimizasyon Algoritması	12
3.3.1.1.Adam.....	12
3.3.1.2.RMSprop.....	12
3.3.1.3.Adadelata	12
3.3.1.4.Adagrad.....	13
3.3.2.Aktivasyon Fonksiyonu	13
3.3.2.1.Sigmoid	13
3.3.2.2.ELU (Exponential Linear Unit)	13
3.3.2.3.CELU (Continuously Differentiable Exponential Linear Unit)	14
3.3.2.4.ReLU (Rectified Linear Unit)	14
3.3.2.5.ReLU6.....	15
3.3.2.6.Softsign	15
3.3.2.7.Softplus	16
3.3.3.Gizli Katman Sayısı	16
3.3.4.Öğrenme Adımı	16
3.3.5.Ağırlık Kaybı	17
3.4. K-Kat Çapraz Doğrulama	17
3.5. Değerlendirme Kriteri.....	17
3.5.1. Ortalama Kare Hata	17

3.5.2. Kök Ortalama Kare Hata	18
3.5.3.Ortalama Mutlak Hata	18
3.5.4. Kesinlik	18
3.5.5.Geri Çağırma.....	19
3.5.6. F1 skoru	19
3.6. Geliştirme Ortamı	19
4.TEZDE ÖNERİLEN DERİN OTOMATİK KODLAYICI TABANLI TAVSİYE SİSTEMİ (DAERec)	20
4.1. DAERec için İki Boyutlu Tür-Oy Matrisinin Oluşturulması ve Ağın Eğitimi....	21
4.2 DAERec ile Tahmin Hesaplama ve Tavsiye Üretme	23
4.3.DAERec Parametre Analizleri ve Elde Edilen Uygulama Sonuçları	25
4.3.1. Tek Boyutlu Giriş için Yapılan Hiperparametre Analizi Sonuçları	25
4.3.2 İki Boyutlu Giriş (Tür-Oy Matrisi) için Yapılan Hiperparametre Analizi Sonuçları	28
5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	32
5.1. Tahmin Değerlendirme Sonuçları.....	33
5.2. Tavsiye Değerlendirme Sonuçları	38
6. SONUÇLAR VE ÖNERİLER	48
7. KAYNAKLAR	50
ÖZGEÇMİŞ	Hata! Yer işareti tanımlanmamış.

ŞEKİLLER LİSTESİ

- Şekil 1.1. Derin Otomatik Kodlayıcının Eksik Resimler için Kullanılmasına Bir Örnek
- Şekil 2.1. Tavsiye Sistemleri Sınıflandırma Şeması
- Şekil 3.1. (a) Biyolojik Sinir Hücresi (b)Yapay Sinir Ağı Hücresi
- Şekil 3.2. Derin Otomatik Kodlayıcı Yapısı
- Şekil 3.3. Sigmoid Aktivasyon Fonksiyonu Grafiği
- Şekil 3.4. ELU Aktivasyon Fonksiyonu Grafiği
- Şekil 3.5. CELU Aktivasyon Fonksiyonu Grafiği
- Şekil 3.6. ReLU Aktivasyon Fonksiyonu Grafiği
- Şekil 3.7.ReLU6 Aktivasyon Fonksiyonu Grafiği
- Şekil 3.8. Softsign Aktivasyon Fonksiyonu Grafiği
- Şekil 3.9. ReLU ve Softplus Aktivasyon Fonksiyonlarının Grafiği
- Şekil 4.1. Önerilen Mimarinin Temel Blok Diyagramı
- Şekil 4.2. MovieLens100K için önerilen DAERec ağ mimarisi
- Şekil 4.3. Tür-Oy Matrisi G'nin oluşturulması ile ilgili sayısal bir örnek
- Şekil 5.1. Movielens100k'ne ait DAERec uygulamadan önceki ve sonraki eksik veri oranları
- Şekil 5.2. Çevrim Sayısına göre Tahmin Hata Grafikleri
- Şekil 5.3. Eğitim Verisinin Örnekleme Dağılımına Göre Tahmin Hata Grafikleri
- Şekil 5.4. Movielens100k 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Kesinlik Performansı
- Şekil 5.5. Movielens100k 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Geri Çağırma Performansı
- Şekil 5.6. Movielens100k 5 kere tekrar edilen 5 Kat Çapraz Doğrulama F1 Skoru Performansı
- Şekil 5.7. Movielens1m 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Kesinlik Performansı
- Şekil 5.8. Movielens1m 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Geri Çağırma Performansı
- Şekil 5.9. Movielens1m 5 kere tekrar edilen 5 Kat Çapraz Doğrulama F1 Skoru Performansı
- Şekil 5.10. Movielens100k @5 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.11. Movielens1m @5 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.12. Movielens100k @10 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.13. Movielens1m @10 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.14. Movielens100k @15 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.15. Movielens1m @15 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.16. Movielens100k @20 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

Şekil 5.17. Movielens1m @20 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skoru Sonuçları

ÇİZELGELER LİSTESİ

- Çizelge 2.1. Tavsiye Sistemlerindeki Son Gelişmeler
- Çizelge 2.2. İş birliği Filtreleme yöntemlerinde sıklıkla kullanılan yöntemler
- Çizelge 3.1. Çalışmada Kullanılan Veri Kümelerinin Özellikleri
- Çizelge 4.1. DAERec Ağının Eğitim için kullanılan algoritmanın sözde kodu
- Çizelge 4.2. DAERec ile Tahmin Hesaplama ve Tavsiye Üretmek için kullanılan algoritmanın sözde kodu
- Çizelge 4.3. Optimizasyon Algoritması ve Aktivasyon Fonksiyonu Seçimi
- Çizelge 4.4. Katman Sayısı Belirleme Deney Sonuçları
- Çizelge 4.5. Öğrenme Adımı Belirleme Deney Sonuçları
- Çizelge 4.6. Ağırlık Kaybı Belirleme Deney Sonuçları
- Çizelge 4.7. Optimizasyon Algoritması ve Aktivasyon Fonksiyonu Seçimi
- Çizelge 4.8. Katman Sayısı Belirleme Deney Sonuçları
- Çizelge 4.9. Öğrenme Adımı Belirleme Deney Sonuçları
- Çizelge 4.10. Ağırlık Kaybı Belirleme Deney Sonuçları
- Çizelge 5.1. Kıyaslama için kullanılan algoritmalar
- Çizelge 5.2. Movielens100k üzerindeki OMH ve KOKH eğitim performansları
- Çizelge 5.3. Movielens1m üzerindeki OMH ve KOKH eğitim performansları
- Çizelge 5.4. DAERec yönteminin Movielens100k veri kümesinde top-n tavsiye için kesinlik, geri çağırma ve F1 skoru performansları
- Çizelge 5.5. DAERec yönteminin Movielens1m veri kümesinde top-n tavsiye için kesinlik, geri çağırma ve F1 skoru performansları

SİMGELER VE KISALTMALAR

x	: Giriş
$f(x)$: Çıkış
a	: Ağırlık
s	: Sapma
K	: Kullanıcı matrisi
M	: Ürün matrisi
k	: Kullanıcı sayısı, $u \in K = \{1, \dots, k\}$
m	: Ürün sayısı, $i \in M = \{1, \dots, m\}$
t	: Tür sayısı
G	: Tür- Oy matrisi, $G \in R^{m \times t}$
G'	: Tahmin matrisi, $G' \in R^{m \times t}$
E_φ	: Otomatik Kodlayıcının kodlayıcı kısmı
D_θ	: Otomatik Kodlayıcının kod çözücü kısmı
\mathcal{L}	: Hata (Loss) Değeri
İFT	: İşbirlikçi Filtreleme Tabanlı Tavsiye Sistemleri
RBM	: Kısıtlanmış Boltzmann Makinesi (Restricted Boltzmann Machine)
MF	: Matris Çarpanlara Ayırma (Matrix Factorization)
SDAE	: Yığılanmış Gürültü Giderici Otomatik Kodlayıcı (Stacked Denoising Autoencoder)
YSA	: Yapay Sinir Ağları
DAE	: Derin Otomatik Kodlayıcı (Deep Autoencoders)
SAE	: Yığılanmış Otomatik Kodlayıcı (Stacked Autoencoder)
RMSprop	: Kök Ortalama Kare Yayılımı (Root Mean Square Propagation)
ReLU	: Düzeltilmiş Doğrusal Birim (Rectified Linear Unit)
ELU	: Üssel Doğrusal Birim (Exponential Linear Unit)
CELU	: Sürekli Türevlenebilir Üstel Doğrusal Birim (Continuously Differentiable Exponential Linear Unit)
OKH	: Ortalama Kare Hata (MSE-Mean Square Error)
KOKH	: Kök Ortalama Kare Hata (RMSE – Root Mean Square Error)
OMH	: Ortalama Mutlak Hata (MAE – Mean Absolute Error)
DAERec	: Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec- Deep AutoEncoder based Recommender System)

1.GİRİŞ

Çok fazla seçenek içerisinde kendisine en uygun olanlarını bulması için kullanıcıya yardım eden tavsiye sistemi algoritmaları günümüzün ilgi çekici araştırma konularındandır. Bu algoritmalar arasından, kullanıcıların kendi fikirlerini/oylamalarını beyan edebildikleri platformların çoğalmasi sebebiyle; temel dinamiği mevcut kullanıcıya benzer kullanıcılar bulup onların beğendiği ve mevcut kullanıcının bilmediği ürünleri tavsiye etmek olan işbirlikçi filtreleme tabanlı (İFT) tavsiye sistemlerinde kullanılan algoritmalar ön plana çıkmaktadır.

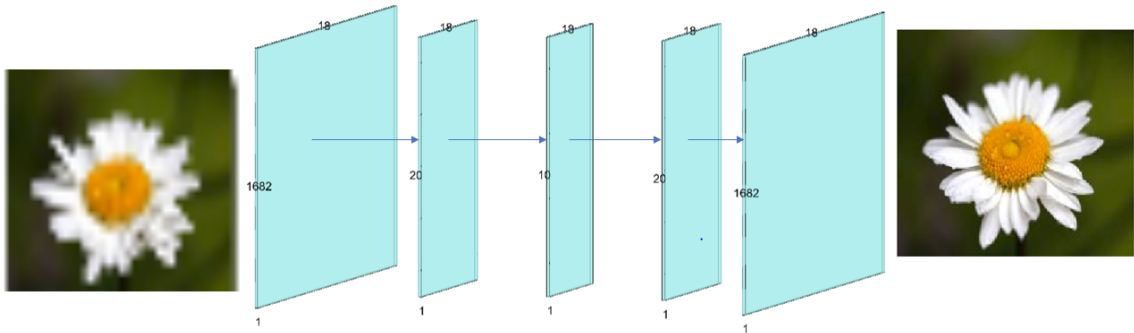
İFT tavsiye sistemleri, bilgisayarın ve insanın iyi yapabildiği şeyi yapmasına izin vererek ortak çalışma imkânı sunar. Şöyle ki; kullanıcı ürünleri okumada veya değerlendirmede iyiyken bilgisayar ise bu değerlendirmeler arasındaki benzerliklerin bulunması için gerekli olan hesaplamaları yapmakta iyidir. Kullanıcıların bir eleman için yaptıkları değerlendirmeye “oylama” denir. Oylama genellikle 1-5 veya 1-7 gibi belli bir aralıkta yapılır ve kullanıcının o eleman hakkındaki düşüncesini sayısallaştırır. Bilgisayarın buradaki rolü ise bir kullanıcının henüz görmediği bir elemana vereceği oyu önceden tahmin etmektir.

İFT tavsiye sistemleri, Komşuluk tabanlı ve Model tabanlı olarak iki alt kategoriye ayrılabilir. Komşuluk tabanlı yöntemler, benzer kullanıcıları veya öğeleri bulmaya çalışır ve oy değerlerini tahmin etmek için geçmiş tercihleri kullanırlar (Breese ve ark., 1998). Bunun için ilk adım elemanları benzer şekilde oylayan kullanıcı gruplarının tespit edilmesidir. Bu kullanıcı grubuna “komşuluk” adı verilir ve bir kullanıcının bir eleman için yaptığı tahmin, o eleman için komşularının yaptığı oylamalara bakılarak hesaplanır. Buradaki esas düşünce eğer bir kullanıcı geçmişte komşuları ile aynı fikirde ise gelecekte de aynı fikirde olma olasılığının yüksek olmasıdır (Miller ve ark., 2003). İlerleyen yıllarda, kullanıcının değerlendirdiği ürünlere verdiği oyları içeren matrislerdeki gizli kalmış kullanıcı-ürün ilişkilerini bulmak için matris çarpanlara ayırma (Matrix Factorization) yöntemlerinden yaygın olarak faydalanılmaya başlanmıştır (Jiang ve ark., 2020).

Model tabanlı yöntemler, makine öğrenmesi temellidir (B.Thorat ve ark., 2015). Veriler arasındaki karmaşık derecelendirme modellerini bularak bu modele uygun tahminler üretir. Bu başlık altında yapay sinir ağı (YSA) tabanlı yaklaşımlar, doğrusal olmayan özellikleri temsil etmede göstermiş oldukları başarılarından dolayı sıklıkla kullanılmaktadırlar. YSA tabanlı derin öğrenme modelleri, literatürde en sık karşımıza

çıkan yöntemlerdendir. Derin öğrenme tabanlı tavsiye sistemleri, performansı etkileyen veri seyrekliği sorunlarıyla karşı karşıyadır. Kullanıcılar çoğu durumda oy vermeye yatkın olmadıklarından, bu seyrek verilere dayanarak kullanıcıların potansiyel ilgi alanlarını öğrenmek zordur. Bu nedenle, kullanıcı tercihlerini öğrenmek için seyrek girdilerden, sağlam özellikleri öğrenmek için geliştirilmiş derin bir sinir ağı olan otomatik kodlayıcı (DAE) modeline dayalı çok sayıda çalışma geliştirilmiştir (Pan ve ark., 2021).

Bu tez çalışmasında da, içerisinde öğelere ait tür bilgisi de bulunan veri kümeleri için yeni bir Derin Otomatik Kodlayıcı mimarisi sunulmuş ve parametre analizleri gerçekleştirilmiştir. Tavsiye sistemlerinde kullanılan DAE mimarilerinin girişi, ürüne (Film tavsiye sistemlerinde ürünler, filmlerdir) verilen oyları içeren tek boyutlu bir vektör şeklinde olmaktadır. Ancak bilgisayarlı görme alanında, 2 boyutlu resim bilgilerini işleyen başarılı DAE mimarileri bulunmaktadır. Özellikle gürültülü/eksik resimler için bu model tercih edilmekte (Şekil 1.1), resmin daha net ve anlamlı şeklinin elde edilmesi için kullanılmaktadır (Ashgar M.H.,2020).



Şekil 1.1. Derin Otomatik Kodlayıcının Eksik Resimler için Kullanılmasına Bir Örnek

Bu motivasyondan yola çıkılarak, ürünlerin oylarını içeren vektöre, her bir ürüne ait tür bilgisi de eklenerek iki boyutlu bir giriş matrisi elde edilmiş ve bu matrisi giriş olarak kullanan yeni bir Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec - Deep AutoEncoder based Recommender System) modeli sunulmuştur. Bilgimiz dahilinde iki boyutlu giriş alan derin otomatik kodlayıcı tabanlı bir tavsiye sistemi literatürde bulunmamaktadır. Sunulan model MovieLens100K ve MovieLens1M veri kümeleri üzerinde test edilmiştir.

Önerilen DAERec yöntemi, literatürdeki on bir farklı yöntem ile kıyaslanmış ve tahmin değerlendirmesinde MovieLens100k veri kümesi üzerinde 0,635 OMH (Ortalama Mutlak Hata) değeri ile en düşük hatayı elde ederek başarısını ortaya koymuştur.

DAERec tavsiye deęerlendirmesinde her iki veri kümesinde de 0,5'in üstünde F1 skor deęeri olarak dięer yöntemler içerisindeki en yüksek başarıyı elde etmiştir.

1.1. Tezin Amacı ve Önemi

İşbirlikçi Filtreleme tekniğinin pek çok başarılı uygulaması olmasına rağmen halen giderilmesi gereken bazı problemler içermektedir. Bunlardan başlıcaları veri seyrekliği (sparsity), ölçeklenebilirlik (scalability) ve soğuk başlangıç (cold start) problemleridir.

Veri seyrekliği, sistemdeki ürünlerin sayısının arttıkça kullanıcının oyladığı ürünlerin oranının azalması ve buna bağlı olarak komşuluk hesabının zorlaşmasıdır. Ölçeklenebilirlik, genellikle büyük veri tabanları için geçerli olan öneri sistemi algoritmalarının veri kümesinin büyüklüğünden dolayı çalışma zamanlarının uzaması ve performanslarının düşmesi olarak tanımlanabilir. Soğuk başlangıç problemi ise, sisteme yeni gelen kullanıcının geçmiş oylama bilgisinin olmaması sorunudur. Buna çözüm olarak en sık kullanılan yöntem, başlangıçta belli sayıda rastgele ürünü kullanıcıya sunmak ve oylamasını istemektir. Böylelikle kullanıcı için bir başlangıç profilinin oluşturulması hedeflenir.

Bu tez çalışmasında, bu sorunları gidermek için yeni bir Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec) modelinin sunulması amaçlanmıştır.

Ürünlerin oylarını içeren vektöre her bir ürüne ait tür bilgisi de eklenerek elde edilen iki boyutlu matrisin önerilen modele giriş olarak verilmesi ile eğitilmiş modelin çıkışında, daha fazla bilgi içeren dolu bir matris elde edilmektedir. Tahmin ve tavsiye için modelin çıkışında elde edilen bu matris kullanılmakta ve veri seyrekliği problemi de modelin doğası gereği eğitim aşamasında çözülmüş olmaktadır. Model eğitim bittikten sonra kaydedilir. Anlık tavsiye oluşturulurken de, kaydedilen bu eğitilmiş model kullanılır. Bu sayede ölçeklenebilirlik sorunundan dolayı yaşanacak tavsiye gecikmelerinden kullanıcı korunmuş olur. Ayrıca deneysel çalışmaların sonucunda, ağın eğitiminin uzun sürelerle gerek kalmadan tamamlandığı görülmektedir. Tür bilgisi ile girişin taşıdığı bilgi miktarı artırılarak eğitim esansında da ölçeklenebilirlik probleminin çözümüne katkı sağlanmıştır. Son olarak, soğuk başlangıç problemi için kullanıcının belli sayıda filmi oylamasının istenmesi yerine, sadece hangi tür filmleri sevdiğini seçmesi ve buna uygun bir başlangıç profilinin oluşturulmasının yeterli olacağı öngörülmüştür.

Çünkü önerilen model film türlerini de öğrenmektedir. Böylece kullanıcıya başlangıçta yüklenen iş yükü oldukça azaltılmış olmaktadır.

1.2. Kaynak Araştırması

İnternetin ortaya çıkışı ve cep telefonlarının gelişimi ile dünyada bir veri patlaması yaşanmıştır. Büyük veri çağı olarak adlandırılan bu dönemde değerli verilerin ayırt edilmesi veya anlamlandırılması büyük bir problem haline gelmiştir. Aşırı Bilgi Yükleme (Information Overload) problemi olarak da isimlendirilen bu sorunu çözenin bir yolu, kullanıcı için gereksiz bilgileri bir tavsiye sistemi ile filtrelemektir (F. Liu ve ark., 2021). Son yıllarda derin öğrenme yöntemlerinde yaşanan gelişmeler sonucunda, ham verileri daha yüksek seviyeli öznetelik uzayına yansıtan derin öğrenme tabanlı tavsiye sistemi algoritmaları önerilmiş ve verimli sonuçlar elde edilmiştir (Sun & Zhang, 2018). Bu hususta literatürde yer alan çalışmalardan örnekler aşağıda verilmiştir.

Tavsiye sistemleri için derin öğrenmeyi kullanmaya yönelik ilk girişimler, sınırlı Boltzmann makinelerini (RBM-Restricted Boltzmann Machine) (Georgiev & Nakov, 2013) içermektedir. 2015 yılında Suvash Sedhain Otomatik Kodlayıcıları Movielens ve Netflix verileri üzerinde kullanmış ve RBM ve MF (Matrix Factorization- Matris Çarpanlarına Ayırma) yöntemlerine göre daha iyi bir performans sağladığını göstermiştir (Sedhain ve ark., 2015). Aynı yıl Florian Strub SDAE (Stacked Denoising Autoencoder- Yığılanmış Gürültü Arındırma Otomatik Kodlayıcı) yapısını Movielens1M ve Jester veri kümesi üzerinde deneyerek 4 katmanlı Otomatik Kodlayıcıların (Derin Otomatik Kodlayıcı) 2 katmanlı Otomatik Kodlayıcılara göre daha iyi sonuçlar verdiğini göstermiştir (Strub ve ark., 2015). 2021 yılında kişiselleştirilmiş tavsiyeler sunmak için önerilen Otomatik Kodlayıcı yapısının kullanıldığı CAPR yöntemi sunulmuştur. Kişiselleştirilmiş tavsiye sistemi, kullanıcıların ürünlerle ilgili tercihlerini tahmin edip, kullanıcılara beğenebilecekleri ürünler önermektedir (Zhu ve ark., 2021). Tahmini geliştirmek için YSA ile işbirlikçi filtreleme kombinasyonu (Strub ve ark., 2016) gibi yöntemler ve sistemin etkinliğini (Kuchaiev & Ginsburg, 2017a) iyileştirmek için gürültü arındırma (denoising) mimarisi ve bırakma (dropout) gibi birçok yöntem önerilmiştir. 2016 yılında soğuk başlangıç problemini çözmek amacıyla sıkı bağlanmış bir hibrit sistem önerilmiştir (Wei ve ark., 2017). “Genel olarak, verimli bir tavsiye sistemi kullanıcılar ve işletmeler arasındaki karmaşık, doğrusal olmayan ilişkileri modelleyebilmeli ve yakalayabilmelidir.” gerekçesi ile işletmeler ve kullanıcılar için

içerik ve işbirliği özelliklerine ek olarak, alınan oy sayısına göre değerlendirilen derecelendirme kalitesini içerecek şekilde YSA kullanan bir hibrit sistem önerilmiştir (Paradarami ve ark., 2017). 2018’de veri seyrekliği problemine çözüm olarak derin sinir ağları ile çift-düzenli matris çarpanlara ayırmayı birleştiren bir yaklaşım önerilmiştir (Wu ve ark., 2018). AODR(Aspect-based Opinion mining using Deep learning method for Recommender system) modeli, ürünün özelliklerini ve temel ağırlıklı kullanıcı görüşlerini inceleme metninden derin bir öğrenme yöntemi kullanarak ayıklayarak tavsiye sistemini geliştirmek için İşbirlikçi Filtreleme yöntemiyle birleştirmiştir (Da’u ve ark., 2020a).Veri seyrekliği problemini çözmek amacıyla 2019’da yapılan bir çalışmada (W. Zhang ve ark., 2019a) ileri beslemeli derin sinir ağı yapısını kullanan DeepRec adlı bir tavsiye sistemi önerilmiştir. 2020 yılında önerilen MCRS (Nassar ve ark., 2020) adlı Çok Kriterli Tavsiye Sistemi çok-ölçütlü bir İşbirlikçi Filtreleme kullanır. 2020 yılında (R ve ark., 2020) tarafından İşbirlikçi filtreleme sistemlerindeki boşlukları ele almak için ağırlık kaybı ve öğrenme adımını dönemler arasında döngüsel olarak değiştiren bir hibrit sistem önerilmiştir.

1.3.Tezin Organizasyonu

Bu tez çalışması 6 bölümden oluşmaktadır.

Giriş bölümünde tezin amacı, önemi, literatür özeti ve tezin organizasyonu hakkında bilgi verilmiştir.

İkinci bölümde tavsiye sistemleri ve problemleri hakkında temel bilgiler sunulmuştur.

Üçüncü bölümde tez çalışmasında kullanılan veri kümeleri, derin otomatik kodlayıcı ağlar ve parametreleri, tavsiye sisteminde kullanılan değerlendirme kriterleri hakkında bilgiler verilmiştir.

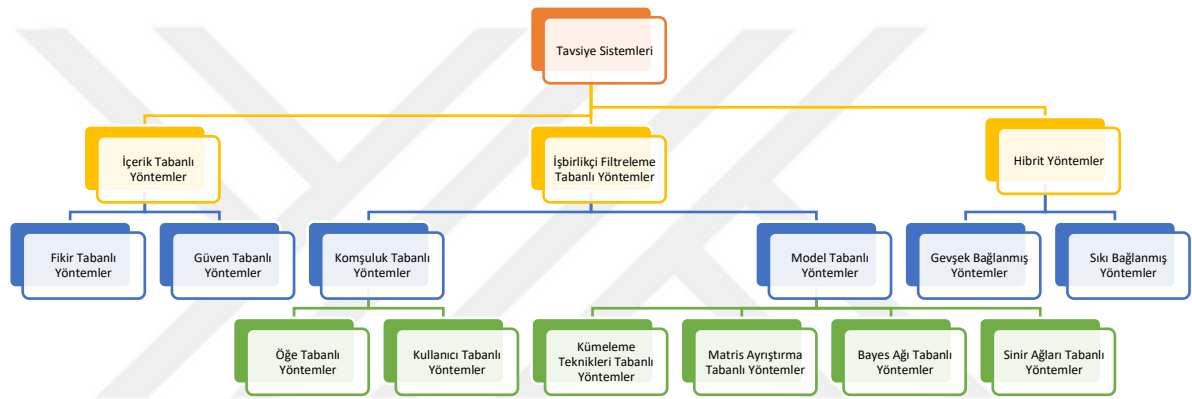
Dördüncü bölümde tez çalışmasında önerilen Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec) yöntemi açıklanmış ve ağa ait parametre analizleri gerçekleştirilmiştir ve tüm sonuçlar sunulmuştur.

Beşinci bölümde önerilen DAERec yöntemi literatürde sıklıkla kullanılan Movielens100k ve Movielens1m veri kümelerine uygulanmıştır. Elde edilen sonuçlar literatürde yer alan 11 algoritma ile kıyaslanarak sonuçlar tartışılmıştır.

Altıncı bölümde sonuçların genel değerlendirmesi yapılmış ve önerilere yer verilmiştir.

2. TAVSİYE SİSTEMLERİ

Uygulama/web sitesi kullanıcılarının ilgi alanlarına ve önceki ürün değerlendirmelerine göre, ilgi alanlarına girebilecek seçenekleri tahmin ederek uygulama kullanımını sürdürmek amacıyla geliştirilen sistemlere Tavsiye Sistemleri denir. Geleneksel arama algoritmaları, tüm kullanıcılar için aynı sonuçları sunarken tavsiye sistemleri farklı kullanıcılar için kişiselleştirilmiş hizmetler/öneriler üretir. Örneğin kullanıcıların siteye olan ilgisini artırmak için kullanıcıların ilgi alanlarına uygun öğeleri öneren Amazon, Netflix ve Youtube gibi platformlar tavsiye sistemlerini kullanmaktadır.



Şekil 2.1. Tavsiye Sistemleri Sınıflandırma Şeması

Tavsiye sistemleri içerik tabanlı, işbirlikçi filtreleme tabanlı ve hibrit yöntemler olmak üzere temelde üç başlık altında incelenir. Bu başlıklar altında yer alan alt başlıklar Şekil 2.1’de gösterilmiştir.

2.1. İşbirlikçi Filtreleme Tabanlı Yöntemler

İşbirlikçi Filtrelemeye dayalı yöntemlerde benzer kullanıcılar veya benzer öğeler esas alınarak tavsiye yapılır (Bell & Koren, 2007). Komşuluk tabanlı veya model tabanlı olarak iki alt kategoriye ayrılabilir. Komşuluk tabanlı yöntemlerde kendi içinde öğe tabanlı ve kullanıcı tabanlı olarak ikiye ayrılır. Komşuluk tabanlı işbirlikçi filtreleme benzer kullanıcıları veya öğeleri bulmaya çalışır ve derecelendirmeleri tahmin etmek için geçmiş tercihleri kullanır (Breese ve ark., 1998). Model tabanlı işbirlikçi filtreleme, makine öğrenmesi temellidir (B.Thorat ve ark., 2015). Veriler arasındaki karmaşık

derecelendirme modellerini bularak bu modele uygun tahmin üretir. AODR modeli, ürünün özelliklerini ve temel ağırlıklı kullanıcı görüşlerini inceleme metninden derin bir öğrenme yöntemi kullanarak ayıklar (Da'u ve ark., 2020b). 2019'da yapılan bir çalışmada (W. Zhang ve ark., 2019b) ileri beslemeli derin sinir ağı yapısını kullanan DeepRec adlı bir tavsiye sistemi önerilmiştir. Kullanıcılara kişiselleştirilmiş tavsiyeler sunmak için (Zhu ve ark., 2021) tarafından Otomatik Kodlayıcı yapısının kullanıldığı CAPR yöntemi sunulmuştur.

Matris Ayırıştırma (matris çarpanlara ayırma - Matrix Factorization) Tabanlı Yöntemler, düşük zaman ve alan karmaşıklığı sağlaması ve ölçeklenebilir olması sebebi ile son yıllarda İFT tavsiye sistemlerinde sıklıkla kullanılmaya başlanmıştır (Luo ve ark.,2020). Bu yöntemler, kullanıcının değerlendirdiği ürünlere verdiği oyları içeren matrislerdeki gizli kalmış kullanıcı-ürün ilişkilerini bulmayı amaçlarlar ancak seyrek verileri işlemede güçlük çekerler (Jiang ve ark., 2020).

İşbirlikçi Filtreleme tabanlı yöntemlerin bazı problemleri vardır. Bu problemlerden ilki veri seyrekliğidir. Verinin seyrek olması tahminlerin doğruluğunu önemli ölçüde etkiler. İkinci problem ise kullanıcı geçmişine dayalı olduğu için yeni kullanıcılar hakkında bilgi bulunmamasıdır. Bu problem soğuk başlangıç olarak adlandırılır. Model tabanlı bağımlılık ağları, kümeleme modelleri, bayesci modeller ve yapay sinir ağı tabanlı yöntemler komşuluk tabanlı İşbirlikçi Filtreleme algoritmalarının problemlerini çözmek için ortaya çıkmıştır (Tilahun ve ark., 2017).

2.2. İçerik Tabanlı Yöntemler

İçeriğe dayalı yöntemlerin amacı ürünlerin özelliklerini kullanarak kullanıcıya daha önce beğendiği ürüne benzer ürün önermektir (Liu ve ark., 2009). İçeriğe dayalı yöntemler kullanıcıların ilgilendikleri içeriğe göre öğeleri ve açıklamalarını kullanır. Fikir tabanlı ve güven tabanlı olmak üzere ikiye ayrılır. Literatürde fikir madenciliği olarak adlandırılan ve kullanıcıların görüşlerini değerlendiren çalışmalar bulunmaktadır (Da'u ve ark., 2020b, 2020a). Özelliğe dayalı fikir madenciliğinin amacı öge metni ve ilişkili kullanıcı görüşlerini değerlendirme metninden (kullanıcı yorumu) ayıklamaktır.

2.3. Hibrit Yöntemler

Hibrit yöntemler, içerik tabanlı ve işbirlikçi filtreleme tabanlı yöntemlerin birlikte kullanılması ile oluşur. Hibrit yöntemler derecelendirme bilgileri ve yardımcı bilgiler arasındaki ilişkiye göre gevşek bağlanmış yöntemler ve sıkı bağlanmış yöntemler olarak iki alt kategoriye ayrılabilir. Gevşek bağlanmış yöntemler tek yönlü etkileşim kullanarak yardımcı bilgileri bir defa işler ve İşbirlikçi Filtreleme modeli için kullanır. Sıkı bağlanmış yöntemler iki yönlü etkileşime sahiptir ve yardımcı bilgilerden özellikleri otomatik olarak öğrenebilir. Bu nedenle gevşek bağlanmış yöntemlere göre daha iyi performans gösterirler (Wang ve ark., 2015).

Tavsiye sistemi yöntemlerine ait avantaj ve dezavantajlar Çizelge 2.1 ve Çizelge 2.2’de sunulmuştur.

Çizelge 2.1. Tavsiye Sistemlerindeki son gelişmeler (Zhu ve ark., 2021)

Yöntemler	Avantajlar	Dezavantajlar	Örnek Çalışmalar
İçerik Tabanlı	Bu yöntemler, soğuk başlatma problemini çözebilir ve öğelerin öznetelik boyutlarını artırarak öneri doğruluğunu iyileştirebilir.	Öğeler için nitelikler oluşturmak ve belirli alanlardaki kullanıcılardan geri bildirim almak zordur.	(Van Den Oord ve ark., 2013), (Wang & Wang, 2014)
İş birlikçi Filtreleme	Bu yöntemler her tür öğeyi filtreleyebilir	Soğuk başlatma sorunu, veri seyrekliği ve ölçeklenebilirlik	(He ve ark., 2017), (Chen ve ark.,2020)
Hibrit	Bu yöntemler öneri doğruluğunu artırabilir ve daha etkili olabilir	Model yapısı daha karmaşıktır	(Shuai ve ark., 2017), (Guo ve ark.,2017)

Çizelge 2.2. İş birlikçi Filtreleme yöntemlerinde sıklıkla kullanılan yöntemler

Yöntem Adı	Avantajlar	Dezavantajlar	Örnek Çalışmalar
Matris ayrıştırması	Düşük zaman ve alan karmaşıklığı, iyi ölçeklenebilirlik	Bu yöntemler, seyrek verileri işlemede güçlük çeker ve genelleme yeteneği zayıftır.	(Lee & Seung, 2001), (Luo ve ark.,2020)
Derin sinir ağları	Doğrusal olmayan dönüşüm ve güçlü temsil öğrenme yeteneği	Yorumlanabilirlik, veri gereksinimi ve kapsamlı hiper parametre ayarı	(Wang ve ark., 2015b), (Chaeve ark., 2019)

3. MATERYAL VE YÖNTEM

3.1. Veri Kümeleri

Tezde yapılan çalışmalar, literatürde sıklıkla kullanılan ve içerisinde öğelere ait tür bilgilerini de barındıran Movielens100k ve Movielens1M veri kümeleri üzerinde gerçekleştirilmiştir. Bu kümelere ait bilgiler Çizelge 3.1’de verilmiştir.

Movielens100k: Movielens100k veri kümesi 943 kullanıcının 1682 filmde en az 20 filmi oyladığı 100.000 derecelendirmeden yani kullanıcının filme verdiği oy değerinden oluşur. Bu veri kümesi kullanıcı kimliği, film kimliği, derecelendirme, film türleri ve zaman damgası niteliklerini içerir (Çizelge 3.1). Bu çalışmada bir zaman verisi beklenmediği için Zaman Damgası niteliği kullanılmamıştır. Her film aksiyon, macera, animasyon, çocuk, komedi, suç, belgesel, dram, fantezi, kara-film, korku, müzikal, gizem, romantik, bilim kurgu, gerilim, savaş ve western türlerinden birine/birkaçına aittir. 124 tane filmin türü bilinmemektedir. Türü bilinmeyen filmler veri kümesinden çıkarılmıştır. Veri kümesinde film ilgili türe ait ise 1, ait değil ise 0 ile gösterilmiştir.

Movielens1M: Movielens1M veri kümesi 6040 kullanıcının 3952 film için kullandığı 1.000.211 oy değerinden oluşur (Çizelge 3.1). Movielens100k veri kümesi ile aynı nitelikleri içerir.

Çizelge 3.1. Çalışmada Kullanılan Veri Kümelerinin Özellikleri

Veri kümesi	Kullanıcı	Öğe	Oy	Oylama Aralığı	Tür Sayısı	Seyreklik
MovieLens100k	943	1682	100.002	1,2,3,4,5	18	%93,7
MovieLens1M	6040	3952	1.000.211	1,2,3,4,5	18	%95,8

3.2. Yapay Sinir Ağları (YSA)

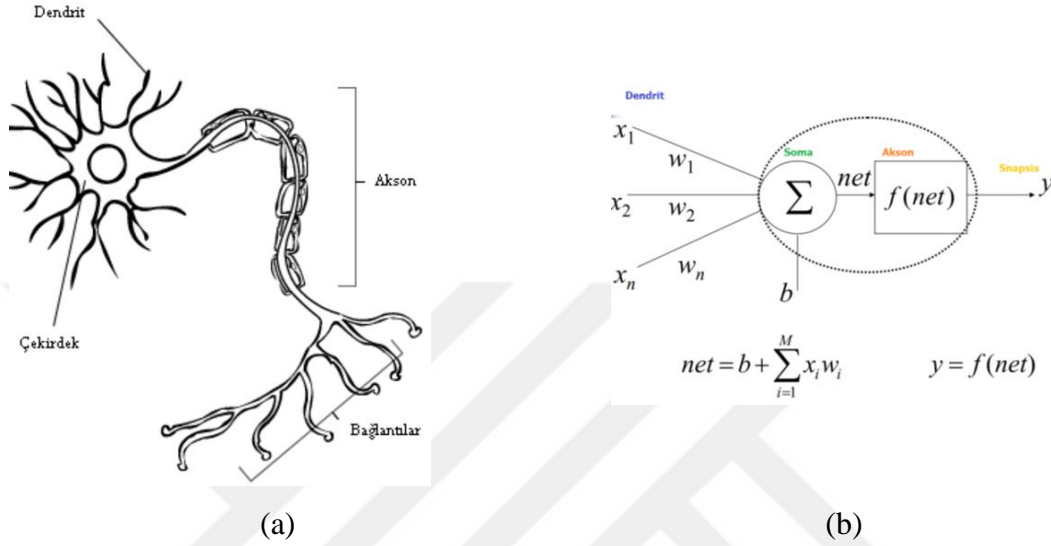
Yapay sinir ağları, insan sinir hücresinden esinlenilerek geliştirilmiş bir yapay öğrenme modelidir. Nöron/Algılayıcı denilen yapı taşlarından oluşmaktadır. YSA'nın amacı insanların biyolojik sinir sistemini taklit etmektir. Şekil 3.1’de Biyolojik bir sinir ağı hücresine ait temel kavramları aşağıdaki gibi ifade edilebilir (Simon, 2006).

Dendrit: Hücrelerden gelen sinyallerin çekirdeğe iletilmesini sağlar.

Hücre Gövdesi: Bir diğer ismi soma olan bu yapıyı dendrit üzerinden gelen sinyalleri toplayan merkezdir.

Akson: Toplanan bilgileri diğer hücelere dağıtan yapıdır.

Sinaps: Aksonlardan gelen bilgiyi ön işlemden geçirerek diğer hücelere iletir.



Şekil 3.1. (a)Biyolojik Sinir Hücresi (b) Yapay Sinir Ağı Hücresi

Yapay sinir hücreleri, matematiksel olarak, Denklem 3.1’de verildiği gibi ifade edilir: n nöron sayısını, w ağırlık değerini ve b bias değerini göstermektedir.

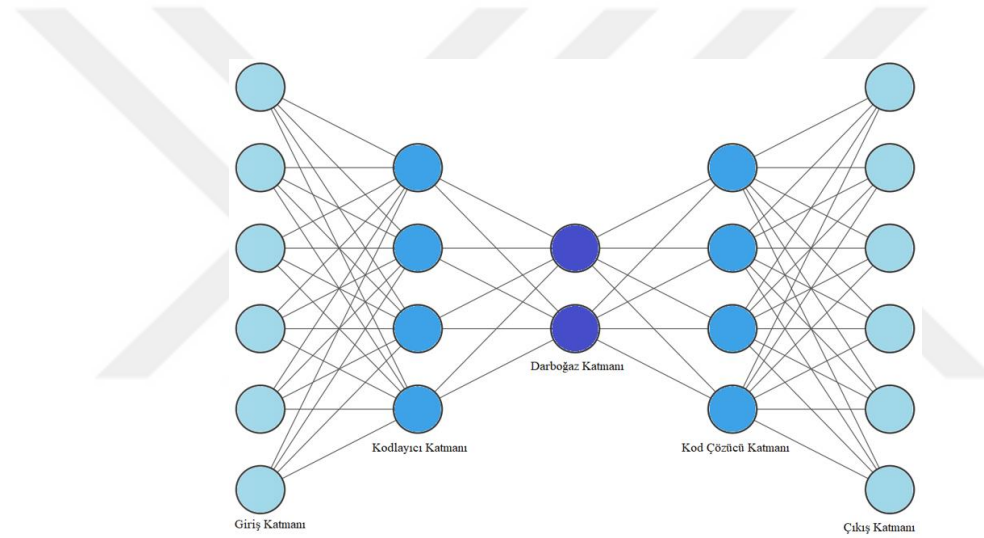
$$y(x) = f\left(\sum_{i=1}^n (w_i x_i + b)\right) \quad (3.1)$$

Yapılan işlem sonucunda aldıkları giriş değerlerinin ağırlıklandırılmış toplamı f aktivasyonundan geçtikten sonra çıkış değeri olarak sonraki yapay sinir hücresine iletirler. Birçok yapay sinir hücresi/nöron bir araya gelerek yapay sinir ağını oluştururlar.

3.3. Derin Otomatik Kodlayıcılar

Otomatik Kodlayıcılar denetimsiz bir makine öğrenmesi sistemidir (Doğan & Türkoğlu, 2019). Kodlayıcı (encoder) ve kod çözücü (decoder) olmak üzere iki bölümden oluşan bir Yapay Sinir Ağı (YSA) çeşididir (Bengio, 2018). “Kavramsal olarak basit olsa da makine öğrenmesinde önemli bir rol oynarlar.” (Baldi, 2012). Bir veya birden fazla gizli katmana sahip olabilir. Bu yapı birden fazla gizli katmana sahip ise Derin Otomatik

Kodlayıcı (DAE-Deep Autoencoders) ya da Diabolo Ağı olarak adlandırılır (Bengio, 2018). Birinin çıkışı diğerinin girişine verilecek şekilde birden fazla Otomatik Kodlayıcı kullanılmasına ise Yığınlanmış Otomatik Kodlayıcı (SAE-Stacked Autoencoder) adı verilir (TÜRK, 2019). Amacı boyutsallığın azaltılmasıyla giriş verisi için bir temsil (kodlama) öğrenilmesidir. Başka bir deyişle “giriş veri kümesini sıkıştırarak en az kayıpla en iyi öğrenmeyi amaçlar.” (Doğan & Türkoğlu, 2019). Bu işlem kodlayıcı(encoder) kısmında yapılır. Kullanılan sinir ağı ileri beslemelidir (Krizhevsky & Hinton, 2011). Çıkış katmanından giriştekine benzer veriler beklendiği için simetrik bir yapı kullanılır (Şekil.3.2). Bu yapı ilk olarak 2006 yılında Hinton ve Salakhutdinov tarafından ortaya konulmuştur (Hinton & Salakhutdinov, 2006). Girişte aldığı veriyi sıkıştırır ve çıkışta orijinal veriyi tekrar oluşturmaya çalışır (Aydın & Surer, 2020).



Şekil 3.2. Derin Otomatik Kodlayıcı Yapısı.

Bir Derin Otomatik Kodlayıcı, kodlayıcı ve kod çözücü olarak iki adet yapay sinir ağının dar boğaz katmanı ile birleşiminden meydana gelir (Kuchaiev & Ginsburg, 2017b). Dar boğaz katmanı giriş ve çıkış katmanından daha az sayıda nöron içerdiğinden veri kümesi bu katmanda sıkıştırılır. Derin Otomatik Kodlayıcılar anormallik tespitinde iyi sonuçlar vermektedir (Lyudchik, 2016). Otomatik Kodlayıcılar yeterli miktarda veri olmadığında veri çoğaltmada kullanılır (Kingma & Lei Ba, 2017). Denetimsiz bir öğrenme algoritması olması sebebiyle sınıf etiketine ihtiyaç duymaz. Bu sebeple tavsiye sistemlerinde kullanıcının derecelendirme tahminini elde etmek için uygun bir yöntemdir. Derin Otomatik Kodlayıcılara ait belirlenmesi gereken parametreler vardır. Bölümün devamında bu parametrelere ilişkin bilgiler verilmiştir. Önerilen yöntem için yapılan parametre analizine ait sonuçlar Bölüm 4.3’ de sunulmuştur.

3.3.1.Optimizasyon Algoritması

Optimizasyon algoritması, yapay sinir ağı temelli bir modelde, eğitim ve test kayıplarını en aza indirmek için sinir ağının ağırlık ve öğrenme hızı gibi özelliklerini güncelleyerek en uygun değerlerini bulmakta kullanılan algoritmadır. Bölümün devamında, sıklıkla kullanılan optimizasyon algoritmaları hakkında temel bilgiler sunulmuştur.

3.3.1.1.Adam

Değişken amaç fonksiyonlarının gradyan tabanlı optimizasyonu için önerilen bir algoritmadır (Denklem 3.2). Yöntemin bellek ihtiyacının az olması sebebiyle uygulanması basittir ve etkilidir. Yöntem gürültülü ve seyrek gradyanlı problemler içinde uygundur (Graves, 2014).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\frac{v_t}{1-\beta_2^t} + \epsilon}} \frac{m_t}{1-\beta_1^t} \quad (3.2)$$

3.3.1.2.RMSprop

Algoritma, öğrenme oranını eğimin değerine göre güncelleyerek eğimin kaybolmasını ve aşırı artmasını engeller (Denklem 3.3)(Zeiler, 2012).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (3.3)$$

3.3.1.3.Adadelta

Uyarlanabilir bir öğrenme metodu olan Adadelta (Kurt ve ark., 2018) RmsProp algoritması ile benzerdir (Denklem 3.4). Adagrad algoritmasının öğrenme oranını düşürerek belli bir zamandan sonra öğrenememe dezavantajını ortadan kaldırmıştır.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \quad (3.4)$$

“Matematiksel olarak RMSprop ile aynı görünse de bu algoritma her adımda öğrenme katsayısını güncellemez. Tüm öncelik gradyan değerlerini almak yerine bunu belirli bir sayıda kısıtlanmasındır.” (Duchi JDUCHI & Singer, 2011).

3.3.1.4. Adagrad

Adagrad algoritması (Clevert ve ark., 2016) öğrenme oranını giderek azaltır (Denklem 3.5).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i} \quad (3.5)$$

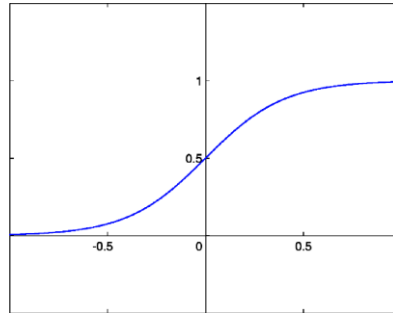
3.3.2. Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, bir sinir ağındaki nöronların, giriş değerlerine göre çıktısını belirlemek için kullanılacak matematiksel fonksiyondur. Bölümün devamında, sıklıkla kullanılan aktivasyon fonksiyonları hakkında temel bilgiler sunulmuştur.

3.3.2.1. Sigmoid

Sigmoid aktivasyon fonksiyonu (Şekil 3.3) sürekli ve türevi alınabilir bir fonksiyondur (Denklem 3.6). Yapay sinir ağlarında sıkça kullanılmasının sebebi doğrusal olmamasıdır.

$$\text{Sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (3.6)$$

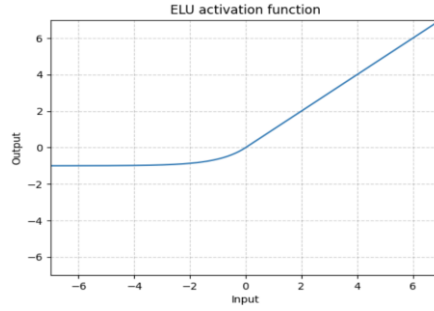


Şekil 3.3. Sigmoid Aktivasyon Fonksiyonu Grafiği

3.3.2.2. ELU (Exponential Linear Unit)

Negatif girdiler dışında ReLU aktivasyon fonksiyonu ile benzerdir (Denklem 3.7). Negatif girdilerde ise α parametresi alır (Şekil 3.4) (Barron, 2017).

$$ELU(x) = \{x, \text{if } x > 0 \quad \alpha * (\exp((x) - 1), \text{if } x \leq 0 \quad (3.7)$$

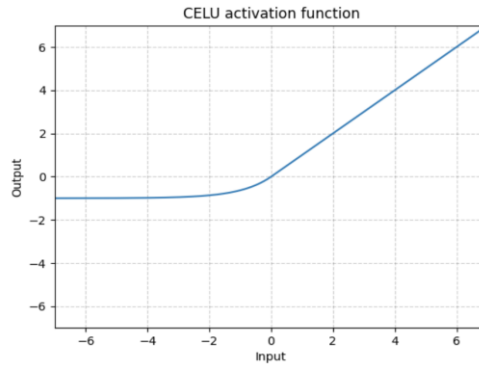


Şekil 3.4. ELU Aktivasyon Fonksiyonu Grafiği (URL 1)

3.3.2.3.CELU (Continuously Differentiable Exponential Linear Unit)

ELU aktivasyon fonksiyonunun α parametresi için sunulan alternatif parametrelendirme (Barron, 2017) ile CELU'da (Şekil 3.5.) doğrultma kolaylaştırılmıştır (Denklem 3.8).

$$CELU(x) = \max(0, x) + \min(0, \alpha * (\exp(x/a) - 1)) \quad (3.8)$$

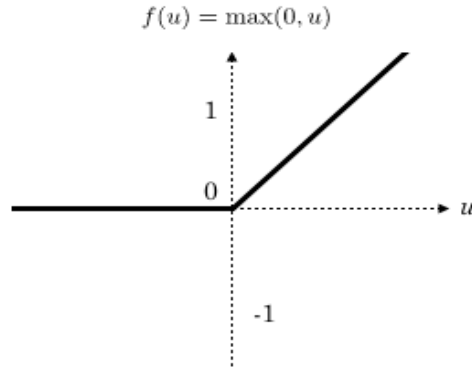


Şekil 3.5. CELU Aktivasyon Fonksiyonu Grafiği (URL 1)

3.3.2.4.ReLU (Rectified Linear Unit)

Doğrultulmuş doğrusal birim fonksiyonu 0'dan küçük giriş değerleri için 0 çıkış değerini verir (Şekil 3.6). Doğrusal olmaması hatalara kolayca geri-yayılım algoritmasını uygulamayı sağlar (Denklem 3.9). Bu nedenle en çok kullanılan aktivasyon fonksiyonlarından biridir.

$$\text{ReLU}(x) = (x^+) = \max(0, x) \quad (3.9)$$

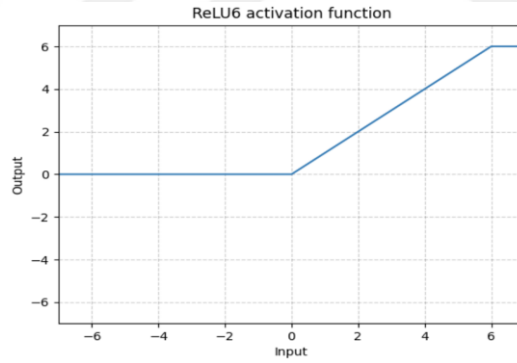


Şekil 3.6. ReLU Aktivasyon Fonksiyonunun Grafiği

3.3.2.5.ReLU6

Bu ReLU çeşidi 6 değerinde doymaya başlar (Şekil 3.7) (Denklem 3.10).

$$\text{ReLU6}(x) = \min(\max(0, x), 6) \quad (3.10)$$

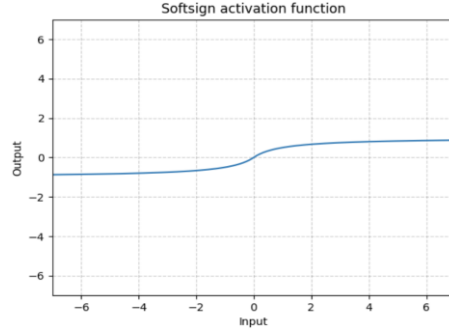


Şekil 3.7. ReLU6 Aktivasyon Fonksiyonu Grafiği (URL 1)

3.3.2.6.Softsign

Softsign aktivasyon fonksiyonu (Şekil 3.8) polinomsal olarak yakınsar (Denklem 3.11). Çok tercih edilmemesinin sebebi hesaplama maliyetidir.

$$\text{SoftSign}(x) = \frac{x}{1+|x|} \quad (3.11)$$

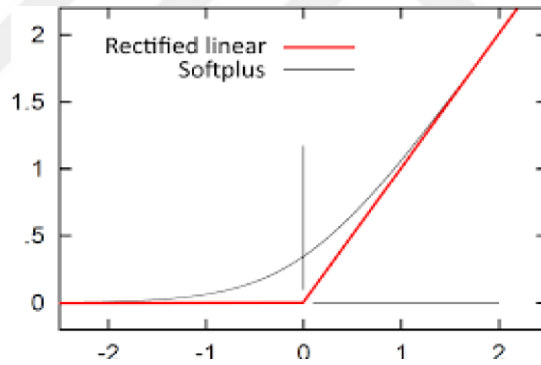


Şekil 3.8. Softsign Aktivasyon Fonksiyonu Grafiği (URL 1)

3.3.2.7. Softplus

SoftPlus, ReLU fonksiyonuna yumuşak bir yaklaşımdır (Şekil 3.5.) ve bir makinenin çıktısını her zaman pozitif olacak şekilde sınırlamak için kullanılabilir (Denklem 3.12).

$$\text{SoftPlus}(x) = \frac{1}{\beta} * \log(1 + \exp(\beta * x)) \quad (3.12)$$



Şekil 3.9. ReLU ve Softplus Aktivasyon Fonksiyonlarının Grafiği

3.3.3. Gizli Katman Sayısı

Gizli katman sayısı, giriş katmanına verilen verinin çıkış katmanına kadar transfer edilmesini ve derin öğrenmeyi sağlayan katmanların sayısıdır. Bu sayede doğrusal olmayan karmaşık problemlerin çözümü mümkün olmaktadır.

3.3.4. Öğrenme Adımı

Eğitim sırasında ağırlıklar güncellenirken kullanılan orandır. Öğrenme adımının küçük olması yavaş yakınsamaya, büyük olması ise yakınsayamamaya sebep olabilir.

3.3.5. Ağırlık Kaybı

En popüler düzenleme algoritmalarından biri, model parametrelerine L-2 cezası uygulamaktır, bu da parametrelerin bozulmasına neden olur ve buna ağırlık kaybı denir.

3.4. K-Kat Çapraz Doğrulama

Bu tez çalışmasında bilimsel çalışmalarda deneysel sonuçların güvenilirliğini artıran bir yöntem olan k-kat çapraz doğrulama kullanılmıştır. 5 kat çapraz doğrulama yapılırken veri kümesinin tamamı 5 parçaya ayrılır ve her adımda 4 parçası eğitim için 1 parçası test için kullanılır. Her adımda test parçası değişeceği için önerilen yöntem 5 farklı parça ile test edilmiş olur. Daha sonra bu parçaların ortalaması, performans değerlendirme değeri olarak sunulur.

3.5. Değerlendirme Kriteri

Bu çalışmada Ortalama Kare Hata (OKH- Mean Squared Error), Kök Ortalama Kare Hata (KOKH- Root Mean Squared Error) ve Ortalama Mutlak Hata (OMH- Mean Absolute Error) tahmin değerlendirme kriteri olarak kullanılmıştır (Denklem 3.13-3.16). Tavsiye değerlendirme kriteri olarak kesinlik, geri çağırma ve F1 skoru kullanılmıştır. Tavsiyelerin değerlendirilmesi için derin otomatik kodlayıcı eğitim kümesi ile eğitildikten sonra test kümesi üzerinde tahmin değerleri elde edilmiştir. Film değerlendirmesi 3,5 puanın üstünde olanlardan en iyi n tanesi kullanıcıya tavsiye olarak sunulmuştur. Ağın hassasiyetinden dolayı tahmin edilen puanlar tamsayı değildir. Ağın eğitiminden sonra elde edilen tahminler tam sayıya yuvarlanırsa 3,5 sayısı 4'e yuvarlanacağı için bu puanın üstünde değer alan filmler kullanıcıya tavsiye olarak sunulur.

3.5.1. Ortalama Kare Hata

OKH bir hata ölçüsü olması nedeniyle düşük sonuçlar yüksek performans anlamına gelir (Denklem 3.13). Ortalama kare hatası bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu gösterir ve her zaman pozitif değerlidir.

$$OKH = \frac{1}{n} \sum_{j=1}^n e_j^2 \quad (3.13)$$

Burada n toplam örnek sayısını, e kullanıcının verdiği gerçek oy ile sistemin hesapladığı tahmini oy değeri arasındaki farkı yani hatayı, j ise mevcut örneği ifade eder.

3.5.2. Kök Ortalama Kare Hata

KOKH, OKH hata ölçüsünün karekökünün alınmasıyla elde edilir (Denklem 3.14, 3.15). Artıklar, regresyon eğrisinin veri noktalarından ne kadar uzakta olduğunun bir ölçüsüdür; KOKH ise bu artıkların ne kadar yayıldığına bir ölçüsüdür. 0 ile ∞ arasında değer alır. 0 olması modelin hiç hata yapmadığı anlamına gelir.

$$KOKH = \sqrt{OKH} \quad (3.14)$$

$$KOKH = \sqrt{\frac{\sum_{j=1}^n e_j^2}{n}} \quad (3.15)$$

3.5.3. Ortalama Mutlak Hata

OMH, kullanıcının verdiği gerçek oy ile modelin hesapladığı tahmin oyu arasındaki farkın mutlak değeridir (Denklem 3.16).

$$OMH = \frac{1}{n} \sum_{j=1}^n |e_j| \quad (3.16)$$

3.5.4. Kesinlik

Kesinlik (Precision) değeri, üretilen en iyi n adet film tavsiyesi ile test veri kümesinde kullanıcıdan 4 puanın üstünde oy almış filmlerin kümesinin eleman sayısını, n (Tavsiye edilen filmlerin sayısına) sayısına oranlayarak hesaplanır (Denklem 3.17).

$$Kesinlik = \frac{|Tavsiye edilen filmler \cap Gerçek filmler|}{Tavsiye edilen filmler} \quad (3.17)$$

3.5.5. Geri Çağırma

Geri çağırma (Recall) değeri, üretilen en iyi n film tavsiyesi ile test veri kümesinde kullanıcıdan 4 puanın üstünde oy almış filmlerin kesişim kümesinin eleman sayısını, test kümesinde kullanıcıdan 4 puanın üzerinde oy alan filmlerin sayısına (*Gerçek filmler*) bölerek hesaplanır (Denklem 3.18).

$$\text{Geri Çağırma} = \frac{|\text{Tavsiye edilen filmler} \cap \text{Gerçek filmler}|}{\text{Gerçek filmler}} \quad (3.18)$$

3.5.6. F1 skoru

F1 skoru, hesaplanan kesinlik ve geri çağırma değerlerinin harmonik ortalaması alınarak elde edilir (Denklem 3.19).

$$F1 \text{ skoru} = \frac{2 \times \text{Kesinlik} \times \text{Geri Çağırma}}{\text{Kesinlik} + \text{Geri Çağırma}} \quad (3.19)$$

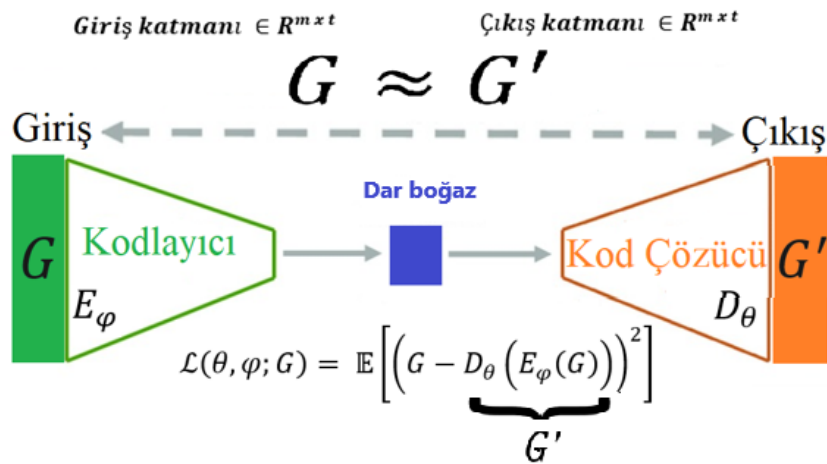
3.6. Geliştirme Ortamı

Bu tezin uygulaması Spyder bütünleşik geliştirme ortamında Python programlama dili kullanılarak gerçekleştirilmiştir. Otomatik Kodlayıcı yapısı için Pytorch kütüphanesi kullanılmıştır. Hesaplama gücü için kullanılan bilgisayar, intel i7 10750H işlemci, RTX 2060 ekran kartı ve 32 GB belleğe sahiptir.

4.TEZDE ÖNERİLEN DERİN OTOMATİK KODLAYICI TABANLI TAVSİYE SİSTEMİ (DAERec)

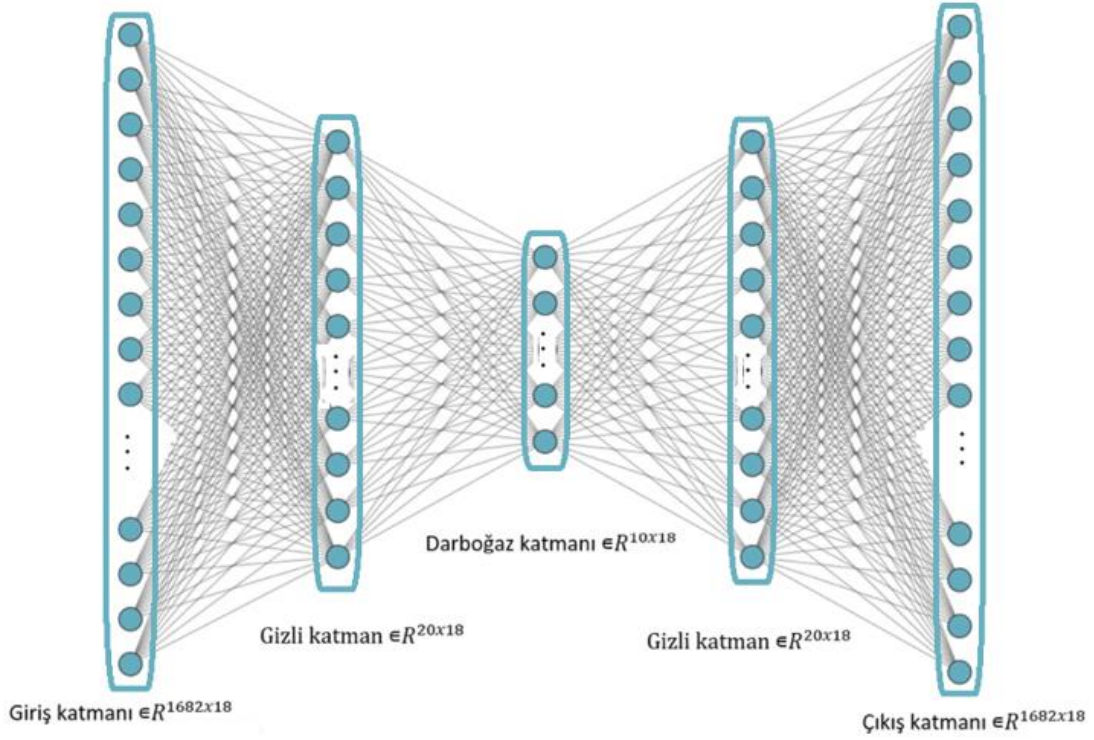
DAERec, içerisinde öğelere ait tür bilgisi de bulunan veri kümeleri için önerilmiş ve parametre analizleri gerçekleştirilmiş yeni bir Derin Otomatik Kodlayıcı (DAE) modelidir. Tavsiye sistemlerinde kullanılan DAE mimarilerinin girişi, ürüne (tavsiye sistemleri için filmlere) verilen oyları içeren tek boyutlu bir vektör şeklinde olmaktadır. Ancak bilgisayarlı görme alanında, 2 boyutlu resim bilgilerini işleyen başarılı DAE mimarileri bulunmaktadır. Özellikle gürültülü/eksik resimler için bu model tercih edilmekte resmin daha net ve anlamlı şeklinin elde edilmesi için kullanılmaktadır (Ashgar M.H.,2020). Buradan ilham alınarak, ürünlerin oylarını içeren vektöre, her bir ürüne ait tür bilgisi de eklenerek iki boyutlu bir giriş matrisi elde edilmiş ve bu matrisi giriş olarak kullanan yeni bir Derin Otomatik Kodlayıcı Tabanlı Tavsiye Sistemi (DAERec) mimarisi önerilmiştir. Bu sayede veri eksikliği probleminin çözümüne katkı sağlanmış, tür ve oy bilgisi birlikte işlendiği için tavsiye başarısı yükseltilmiştir. Deneysel çalışmaların sonucunda, ağın eğitiminin uzun sürelerle gerek kalmadan tamamlandığı görülmektedir. Ağın kısa sürede eğitimi ölçeklenebilirlik probleminin çözümüne katkı sağlamaktadır. Bilgimiz dahilinde iki boyutlu giriş alan derin otomatik kodlayıcı tabanlı bir tavsiye sistemi literatürde bulunmamaktadır.

Önerilen derin ağ mimarisi simetriktir ve 5 katmandan oluşmaktadır. Bunlar giriş, kodlayıcı, dar boğaz, kod çözücü ve çıkış katmanlarıdır. Giriş katmanına $m \times t$ boyutlu film-tür matrisi uygulanmıştır. Film sayısı m , tür sayısı t ile gösterilmiştir. Otomatik kodlayıcı yapısı gereği çıkış katmanı, giriş katmanı ile simetrik yapıda bulunmaktadır.



Şekil 4.1. Önerilen Mimarinin Temel Blok Diyagramı

Şekil 4.1’de önerilen mimarinin blok diyagramı verilmiştir. Giriş katmanına verilen *Tür-Oy Matrisi* G ile, çıkışta kullanıcıya ait oy tahminlerini içeren *Tahmin Matrisi* G' ile gösterilmiştir. Eğitim sonucunda çıkış katmanında elde edilen bu G' içindeki değerler oy tahmini ve tavsiye üretmede kullanılmaktadır.



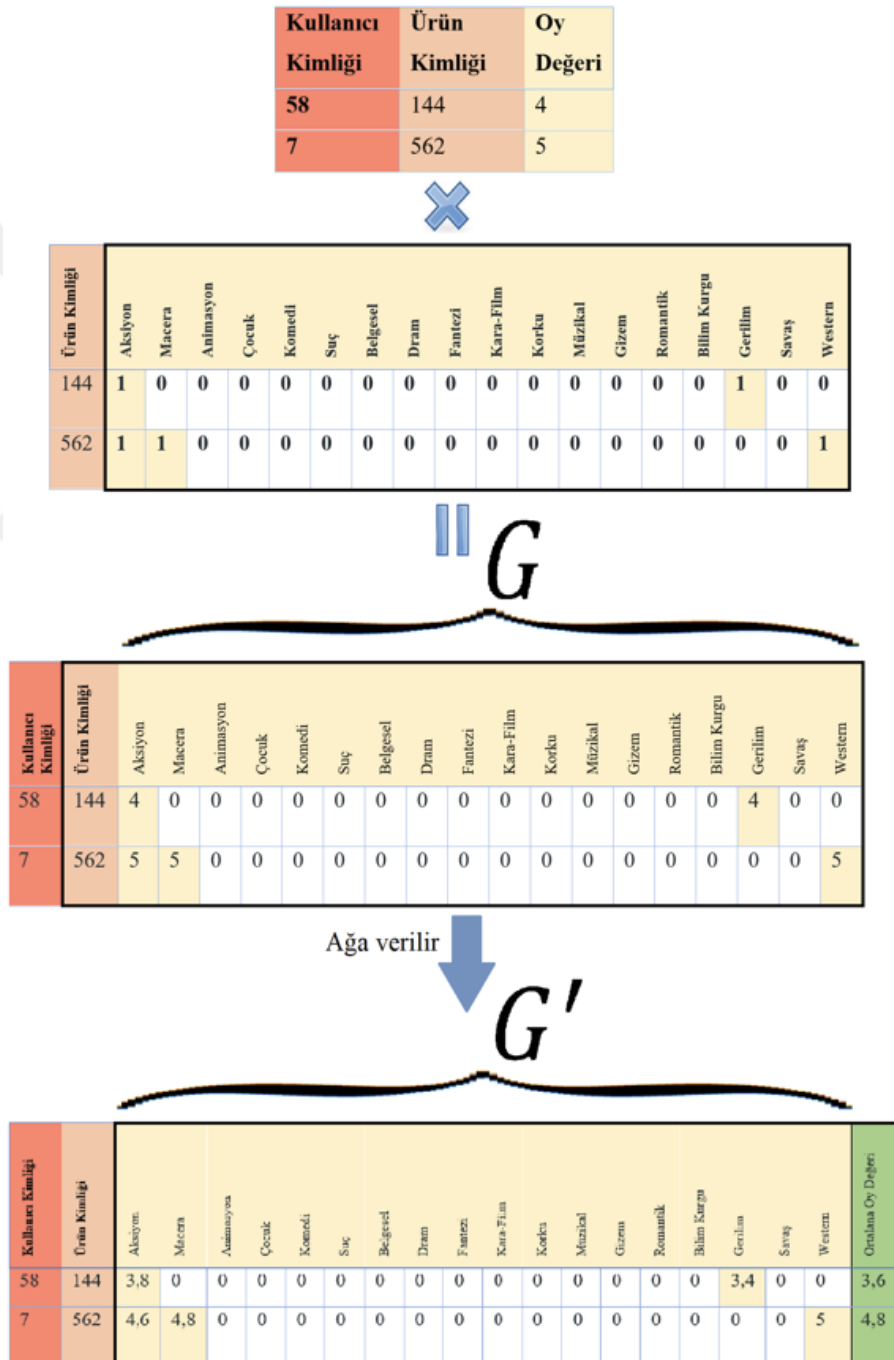
Şekil 4.2. MovieLens100K için önerilen DAERec ağı mimarisi

Örnek olarak Movielens100k veri kümesi için önerilen DAERec’e ait ağı mimarisi Şekil 4.2’de verilmiştir. Movielens100k kümesinde 1682 film ve 18 adet film türü (aksiyon, komedi, korku, vb.) bulunmaktadır. Ağ, simetrik olduğu için giriş ve çıkış katmanlarının boyutu aynıdır ve 1682×18 boyutlu bir matristir. Gizli katmanlar 20×18 ve darboğaz katmanı 10×18 boyuttan oluşmaktadır. Gizli katman sayısı ve her bir katmanda yer alacak nöron sayıları yapılan parametre analizleri sonucunda belirlenmiştir. Yapılan parametre analizlerinin ayrıntılı sonuçları Bölüm 4.3’te sunulmuştur.

4.1. DAERec için İki Boyutlu Tür-Oy Matrisinin Oluşturulması ve Ağı Eğitimi

Bu tez çalışmasında kullanılan otomatik kodlayıcı mimarisine filmlerin tür bilgileri de verilmiş ve ağın hem kullanıcının oy verme özelliklerini, hem de filmlerin tür özelliklerini öğrenerek tahmin üretmesi amaçlanmıştır.

Veri kümesinde macera, animasyon, çocuk, komedi, suç, belgesel, dram, fantezi, kara-film, korku, müzikal, gizem, romantik, bilim kurgu, gerilim, savaş ve western olmak üzere 18 tür bulunmaktadır. Öncelikle her bir film için 18 bitlik bir vektör oluşturulmuş ve filmin ait olduğu türe karşılık gelen bitlere 1, diğerlerine 0 yazılmıştır. Devamında kullanıcının bu filme verdiği oy ile vektör çarpılarak, 1 yazan bitlerin değeri oy bilgisi ile değiştirilmesi sağlanmıştır. Bu şekilde her bir kullanıcıya ait *Tür-Oy Matrisi* G oluşturulmuştur. Konuya ilişkin sayısal bir örnek Şekil 4.3’de verilmiştir.



Şekil 4.3. Tür-Oy Matrisi G 'nin oluşturulması ile ilgili sayısal bir örnek

58 numaralı kullanıcının 144 numaralı film için verdiği oy 4'tür. 144 numaralı filmin türü aksiyon ve gerilimdir. Tür bilgilerini gösteren 0 ve 1 sayıları nokta çarpım şeklinde kullanıcının verdiği oy değeri ile çarpılınca G tür-oy matrisi elde edilir. Yani G matrisi hem filmin aksiyon ve gerilim türünde olduğunu hem de kullanıcının bu filme 4 puan verdiği bilgisini tutmaktadır. G tür-oy matrisi ağın giriş katmanına verilerek çıkışta G' elde edilir. G' matrisinde her tür için farklı bir oy değeri tahmin edilmiştir. Ağın çıkışında 58 numaralı kullanıcı için 144 numaralı filmin aksiyon türüne 3,8, gerilim türüne 3,4 puan vereceği hesaplanmıştır. G' matrisindeki tahmin değerlerini literatürdeki çalışmalarla karşılaştırabilmek için filmin ait olduğu tür sayısına göre ortalama alınarak tek bir oy değeri elde edilmiştir (Algoritma 2). Ortalama oy değeri, 3,8 ve 3,4 değerlerinin aritmetik ortalaması alınarak 3,6 bulunmuştur. Kullanıcının gerçek oy değeri 4 olduğu için bu film için hesaplanan OKH (Denklemler 3.13) değeri 0,16'dır.

4.2 DAERec ile Tahmin Hesaplama ve Tavsiye Üretme

Önerilen yöntemi eğiten algoritmanın sözde kodu Çizelge 4.1'de verilmiştir. Algoritmada öncelikle DAERec modeli başlatılır. Belirlenen çevrim sayısı boyunca, her bir kullanıcıya ait G_k matrisi algoritmanın girişine uygulanır. Çıkışında G'_k matrisi elde edilir. İki matris arasındaki hata değeri OKH kullanılarak hesaplanır ve ağırlıklar geriye doğru yayılır.

Çizelge 4.1: DAERec Ağının Eğitim için kullanılan algoritmanın sözde kodu

Algoritma 1 : DAERec Eğitim Algoritması sözde kodu	
<hr/>	
Giriş :	
G	: Film-Tür matrisi, $G \in R^{m \times t}$
k	: Kullanıcı sayısı
m	: Ürün sayısı
t	: Tür sayısı
Çıkış:	
G'	: Tahmin matrisi, $G' \in R^{m \times t}$
<hr/>	
begin	
	$model=DAERec()$ //önerilen ağ yapısı başlatıldı
	$j=0;$
	//Ağın eğitilmesi
	while $j < \text{çevrim_sayısı}$
	for kullanıcı_id=1 to k do // her bir kullanıcı için
	$G'_k = model(G_k);$ //G film-tür matrisini ağın girişine vererek G' elde et
	$\mathcal{L} = OKH(G_k, G'_k);$
	\mathcal{L} hatasını geriye yay;
	Ağırlıkları güncelle ;
	end for
	$j++;$
	end while
end begin	

Tahmin Hesaplama ve Tavsiye Üretmek için kullanılan Algoritma-2'nin sözde kodu Çizelge 4.2'de verilmiştir. Giriş olarak G' matrisini alan algoritma, çıkış olarak tavsiye edilecek n film ve bu filmler için hesaplan tahmin değerlerini döndürür. Bu değerler, algoritmanın başarısını ölçen metriklerin hesaplanmasında kullanılacaktır.

Çizelge 4.2: DAERec ile Tahmin Hesaplama ve Tavsiye Üretmek için kullanılan algoritmanın sözde kodu

Algoritma 2 : DAERec ile Tahmin Hesaplama ve Tavsiye Üretme	
Giriş :	
G'	: Tahmin matrisi, $G' \in R^{m \times t}$
k	: Kullanıcı sayısı
m	: Ürün sayısı
t	: Tür sayısı
n	: Tavsiye olarak döndürülecek film sayısı
Çıkış:	
	Veri kümesindeki filmler için üretilen oy tahminleri
	Tavsiye edilecek n film bu filmler için hesaplan tahmin değerleri
<hr/>	
begin	
	for kullanıcı_id=1 to k do // her bir kullanıcı için
	türler=Filmin dahil olduğu türlere karşılık gelen indeks değerleri;
	hesaplananTahmin[kullanıcı_id]=ortalama_hesapla(G' [film _{id} , türler]);
	// G' [film _{id} , türler]) : filmin ait olduğu türler için ağın ürettiği oy değerleri
	if kullanıcı_id'nin test kümesinde n 'den fazla 4'ün üzerinde oy almış filmi varsa
	önerilecek_filmler = G [G' [film _{id} , türler] \geq 3.5];
	sıralı_filmler=sort(önerilecek_filmler) //büyükten küçüğe sıralanır;
	tavsiyeler[kullanıcı_id]=sıralı_filmler ilk n elemanını tavsiye olarak ata;
	else
	Devam et;
	end if
	end for
	return hesaplananTahmin, tavsiyeler
end begin	

Algoritma 2'ye göre veri kümesindeki her bir kullanıcı sırası ile eğitimi tamamlanmış DAERec ağına verilir, çıkışı hesaplanır. İlgili kullanıcının test kümesinde bulunan ve 4 ve üzerinde oy puanı alan filmler bulunur. Eğer bu filmlerin sayısı n 'den fazla ise, DAERec ağının çıkışında ilgili filmlere verdiği oylara bakılır ve 3,5 üzerinde oy alanlar, aldığı oya göre büyükten küçüğe göre sıralanır ve ilk n film tavsiye olarak döndürülür. Test kümesinde oylar tam sayı iken, ağın çıkışındaki oylar reel sayıdır. Bu yüzden 3,5 ve üzerinde oy alanlar 4 ve 4'ün üzerinde oy almış olarak düşünülmüştür. Tahmin hesaplamada, her bir kullanıcı için ağın çıkışındaki Tahmin Matrisindeki, tahmin üretilen filmin türlerine karşılık gelen değerlerin ortalaması alınır ve ilgili filme kullanıcının vereceği tahmini oy olarak kullanılır. Şekil 4.3'ün en son bloğunda, bu işlem için sayısal bir örnek gösterilmiştir.

4.3.DAERec Parametre Analizleri ve Elde Edilen Uygulama Sonuçları

Bu çalışmada model tabanlı bir tavsiye sistemi gerçekleştirilmiştir. Model olarak derin otomatik kodlayıcı ağ yapısı kullanılmıştır. Modele ait hiperparametrelerin uygun değerlerinin araştırılması, önerilen tavsiye sistemlerinin başarısı için önemlidir.

Bu tez çalışması için;

- Ara Katman Sayısı,
- Ara Katmanlardaki Nöron Sayıları,
- Aktivasyon Fonksiyonu,
- Optimizasyon Algoritması,
- Öğrenme Adımı,
- Ağırlık Kaybı

parametrelerinin değerleri yapılan deneysel çalışmalar sonucunda belirlenmiştir. Parametre analizi iki aşamada yapılmıştır. İlk aşamada, ağın girişine sadece oylar verilmiş, film tür bilgileri ağa verilmemiştir. Tek boyutlu giriş ile elde edilen deneysel çalışma sonuçları bildiri olarak ICCEES 2020 uluslararası konferansında sunulmuştur (Ünaldı S. S. ve ark. ,2020). İkinci aşamada, Bölüm 4.1’de anlatılan Tür-Oy matrisi ağa verilmiş, bu ağ için uygun parametre değerleri araştırılmıştır. Her iki mimari için de, araştırma konusu olan parametrelerin en uygun değerleri Movielens100k veri kümesi üzerinde yapılan deneyler ile elde edilmiştir. Analiz için kullanılan parametre değer aralıkları, daha önce yapılmış analiz çalışmalardan elde edilmiştir (Kantepe & Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı DANIŞMAN Hasan ERDAL, 2019; Tran ve ark., 2018).

4.3.1. Tek Boyutlu Giriş için Yapılan Hiperparametre Analizi Sonuçları

Movielens100k veri kümesi %80 eğitim ve %20 test olarak bölünmüştür. Tüm analizler aynı eğitim ve test kümesi üzerinde gerçekleştirilmiş ve her iki kümeye de ait OKH değerleri çizelgelerde verilmiştir.

İlk analiz, optimizasyon algoritması ve aktivasyon fonksiyonu seçimi için gerçekleştirilmiştir. Çevrim sayısı=200, Öğrenme Oranı=0,001, Ağırlık Kaybı=0,01 Gizli Katman Sayısı=2, Katmanlardaki Nöron Sayıları=(64,4) iken, Adam, Rmsprop, Adagrad ve Adadelta olarak 4 optimizasyon algoritması ve Sigmoid, ELU, CELU, ReLU, ReLU6, Softsign ve Softplus olmak üzere 7 aktivasyon fonksiyonuna ait sonuçlar Çizelge 4.3’de raporlanmıştır.

Sonuçlar incelendiğinde, test kümesi için en düşük OKH 0,9638 ile Adagrad-CELU ikilisinde, ikinci en düşük OKH 0,9789 ile değer ise Adagrad-ReLU ikilisinde elde edilmiştir. Çevrim sayısı boyunca, Adagrad-CELU ikilisi için OKH değeri 0,9638 – 38,0179 aralığında değer alırken, Adagrad-ReLU 0,9789 – 1,0209 aralığında değer almaktadır. Hata aralığı daha düşük olduğu için aktivasyon fonksiyonu olarak ReLU, optimizasyon algoritması için Adagrad seçilmiştir.

Çizelge 4.3. Optimizasyon Algoritması ve Aktivasyon Fonksiyonu Seçimi, Çevrim sayısı=200, Öğrenme Oranı=0,001, Ağırlık Kaybı=0,01 Gizli Katman Sayısı=2, Katmanlardaki Nöron Sayıları=(64,4)

Optimizasyon Algoritması	Aktivasyon Fonksiyonu	Eğitim için OKH	Test için OKH
Adam	Sigmoid	1,0232	1,0128
Adam	ELU	16,2027	16,6900
Adam	CELU	17,9262	16,6365
Adam	ReLU	1,0226	1,0115
Adam	ReLU6	1,0215	1,0154
Adam	Softsign	16,2150	17,1571
Adam	Softplus	1,0214	1,0270
RMSprop	Sigmoid	1,0779	1,0528
RMSprop	ELU	4,9814	4,5693
RMSprop	CELU	4,8802	4,6728
RMSprop	ReLU	1,0209	1,0173
RMSprop	ReLU6	1,0188	1,0192
RMSprop	Softsign	5,0276	4,5369
RMSprop	Softplus	1,0239	1,0119
Adadelta	Sigmoid	1,1804	1,1676
Adadelta	ELU	1,5649	1,5583
Adadelta	CELU	2,2683	2,2619
Adadelta	ReLU	1,1145	1,1385
Adadelta	ReLU6	1,4453	1,4149
Adadelta	Softsign	1,8291	1,7851
Adadelta	Softplus	1,0443	1,0624
Adagrad	Sigmoid	0,9922	1,0095
Adagrad	ELU	0,8707	0,9956
Adagrad	CELU	0,8905	0,9638
Adagrad	ReLU	0,9376	0,9789
Adagrad	ReLU6	0,9927	1,0101
Adagrad	Softsign	1,0042	1,0265
Adagrad	Softplus	0,9916	1,0073

Daha sonra uygun katman sayısı araştırılmıştır (Çizelge 4.4). Otomatik Kodlayıcılar simetrik bir yapıya sahip olduğu için kod çözücü kısmındaki katman ve nöron sayıları Çizelgelerde verilmemiştir. Tüm çizelgeler için, gizli katmandaki en son nöron sayısı, dar boğaz katmanındaki nöron sayısını göstermektedir. Çizelge 4.4 incelendiğinde, test kümesi için en düşük OKH'nın **0,9626** ile ReLU aktivasyon fonksiyonunda, gizli katmandaki nöron sayıları 20 (kodlayıcı gizli katmanındaki nöron sayısı)-10 (dar boğaz gizli katmanındaki nöron sayısı) -20 (kod çözücü gizli katmanındaki nöron sayısı, ağ yapısı simetrik olduğu için çizelgelerde bu katman gösterilmemiştir) iken elde edilmiştir.

Çizelge 4.4. Katman Sayısı Belirleme Deneysel Sonuçları, Optimizasyon Algoritması = Adagrad, Öğrenme Adımı=0,001 Ağırlık Kaybı=0,01, Çevrim Sayısı=200)

Gizli Katman Sayısı(Kodlayıcı+ Darboğaz)	Sırası ile Gizli Katmandaki Nöron Sayıları	Aktivasyon Fonksiyonu	Eğitim için OKH	Test için OKH
2	64 - 4	ReLU	0,9376	0,9789
2	128 - 64	ReLU	0,8672	1,0209
2	80 - 40	ReLU	0,9346	1,0179
2	20- 10	ReLU	0,9044	0,9626
3	256 - 64 - 16	ReLU	0,9935	0,9944
3	128 - 32 - 8	ReLU	0,9739	1,0060
3	64 - 16 - 4	ReLU	0,9834	1,0042
3	40 - 20 - 10	ReLU	0,8977	0,9740
4	256 - 64 - 16 - 4	ReLU	0,9893	0,9986
4	1024 - 128 - 16 - 2	ReLU	0,8948	0,9938
4	80 - 40 - 20 - 10	ReLU	0,9951	1,0056
2	64 - 4	CELU	0,8905	0,9638
2	128 - 64	CELU	1,0066	1,0269
2	80 - 40	CELU	0,8531	0,9923
2	20- 10	CELU	0,9175	0,9706
3	256 - 64 - 16	CELU	1,0903	1,0839
3	128 - 32 - 8	CELU	1,0174	1,0506
3	64 - 16 - 4	CELU	0,8714	0,9880
3	40 - 20 - 10	CELU	0,8244	1,0072
4	256 - 64 - 16 - 4	CELU	0,9977	1,0466
4	1024 - 128 - 16 - 2	CELU	1,4456	1,4445
4	80 - 40 - 20 - 10	CELU	42,256	38,0179

Çizelge 4.5. Öğrenme Adımı Belirleme Deneysel Sonuçları, Optimizasyon Algoritması=Adagrad, Aktivasyon Fonksiyonu=ReLU, Ağırlık Kaybı=0,01, Çevrim Sayısı=200

Gizli Katman Sayısı (Kodlayıcı+ Darboğaz)	Sırası ile Gizli Katmandaki Nöron Sayıları	Öğrenme Adımı	Eğitim için OKH	Test için OKH
2	64 - 4	0,001	0,9376	0,9789
2	20 - 10	0,001	0,9044	0,9626
3	64 - 16 - 4	0,001	0,9834	1,0042
3	40 - 20 - 10	0,001	0,8977	0,9740
2	64 - 4	0,005	0,9921	1,0185
2	20 - 10	0,005	0,9970	1,0115
3	64 - 16 - 4	0,005	0,9949	1,0018
3	40 - 20 - 10	0,005	1,0046	1,0172
2	64 - 4	0,01	0,9921	1,0052
2	20 - 10	0,01	0,9998	1,0080
3	64 - 16 - 4	0,01	0,9942	1,0026
3	40 - 20 - 10	0,01	0,9968	1,0051
2	64 - 4	0,05	0,9947	1,0081
2	20 - 10	0,05	0,9937	1,0089
3	64 - 16 - 4	0,05	0,9953	1,0059
3	40 - 20 - 10	0,05	0,9994	0,9902

Adagrad optimizasyon algoritması ve ReLU aktivasyon fonksiyonu için Öğrenme Adımı ve Ağırlık Kaybı parametre değerleri de araştırılmış, sonuçlar sırası ile Çizelge 4.5 ve Çizelge 4.6’te sunulmuştur. En iyi öğrenme adımının tespiti için 4 farklı değer araştırılmış, en düşük OKH’lar 0,001 grubunda elde edilmiştir (Çizelge 4.5).

Çizelge 4.6. Ağırlık Kaybı Belirleme, Optimizasyon Algoritması=Adagrad, Aktivasyon Fonksiyonu=ReLU, Öğrenme Adımı=0,01, Çevrim Sayısı=200

Gizli Katman Sayısı (Kodlayıcı+ Darboğaz)	Sırası ile Gizli Katmandaki Nöron Sayıları	Ağırlık Kaybı	Eğitim için OKH	Test için OKH
2	64 - 4	0,001	0,9769	1,0210
2	20 - 10	0,001	0,8735	0,9634
3	64 -16 - 4	0,001	0,9778	1,0062
3	40 -20 - 10	0,001	0,9033	0,9659
2	64 - 4	0,005	0,8864	0,9962
2	20 - 10	0,005	0,9461	0,9925
3	64 -16 - 4	0,005	0,9157	1,0259
3	40 -20 - 10	0,005	0,9316	0,9834
2	64 - 4	0,01	0,9376	0,9789
2	20 - 10	0,01	0,9044	0,9626
3	64 -16 - 4	0,01	0,9834	1,0042
3	40 -20 - 10	0,01	0,8977	0,9740
2	64 - 4	0	0,9049	1,0126
2	20 - 10	0	0,9046	0,9671
3	64 -16 - 4	0	0,9875	1,0199
3	40 -20 - 10	0	0,9675	1,0044

En iyi ağırlık kaybının tespiti için 4 farklı değer araştırılmış ve en uygun değer 0,01 olarak bulunmuştur (Çizelge 4.6).

4.3.2 İki Boyutlu Giriş (Tür-Oy Matrisi) için Yapılan Hiperparametre Analizi Sonuçları

Önerilen DAERec mimarisi için Bölüm 4.3.1’de yapılan parametre analizleri tekrarlanmıştır. İlk aşama olan optimizasyon algoritması-aktivasyon fonksiyonu ikilisinin tespitinde en iyi eğitim ve en iyi test sonuçları seçilmiştir. Test kümesi için OKH değerleri incelendiğinde, Adagrad-ReLU ikilisi 0,2337 ile en düşük hataya sahip olduğu görülmüştür (Çizelge 4.7).

Çizelge 4.8’de katman ve katmandaki en uygun nöron sayısı değerleri, optimizasyon algoritması-aktivasyon fonksiyonu ikililerine göre araştırılmıştır. İlk gizli katmanda 64, ikinci ve darboğaz olarak adlandırılan katmanda 4 nöron bulunan ağda 0.2337 değeri ile en düşük hata elde edilmiştir. En düşük hatayı elde eden 2 ve 3 gizli katmanlı yapılar bir sonraki adımda öğrenme adımı analizi yapmak için kullanılacaktır. Daha düşük hatalar elde etmesine rağmen 4 gizli katmanlı yapının kullanılmamasının sebebi ağın eğitim süresini artırması fakat hata düşüşünde önemli bir fark sağlamamasıdır. En düşük hatalar Adagrad-ReLU ikilisinde elde edildiği için sonraki deneylerde bu ikili kullanılacaktır. Bu çizelge Çizelge 4.7 elde edilen bulguyu desteklemektedir.

Çizelge 4.7. Optimizasyon Algoritması ve Aktivasyon Fonksiyonu Seçimi, Çevrim sayısı=200, Öğrenme Oranı=0,001, Ağırlık Kaybı=0,01 Gizli Katman Sayısı=2, Katmanlardaki Nöron Sayıları=(64,4)

Optimizasyon Algoritması	Aktivasyon Fonksiyonu	Eğitim için OKH	Test için OKH
Adam	Sigmoid	0,2393	0,2383
Adam	ELU	6,3307	8,2404
Adam	CELU	2,3816	2,2928
Adam	ReLU	0,2387	0,2362
Adam	ReLU6	0,2445	0,2454
Adam	Softsign	1,8778	1,8889
Adam	Softplus	0,2315	0,2371
RMSprop	Sigmoid	0,2036	0,2451
RMSprop	ELU	0,2493	0,2493
RMSprop	CELU	0,3025	0,2967
RMSprop	ReLU	0,2398	0,2372
RMSprop	ReLU6	0,2504	0,2391
RMSprop	Softsign	0,1968	0,2844
RMSprop	Softplus	0,2252	0,2379
Adadelta	Sigmoid	0,2667	0,2723
Adadelta	ELU	0,4105	0,4087
Adadelta	CELU	0,4034	0,4536
Adadelta	ReLU	0,2741	0,2676
Adadelta	ReLU6	0,3563	0,3547
Adadelta	Softsign	0,3226	0,3141
Adadelta	Softplus	0,2577	0,2536
Adagrad	Sigmoid	0,2424	0,2422
Adagrad	ELU	0,2016	0,2660
Adagrad	CELU	0,1978	0,2649
Adagrad	ReLU	0,2128	0,2337
Adagrad	ReLU6	0,2336	0,2356
Adagrad	Softsign	0,2278	0,2630
Adagrad	Softplus	0,2290	0,2369

Çizelge 4.8. Katman Sayısı Belirleme, Optimizasyon Algoritması = Adagrad, Öğrenme Adımı=0,001 Ağırlık Kaybı=0,01 , Çevrim Sayısı=200

Gizli Katman Sayısı (Kodlayıcı+Darboğaz)	Sırası ile Gizli Katman.Nöron Sayıları	Akt. Fonk.-Opt Alg	Eğitim OKH	Test OKH
2	64 - 4	Softsign-RMSprop	0,1968	0,2844
2	128 - 64	Softsign-RMSprop	0,0844	0,4545
2	80 - 40	Softsign-RMSprop	0,1167	0,4019
2	20- 10	Softsign-RMSprop	0,2034	0,2699
3	256 - 64 - 16	Softsign-RMSprop	0,1764	0,2910
3	128 - 32 - 8	Softsign-RMSprop	0,1971	0,2647
3	64 - 16 - 4	Softsign-RMSprop	0,1933	0,2676
3	40 - 20 - 10	Softsign-RMSprop	0,2070	0,2671
4	256 - 64 - 16 - 4	Softsign-RMSprop	0,2342	0,2589
4	1024 - 128 - 16 - 2	Softsign-RMSprop	0,2324	0,2533
4	80 - 40 - 20 - 10	Softsign-RMSprop	0,2378	0,2484
2	64 - 4	ReLU-Adagrad	0,2128	0,2337
2	128 - 64	ReLU-Adagrad	0,1905	0,2528
2	80 - 40	ReLU-Adagrad	0,2111	0,2344
2	20- 10	ReLU-Adagrad	0,2166	0,2343
3	256 - 64 - 16	ReLU-Adagrad	0,2124	0,2461
3	128 - 32 - 8	ReLU-Adagrad	0,2156	0,2345
3	64 - 16 - 4	ReLU-Adagrad	0,2332	0,2356
3	40 - 20 - 10	ReLU-Adagrad	0,2333	0,2351
4	256 - 64 - 16 - 4	ReLU-Adagrad	0,2334	0,2350
4	1024 - 128 - 16 - 2	ReLU-Adagrad	0,2340	0,2352
4	80 - 40 - 20 - 10	ReLU-Adagrad	0,2188	0,2350

Öğrenme adımı deneyinde dört farklı değer kullanılmıştır. En düşük test hatası 0,2323 olup 0,001 öğrenme adımı ile elde edilmiştir.

Çizelge 4.9. Öğrenme Adımı Belirleme, Optimizasyon Algoritması=Adagrad, Aktivasyon Fonksiyonu=ReLU, Ağırlık Kaybı=0,01, Çevrim Sayısı=200

Gizli Katman Sayısı (Kodlayıcı+Darboğaz)	Sırası ile Gizli Katmandaki Nöron Sayıları	Öğrenme Adımı	Eğitim için OKH	Test için OKH
2	64 - 4	0,001	0,2255	0,2334
2	20 - 10	0,001	0,2297	0,2343
3	64 -16 - 4	0,001	0,2375	0,2395
3	40 -20 - 10	0,001	0,2162	0,2323
2	64 - 4	0,005	0,2340	0,2351
2	20 - 10	0,005	0,2221	0,2376
3	64 -16 - 4	0,005	0,2334	0,2365
3	40 -20 - 10	0,005	0,2328	0,2359
2	64 - 4	0,01	0,2221	0,2462
2	20 - 10	0,01	0,2337	0,2692
3	64 -16 - 4	0,01	0,2343	0,2370
3	40 -20 - 10	0,01	0,2348	0,2331
2	64 - 4	0,05	0,2376	0,2334
2	20 - 10	0,05	0,2382	0,2341
3	64 -16 - 4	0,05	0,2390	0,2334
3	40 -20 - 10	0,05	0,2383	0,2349

Ağırlık kaybı deneyinde 4 farklı değer kullanılmıştır. En düşük test hatası 0,2324'dir ve 0,01 ağırlık kaybı değeri ile elde edilmiştir.

Çizelge 4.10. Ağırlık Kaybı Belirleme, Optimizasyon Algoritması=Adagrad, Aktivasyon Fonksiyonu=ReLU, Öğrenme Adımı=0,001, Çevrim Sayısı=200

Gizli Katman Sayısı (Kodlayıcı+Darboğaz)	Sırası ile Gizli Katmandaki Nöron Sayıları	Ağırlık Kaybı	Eğitim için OKH	Test için OKH
2	64 - 4	0,001	0,2200	0,2374
2	20 - 10	0,001	0,2198	0,2374
3	64 -16 - 4	0,001	0,2192	0,2357
3	40 -20 - 10	0,001	0,2318	0,2345
2	64 - 4	0,005	0,2080	0,2360
2	20 - 10	0,005	0,2311	0,2359
3	64 -16 - 4	0,005	0,2279	0,2362
3	40 -20 - 10	0,005	0,2247	0,2325
2	64 - 4	0,01	0,2041	0,2424
2	20 - 10	0,01	0,2185	0,2324
3	64 -16 - 4	0,01	0,2414	0,2398
3	40 -20 - 10	0,01	0,2298	0,2347
2	64 - 4	0	0,2272	0,2369
2	20 - 10	0	0,2213	0,2349
3	64 -16 - 4	0	0,2111	0,2340
3	40 -20 - 10	0	0,2210	0,2338

Yapılan 2 boyutlu parametre analizlerinde elde edilen parametre seçimleri, daha önce tür bilgileri eklenmeden yapılan analizler sonucunda elde edilenler ile aynıdır. Bu

hiperparametreler ReLU aktivasyon fonksiyonu, Adagrad optimizasyon algoritması, 0,001 öğrenme adımı ve 0,01 ağırlık kaybı ve kodlama gizli katmanında 20, dar boğaz katmanında 10 ve kod çözücü katmanında 20 nöron sayısı olarak belirlenmiştir. Ancak, tek boyutlu parametre analizlerinde OMH ortalama olarak 0,9 civarındayken 2 boyutlu parametre analizlerinde 0,2 civarındadır. Aynı parametre seçimleri ile aynı veri kümesi üzerindeki hatalar arasındaki fark, tür bilgilerinin öğrenmeye olan olumlu etkisini göstermektedir.

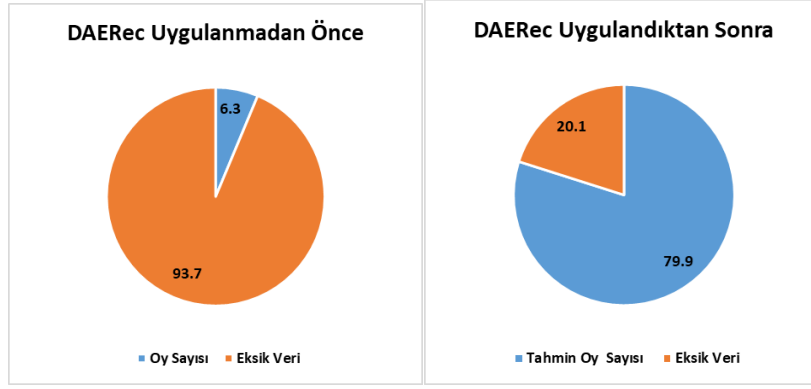


5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde, DAERec yönteminin başarısının incelenmesi için deneysel çalışmalar yürütülmüş ve sonuçları sunulmuştur. Deneysel çalışma, eğitim kümesine ve test kümesine ait başarıların ölçülmesi şeklinde iki aşamalı olarak planlanmıştır. Literatür incelendiğinde (Zhuang ve ark., 2017, Dong ve ark., 2020, Zhu ve ark., 2021) önerilen yönteme ait eğitim başarısını ölçmek için OMH (MAE), OKH (MSE) ve KOKH (RMSE) metriklerinin kullanıldığı görülmüştür. Bunun sebebi, iyi eğitilmiş bir Derin Otomatik Kodlayıcı modelinin çıkışındaki veri ile girişindeki verinin mümkün olduğunca aynı olmasının beklenmesidir. Tavsiye sistemlerinde, ağırlık girişine uygulanan veri kullanıcının oy değerlerinden, çıkış verisi ise ağırlık tahminlerinden oluşmaktadır. Tahmin değerlerinin girişte verilen oy değerlerine yakın olması ağırlık öğrenme başarısını göstermektedir. OMH, OKH ve KOKH metrikleri yöntemin tahmin hesaplama hatasını ölçerler ve düşük değerli olması makbuldür. Test başarısını ölçmek için ise, verilen tavsiyelerin doğruluğu Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor değerleri kullanılarak hesaplanmaktadır. Bu ölçütlerin yüksek değerli olması istenir. Yani değer ne kadar yüksek ise, yapılan tavsiyeler o kadar isabetlidir.

Bu deneysel çalışmalarda Bölüm 4.3'te yapılan parametre analizleri sonucunda belirlenen hiperparametre değerleri kullanılmıştır. Bu hiperparametreler ReLU aktivasyon fonksiyonu, Adagrad optimizasyon algoritması, 0,001 öğrenme adımı ve 0,01 ağırlık kaybı ve kodlama gizli katmanında 20, dar boğaz katmanında 10 ve kod çözücü katmanında 20 nöron sayısı olarak belirlenmiştir.

Tavsiye sistemlerinde kullanılan veri kümelerinin seyrekliği isabetli öneriler üretilmesini etkilemektedir. Önerilen DAERec yönteminde kullanılan Derin Otomatik Kodlayıcı modeli veri seyrekliğini azaltarak tavsiye sisteminin başarısını artırması beklenmektedir. Eğitilen Derin Otomatik Kodlayıcı ağına G tür-oy matrisi verildiğinde çıkışında G' tür-oy matrisi elde edilmektedir. G' matrisinde her tür için farklı bir oy değeri tahmin edilmiştir. G' matrisinde tahmin edilen oy değerinin 0 olması ilgili tür için tahmin üretilmediği anlamına gelmektedir. Movielens100k G' matrisinin oy doluluk oranı %80'dir. Movielens100k veri kümesi için seyreklik %93,7 (Çizelge 3.1).iken eğitilen ağırlık çıkışında veri seyrekliği %20.1 olarak hesaplanmıştır (Şekil 5.1). Özetle veri seyrekliği %93.7'den %20.1'e düşürülmüştür.



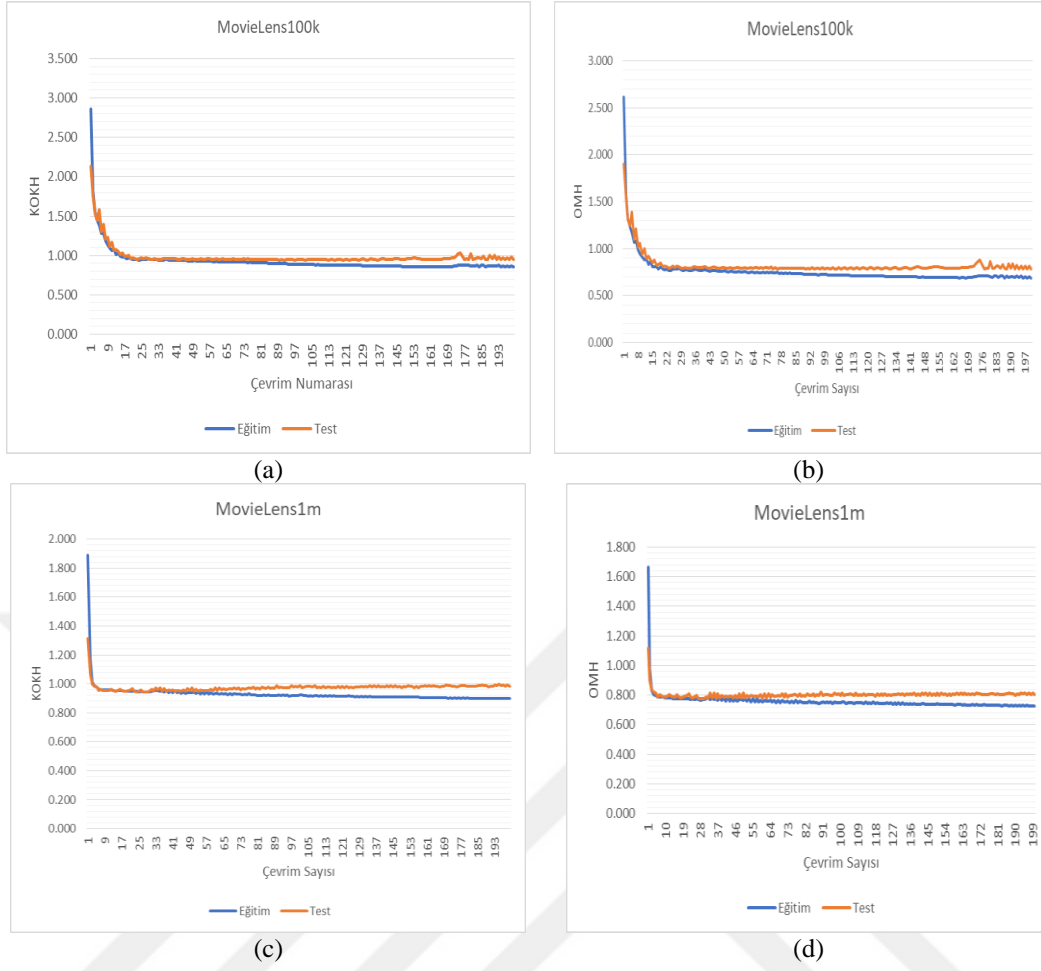
Şekil 5.1. MovieLens100k'ne ait DAERec uygulamadan önceki ve sonraki eksik veri oranları

5.1. Tahmin Değerlendirme Sonuçları

Tahmin hesaplamada, her bir kullanıcı için ağın çıkışındaki Tahmin Matrisindeki, tahmin üretilecek filmin türlerine karşılık gelen değerlerin ortalaması alınır ve ilgili filme kullanıcının vereceği tahmini oy olarak kullanılır. Gerçek oy ile tahmin edilen oyun kıyaslanması için OMH ve KOKH hata ölçütleri hesaplanır. Hata ne kadar düşükse, tahmin o kadar başarılıdır.

Eğitim kümesinin başarısını ölçmek için tasarlanan deneysel çalışmada, beş kez tekrarlanan 5 çapraz doğrulama yönteminden gelen değerlendirme ölçütlerinin ortalaması hesaplanmıştır. Yani her bir sonuç için DAERec 25'er kez çalıştırılmıştır. Çevrim sayısı iki yüz olarak alınmıştır.

MovieLens100k ve MovieLens1m üzerinde yapılan deneylerin çevrim sayısına göre tahmin hata grafikleri değerlendirildiğinde ağın 200 çevrime gerek kalmadan eğitimini tamamladığı görülmektedir (Şekil 5.2). Ağın kısa sürede eğitilmesi ölçeklenebilirlik probleminin çözümüne katkı sağladığına delil olarak gösterilebilir. Ayrıca veri setine yeni kayıtlar eklendikçe, ağın güncellenmesi de hantal bir süreç olmaktan çıkmaktadır.



Şekil 5.2. Çevrim Sayısına göre Tahmin Hata Grafikleri (a) MovieLens100k veri kümesi için KOKH ölçütü ile (b) MovieLens100k veri kümesi için OMH ölçütü ile (c) MovieLens1m veri kümesi için KOKH ölçütü ile (d) MovieLens1m veri kümesi için OMH ölçütü ile

Ek olarak, Zhu ve ark., (2021) çalışmalarında bildirdikleri on bir farklı yöntem (çizelge 5.1) ait sonuçlar ile DAERec yönteminden elde edilen sonuçları kıyaslamak için, ilgili makalede bahsedilen deneysel çalışma gerçekleştirilmiştir. Bunun için eğitim kümesinin dağılımı rasgele ayırma yöntemi kullanılarak sırası ile %60, %70, %80 ve %90 oranları ile örneklenmiş ve bu işlem beş kez tekrarlanarak elde edilen sonuçların ortalaması hesaplanmıştır. Zhu ve ark., (2021), çevrim sayısını üç yüz olarak kullandıkları için, yürütülen deneysel çalışmada çevrim sayısı üç yüz alınmıştır. Tüm hesaplamalar, MovieLens100k ve MovieLens1M veri kümeleri kullanılarak gerçekleştirilmiştir.

Çizelge 5.1’de listesi verilen yöntemler, matris ayrıştırma ve derin otomatik kodlama konusunda yapılan çalışmalardır ve tümü İFT tabanlı yöntemler altında yer almaktadır. Derin otomatik kodlayıcı tabanlı yöntemlerin tümü son beş yıl içinde yayınlanmıştır.

Çizelge 5.1. Kıyaslama için kullanılan algoritmalar

Kategorisi	Kısa Adı	Açık Adı	Basım Yılı	Referansı
Matris Ayırıştırma Tabanlı (İFT)	NMF	Non-negative matrix factorization	2001	(Lee & Seung, 2001)
	PMF	<i>Probabilistic Matrix Factorization</i>	2007	(Salakhutdinov & Mnih, 2007)
	BPMF	Bayesian probabilistic matrix factorization using markov chain Monte Carlo	2008	(Salakhutdinov & Mnih, 2008)
	SVD++	Factorization meets the neighborhood	2008	(Koren, 2008)
	PRA	A probabilistic rating auto-encoder for personalized recommender systems.	2015	(Liang & Baldwin, 2015)
Derin Otomatik Kodlayıcı Tabanlı (İFT)	HCRSA	Hybrid Collaborative Recommendation via Semi-AutoEncoder	2017	(S. Zhang ve ark., 2017)
	ReDa	Representation learning via Dual-Autoencoder for recommendation	2017	(Zhuang ve ark., 2017)
	GraphRec	Attribute-aware non-linear co-embeddings of graph features	2019	(Rashed ve ark., 2019)
	HCRDa	Hybrid Collaborative Recommendation via Dual-Autoencoder	2020	(Dong ve ark., 2020)
	Item-Agrec	Hybrid collaborative recommendation of co-embedded item attributes and graph features	2021	(Dong ve ark., 2021)
	CAPR	Collaborative Autoencoder for Personalized Recommendation	2021	(Zhu ve ark., 2021)

Çizelge 5.1’de yer alan yöntemlere ait kısa açıklamalar aşağıda verilmiştir:

NMF (Lee & Seung, 2001): Tavsiye sistemleri için matris çarpanlara ayırma yöntemi kullanılmıştır ve deneylerde güncelleme kuralları olarak geliştirilmiş Kullback-Leibler sapması kullanılmıştır.

PMF (Salakhutdinov & Mnih, 2007) : Oyların sayısı ile doğrusal olarak ölçeklenen ve benzer öğeleri değerlendiren kullanıcıların, benzer tercihlere sahip olduğu varsayımına dayalı bir yöntem önerilmiştir.

BPMF (Salakhutdinov & Mnih, 2008) : Tüm model parametreleri ve hiperparametreleri entegre ederek model kapasitesinin otomatik olarak kontrol edildiği ve model eğitimi için Markov zinciri Monte Carlo yönteminin uygulandığı PMF'ye dayalı tamamen Bayesian bir işlem önerilmiştir.

SVD++ (Koren, 2008) : Gizli faktör ve komşuluk modellerini bir modelde birleştiren bir yöntemdir ve bu yöntemde kullanıcıların açık ve örtük geri bildirimlerinden yararlanır.

PRA (Liang & Baldwin, 2015) : Otomatik kodlayıcı tabanlı gizli kullanıcı özelliği profilleri oluşturur ve oluşturulan profillere dayalı kişiselleştirilmiş öneriler için İFT yöntemi kullanılmıştır.

HCRSA (S. Zhang ve ark., 2017) : Yarı otomatik kodlayıcıya dayalı bir işbirlikçi filtreleme yöntemi önerilmiştir. Bu yöntemde, tahmin matrisi doğrudan yarı otomatik kodlayıcı modelin çıktı katmanından elde edilir.

ReDa (Zhuang ve ark., 2017) : Yardımcı bilgi olmadan yalnızca derecelendirme matrisine dayanan, kullanıcıların ve öğelerin gizli özellik temsillerini aynı anda öğrenen ReDa yönteminde Otomatik Kodlayıcı kullanımı önerilmiştir.

GraphRec (Rashed ve ark., 2019) : Bu çalışmada kullanıcıların ve ürünlerin vektörünü, öznitelik vektörünü ve grafik vektörünü birlikte modele gömen laplasiyen bir model önerilmiştir.

HCRDa (Dong ve ark., 2020) : Bu yöntemde, kullanıcıların ve ürünlerin özelliklerini aynı anda öğrenmek için bir çift otomatik kodlayıcı modeli tanıtılır ve matris çarpanlarına ayırma, otomatik kodlayıcının eğitim sürecine gömülür.

Item-Agrec (Dong ve ark., 2021) : Oylama tahmini için ürünlerin grafik özelliklerini ve niteliklerini birlikte gömen bir yarı otomatik kodlayıcı kullanılmıştır.

CAPR (Zhu ve ark., 2021) : Kullanıcı tabanlı ve öğe tabanlı özellik temsilleri, verilerin farklı özelliklerini yakalamak için iki farklı otomatik kodlayıcı tarafından öğrenilmesi önerilmiştir.

Bu yöntemlere ait raporlanan değerler, Zhu ve ark. tarafından sunulan (Zhu ve ark., 2021) çalışmadan alınmıştır. Çalışmada belirtilen koşullara uygun DAERec yöntemi de çalıştırılmış ve tüm sonuçlar Çizelge 5.2 ve 5.3’de sunulmuştur.

Çizelge 5.2. Movielens100k üzerindeki OMH ve KOKH eğitim performansları

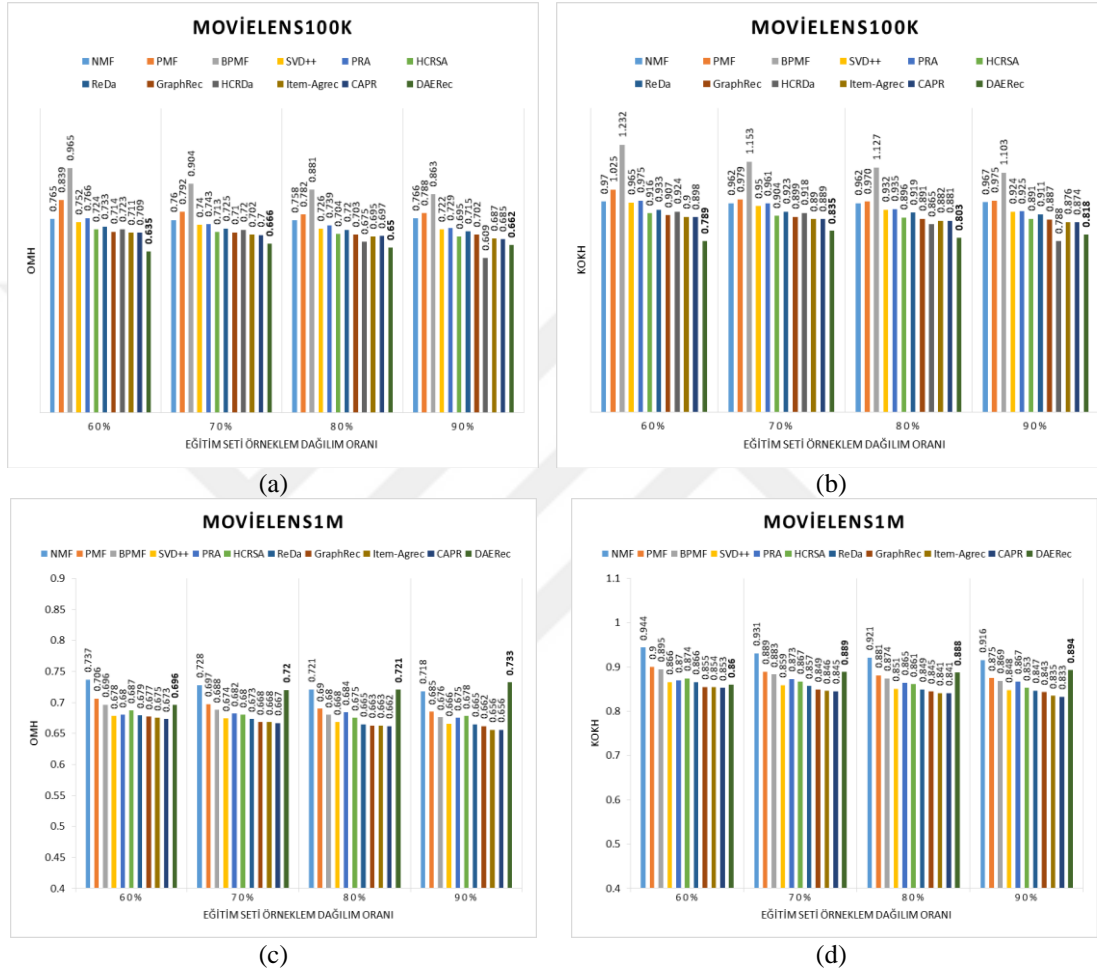
Değerlendirme Kriteri	Yöntem	Eğitim verisi örnekleme yüzdesi			
		60%	70%	80%	90%
OMH	NMF	0,765	0,76	0,758	0,766
	PMF	0,839	0,792	0,782	0,788
	BPMF	0,965	0,904	0,881	0,863
	SVD++	0,752	0,74	0,726	0,722
	PRA	0,766	0,743	0,739	0,729
	HCRSA	0,724	0,713	0,704	0,695
	ReDa	0,733	0,725	0,72	0,715
	GraphRec	0,714	0,71	0,703	0,702
	HCRDa	0,723	0,72	0,675	0,609
	Item-Agrec	0,711	0,702	0,695	0,687
	CAPR	0,709	0,7	0,697	0,685
	DAERec	0,635	0,666	0,650	0,662

KOKH	NMF	0,97	0,962	0,962	0,967
	PMF	1,025	0,979	0,97	0,975
	BPMF	1,232	1,153	1,127	1,103
	SVD++	0,965	0,95	0,932	0,924
	PRA	0,975	0,961	0,935	0,925
	HCRSA	0,916	0,904	0,896	0,891
	ReDa	0,933	0,923	0,919	0,911
	GraphRec	0,907	0,899	0,891	0,887
	HCRDa	0,924	0,918	0,865	0,788
	Item-Agrec	0,9	0,89	0,882	0,876
	CAPR	0,898	0,889	0,881	0,874
DAERec	0,789	0,835	0,803	0,818	

Çizelge 5.3. Movielens1m üzerindeki OMH ve KOKH eğitim performansları

Değerlendirme Kriteri	Yöntem	Eğitim verisinin dağılımı			
		60%	70%	80%	90%
OMH	NMF	0,737	0,728	0,721	0,718
	PMF	0,706	0,697	0,690	0,685
	BPMF	0,696	0,688	0,680	0,676
	SVD++	0,678	0,674	0,668	0,666
	PRA	0,680	0,682	0,684	0,675
	HCRSA	0,687	0,680	0,675	0,678
	ReDa	0,679	0,673	0,665	0,665
	GraphRec	0,677	0,668	0,663	0,662
	Item-Agrec	0,675	0,668	0,663	0,656
	CAPR	0,673	0,667	0,662	0,656
	DAERec	0,696	0,720	0,721	0,733
KOKH	NMF	0,944	0,931	0,921	0,916
	PMF	0,900	0,889	0,881	0,875
	BPMF	0,895	0,883	0,874	0,869
	SVD++	0,866	0,859	0,851	0,848
	PRA	0,870	0,873	0,865	0,867
	HCRSA	0,874	0,867	0,861	0,853
	ReDa	0,866	0,857	0,849	0,847
	GraphRec	0,855	0,849	0,845	0,843
	Item-Agrec	0,854	0,846	0,841	0,835
	CAPR	0,853	0,845	0,841	0,833
	DAERec	0,860	0,889	0,888	0,894

Eğitim verisinin örneklem dağılımına göre tahmin hata grafikleri değerlendirildiğinde önerilen DAERec yönteminin Movielens1m veri kümesi üzerinde diğer yöntemlere göre yüksek sonuçlar aldığı ancak kabul edilebilir aralıkta olduğu görülmüştür (Şekil 5.3). Movielens100k veri kümesi üzerinde en düşük hata değerleri önerilen DAERec yöntemi tarafından elde edilmiştir.

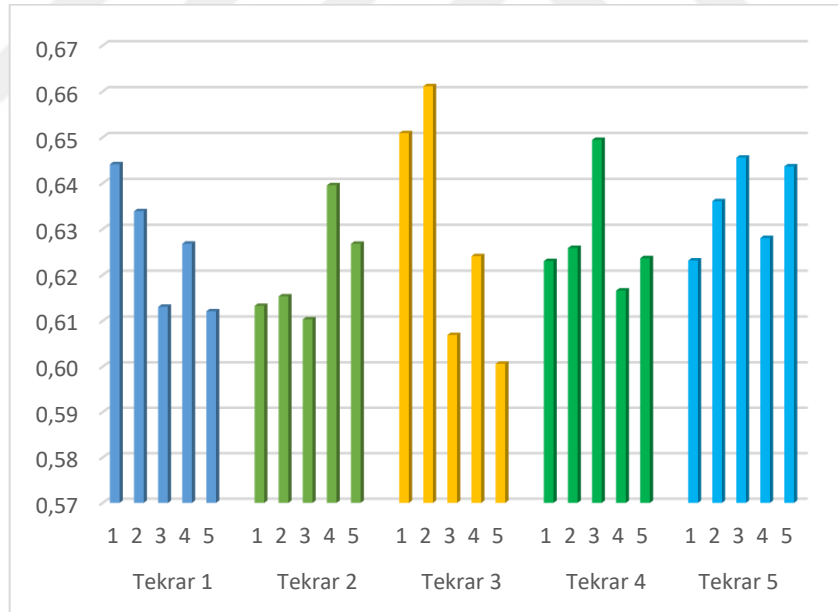


Şekil 5.3. Eğitim verisinin Örneklem Dağılımına Göre Tahmin Hata Grafikleri (a) Movielens100k veri kümesi için KOKH ölçütü ile (b) Movielens100k veri kümesi için OMH ölçütü ile (c) Movielens1m veri kümesi için KOKH ölçütü ile (d) Movielens1m veri kümesi için OMH ölçütü ile

5.2. Tavsiye Değerlendirme Sonuçları

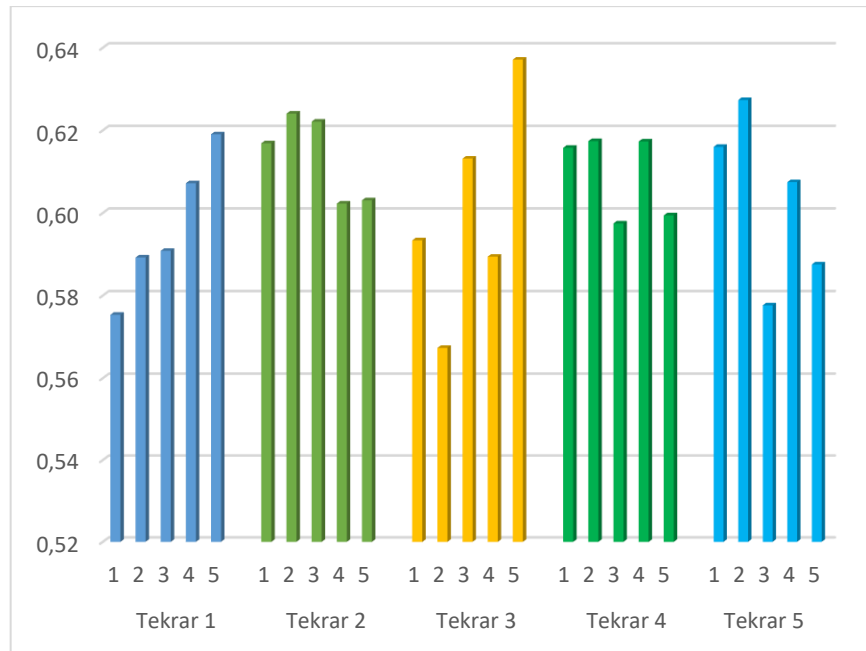
Tavsiye sistemlerinde kullanılan derin ağların eğitimi bittikten sonra, test kümesindeki kullanıcılara film tavsiyeleri sunulmaktadır. Kullanıcıya tavsiye olarak sunulan ilk n film $top-n$ olarak adlandırılmaktadır. Bu tez çalışmasında, 5,10,15 ve 20 film sayıları için tavsiye doğrulukları araştırılmıştır.

Bölüm 4.2’de açıklanan Algoritma-2 kullanılarak tavsiyeler oluşturulur. Şöyle ki, veri kümesindeki her bir kullanıcı sırası ile eğitimi tamamlanmış DAERec ağına verilir, çıkışı hesaplanır. İlgili kullanıcının test kümesinde bulunan ve 4 ve üzerinde oy puanı alan filmler bulunur. Eğer bu filmlerin sayısı n ‘den fazla ise, DAERec ağının çıkışında ilgili filmlere verdiği oylara bakılır ve 3,5 üzerinde oy alanlar, aldığı oya göre büyükten küçüğe göre sıralanır ve ilk n film tavsiye olarak döndürülür. Test kümesinde oylar tam sayı iken, ağın çıkışındaki oylar reel sayıdır. Bu yüzden 3,5 ve üzerinde oy alanlar 4 oy almış olarak düşünülmüştür. Gerçek tavsiye listesi ile DAERec kullanılarak oluşturulan tavsiye listesinin ne kadar uyumlu olduğu *top-10* için Kesinlik, Geri Çağırma ve F1 skor metrikleri kullanılarak değerlendirilir. Movielens100k ve Movielens1m veri kümelerine ait kesinlik, geri çağırma ve F1 skoru sonuçları Şekil 5.4-5.9’de verilmiştir. Sonuçlar 5 kat çapraz doğrulamanın 5 kere çalıştırılması ile elde edilmiştir. Literatürde veri kümesi ve tavsiye olarak döndürülen film sayısı @ işareti ile birleştirilerek isimlendirilmektedir. Örneğin Movielens100k için *top-10* tavsiye listesi *Movielens100k@10* şeklinde isimlendirilir. Tez çalışmasındaki grafik başlıklarında da bu yapı kullanılmıştır.



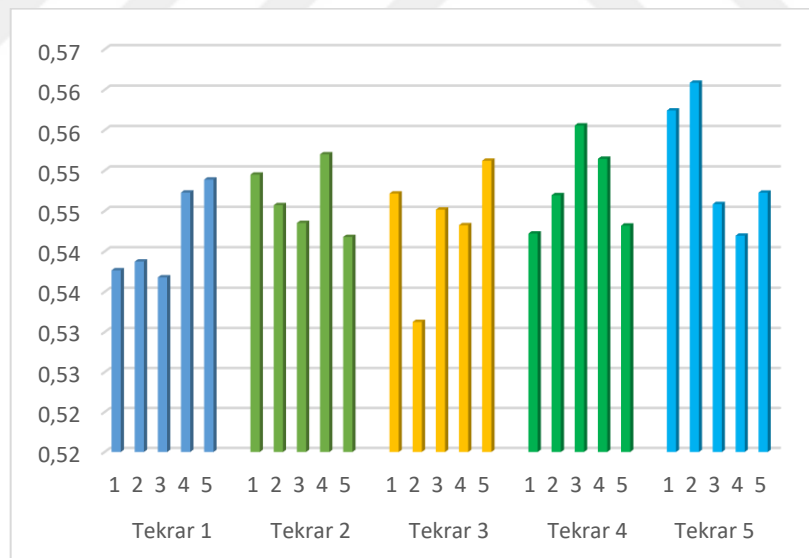
Şekil 5.4. Movielens100k@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Kesinlik Performansı

Movielens100k veri kümesi için en iyi sonuç 0,66 değeri ile 3. tekrarın 2. parçasında elde edilmiştir. Kesinlik değerlerinin en az 0,6 olduğu ve 3. tekrarın 5. parçasında elde edildiği görülmüştür (Şekil 5.4).



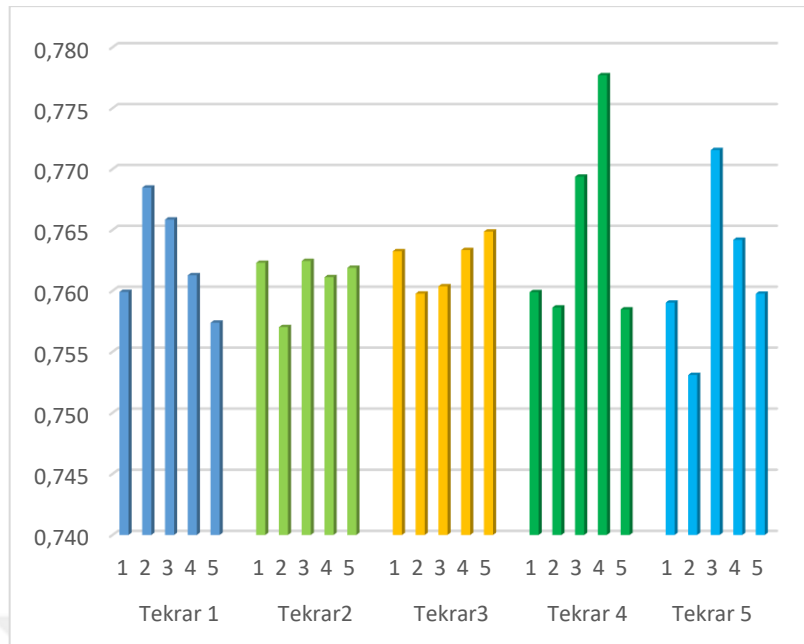
Şekil 5.5. MovieLens100k@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Geri Çağırma Performansı

Şekil 5.5'te görüldüğü üzere en iyi geri çağırma değeri 0,64'tür. Geri çağırma değerleri 0,57'nin altına düşmemiştir.



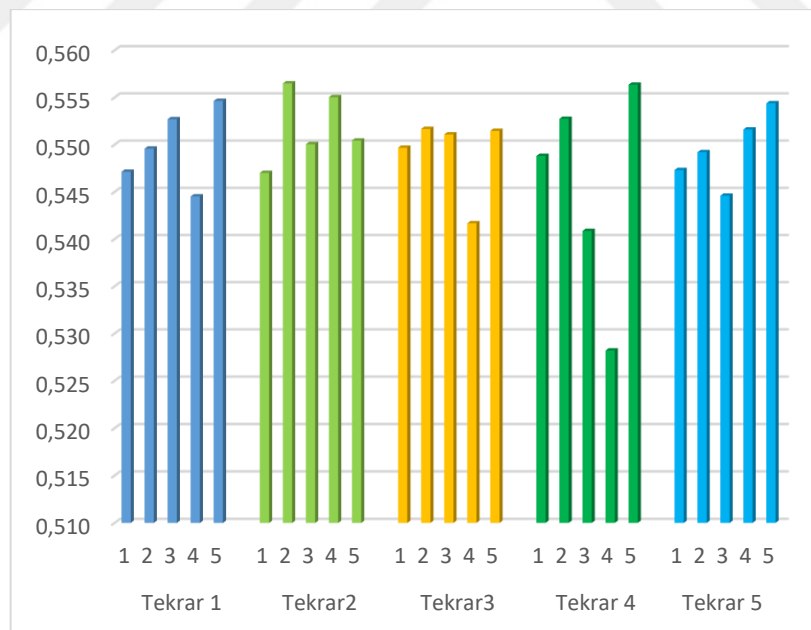
Şekil 5.6. MovieLens100k@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama F1 Skoru Performansı

MovieLens100k veri kümesi üzerinde ölçülmüş en iyi F1 skoru 0,56, en düşük 0,53'tür. Elde edilen sonuçlar kullanıcıya sunulan tavsiyelerden en az yarısının kullanıcının ilgi alanına girdiği anlamına gelmektedir (Şekil 5.6).



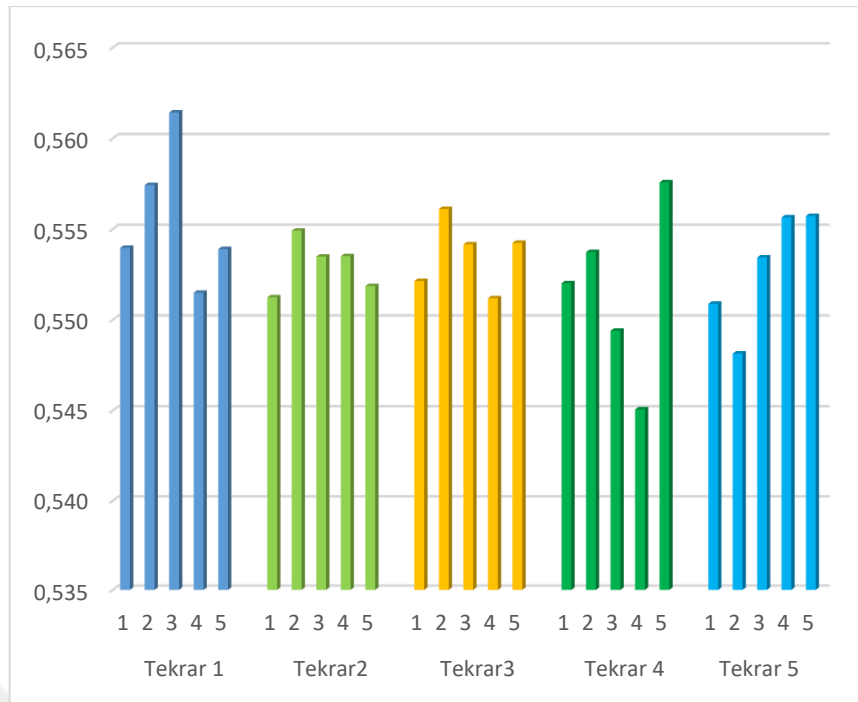
Şekil 5.7. Movielens1m@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Kesinlik Performansı

Movielens1m veri kümesi için en iyi kesinlik sonucu 0,778'dir ve 4. tekrarın 4. parçasında elde edilmiştir. Kesinlik değerlerinin 0,75'in altına düşmediği görülmüştür (Şekil 5.7).



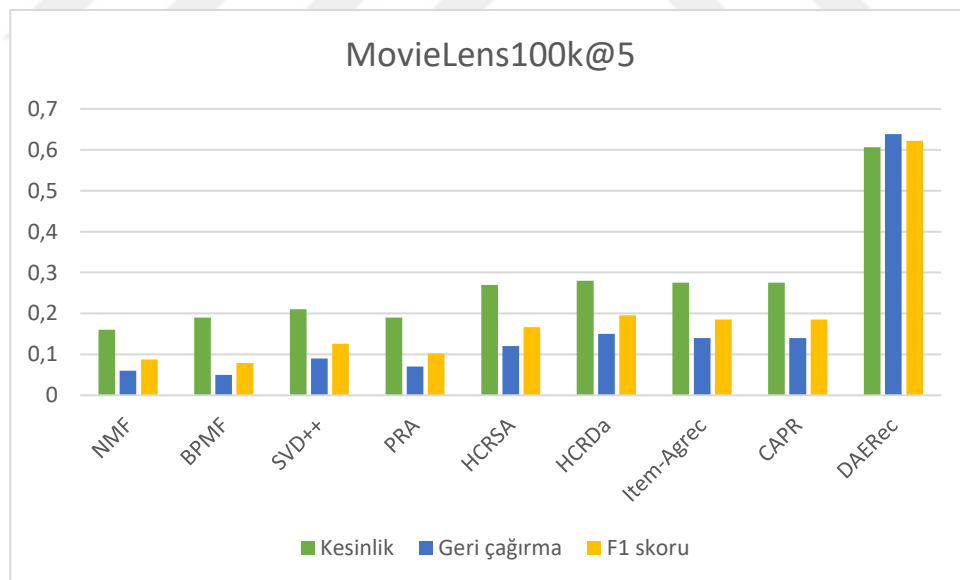
Şekil 5.8. Movielens1m@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama Geri Çağırma Performansı

Şekil 5.8.'de görüldüğü gibi Movielens1m veri kümesi üzerinde en iyi geri çağırma değeri 0,556'dır. En iyi geri çağırma değerleri 2. tekrarın 2. parçası ve 4. tekrarın 5. parçasında elde edilmiştir.



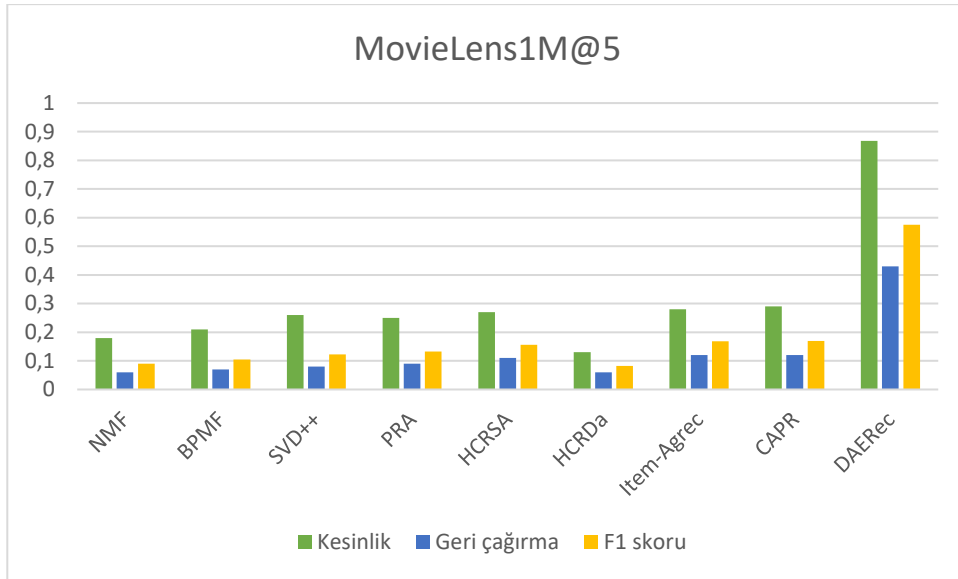
Şekil 5.9. Movielens1m@10 için 5 kere tekrar edilen 5 Kat Çapraz Doğrulama F1 Skoru Performansı

Movielens1m veri kümesi üzerinde elde edilen en iyi F1 skoru 0,561'dir. F1 skoru değerlerinin 0,54'ün altına düşmediği görülmektedir (Şekil 5.9).



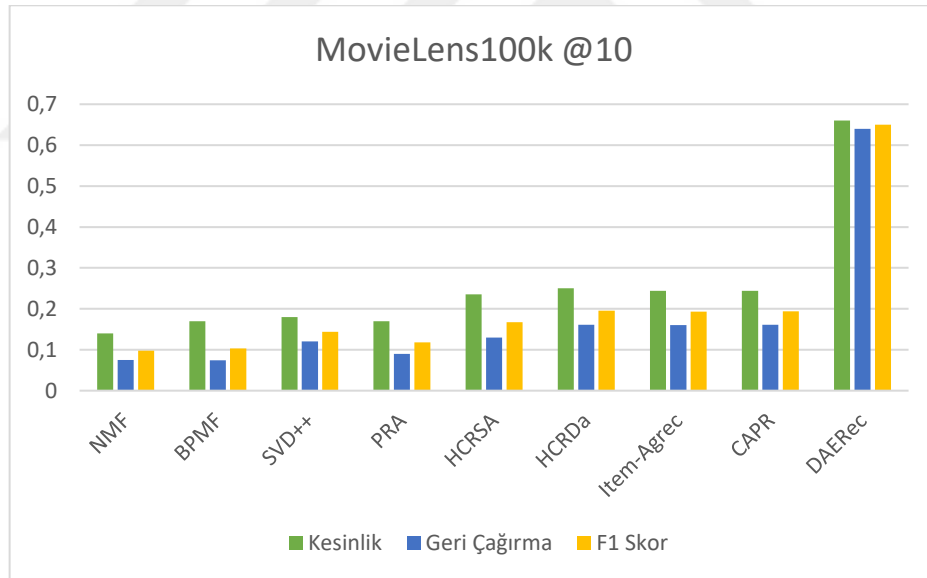
Şekil 5.10. Movielens100k @5 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Şekil 5.10'da görüldüğü üzere önerilen DAERec yöntemi karşılaştırma yapılan yöntemlere göre Movielens100k veri kümesi üzerinde üretilen top-5 tavsiye için daha isabetli tavsiyeler üretmiştir.



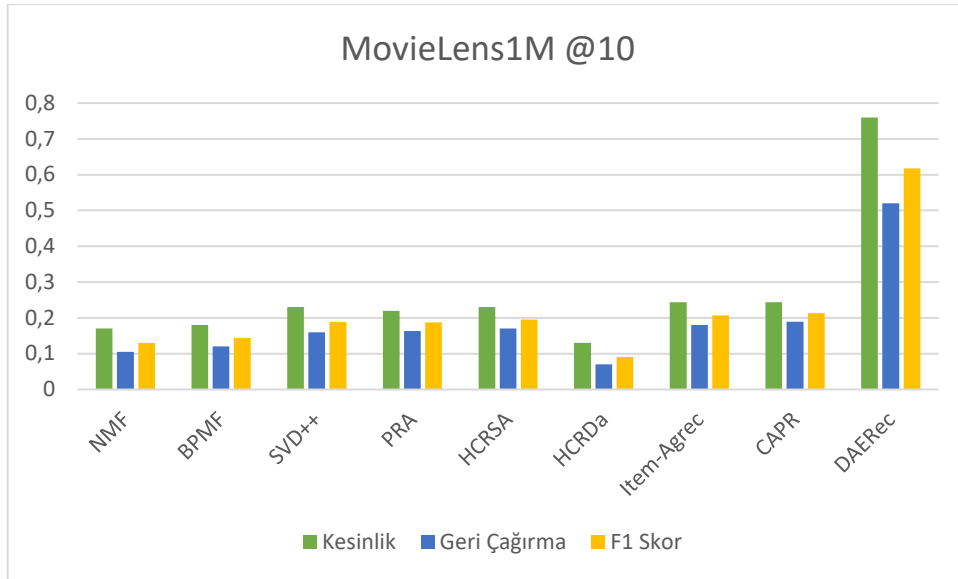
Şekil 5.11. MovieLens1m @5 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

MovieLens1m üzerinde top-5 tavsiye için yapılan deney sonuçlarına göre en başarılı sonuçları önerilen DAERec yöntemi elde etmiştir (Şekil 5.11).



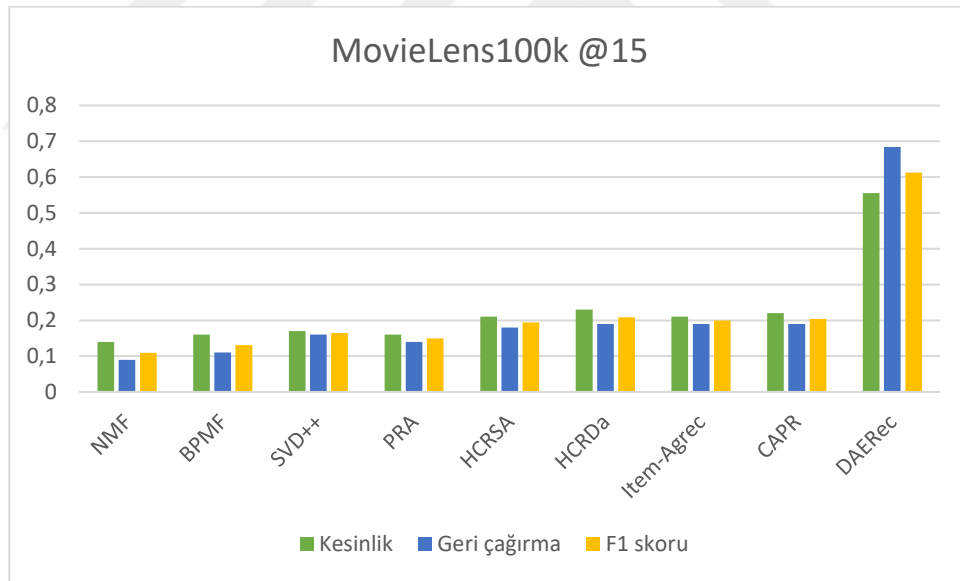
Şekil 5.12. MovieLens100k @10 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Şekil 5.12’da yöntemlere ait MovieLens100k veri kümesi üzerinde elde edilen top-10 (MovieLens100k @10 olarak gösterilmektedir) için kesinlik, geri çağırma ve F1 skor sonuçları paylaşılmıştır.



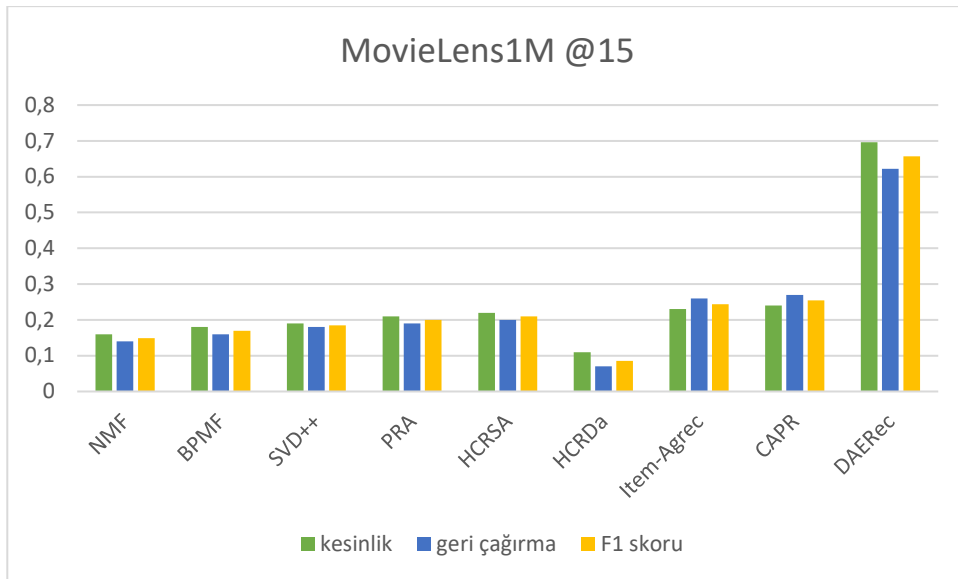
Şekil 5.13. MovieLens1m @10 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Top-10 tavsiye için MovieLens1m veri kümesi üzerinde yapılan deney sonucunda 0,72 kesinlik, 0,51 geri çağırma ve 0,61 F1 skoru değerleri elde edilmiştir (Şekil 5.13).



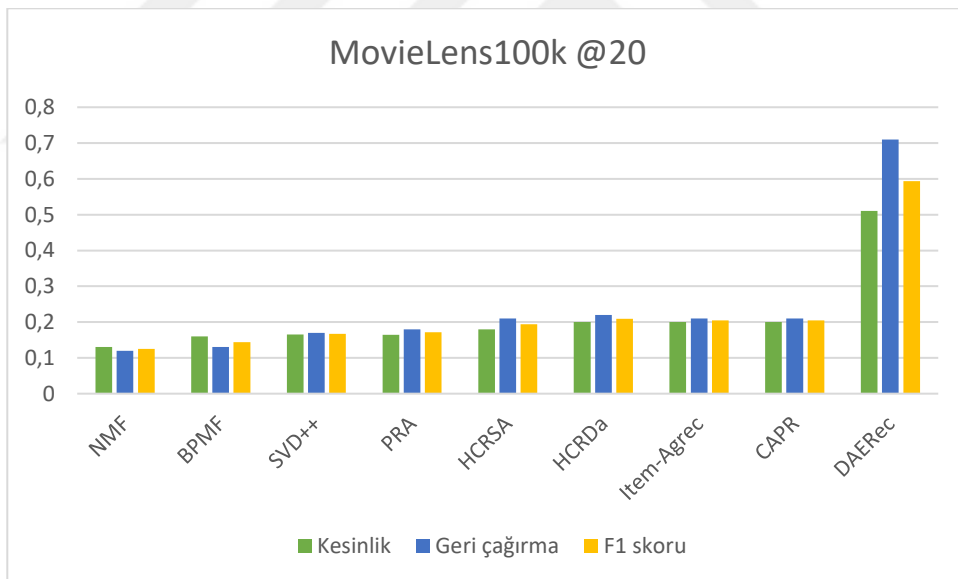
Şekil 5.14. MovieLens100k @15 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Şekil 5.14'te MovieLens100k veri kümesi üzerinde top-15 tavsiye için sonuçlar sunulmuştur. Önerilen DAERec yönteminin en isabetli top-15 tavsiyeyi ürettiği gözlemlenmiştir.



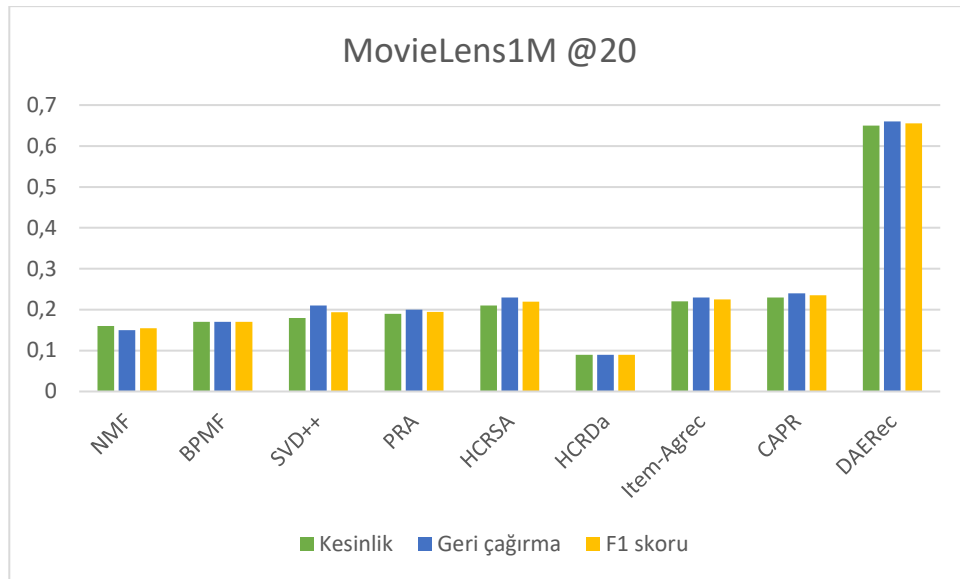
Şekil 5.15. MovieLens1m @15 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Önerilen DAERec yöntemi, MovieLens1m veri kümesi üzerinde top-15 tavsiye için 0,7 kesinlik, 0,61 geri çağırma ve 0,66 F1 skoru sonuçlarını elde etmiştir (Şekil 5.15).



Şekil 5.16. MovieLens100k @20 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Şekil 5.16'da verilen deney sonuçlarına göre önerilen DAERec yönteminin top-20 tavsiye için en isabetli sonuçları sunduğu görülmüştür.



Şekil 5.17. MovieLens1m @20 için Kesinlik (Precision), Geri Çağırma (Recall) ve F1 Skor Sonuçları

Şekil 5.17’de DAERec yönteminin top-20 tavsiye için kesinlik, geri çağırma ve F1 skoru sonuçlarının 0,6 değeri üzerinde olduğu görülmektedir.

Tavsiye değerlendirme kriterlerinin MovieLens100k veri kümesi üzerindeki ortalama sonuçları Çizelge 5.4’te verilmiştir. Tavsiye edilen film sayısı arttıkça kesinliğin azalıp, geri çağırma değerinin arttığı gözlemlenmiştir.

Çizelge 5.4. DAERec yönteminin MovieLens100k veri kümesinde top-n tavsiye için kesinlik, geri çağırma ve F1 skoru performansları

	<i>top-5</i>	<i>top-10</i>	<i>top-15</i>	<i>top-20</i>
Kesinlik	0,62	0,60	0,58	0,53
Geri Çağırma	0,62	0,63	0,65	0,68
F1 skoru	0,55	0,55	0,55	0,55

MovieLens1m veri kümesi üzerinde yapılmış tavsiye performansı deneylerinin ortalama sonuçları Çizelge 5.5’te verilmiştir. Tavsiye edilen film sayısı azaldıkça kesinliğin artıp geri çağırma değerlerinin azaldığı görülmüştür.

Çizelge 5.5. DAERec yönteminin MovieLens1m veri kümesinde top-n tavsiye için kesinlik, geri çağırma ve F1 skoru performansları

	<i>top-5</i>	<i>top-10</i>	<i>top-15</i>	<i>top-20</i>
Kesinlik	0,86	0,76	0,71	0,66
Geri Çağırma	0,42	0,54	0,60	0,64
F1 skoru	0,49	0,55	0,58	0,58

Sunulan şekiller incelendiğinde her iki veri kümesi için de tüm metriklerde DAERec ‘in daha iyi performans gösterdiği görülmektedir. Kesinlik değerinin yüksek

olması, üretilen tavsiyelerinin çoğunun, test kümesindeki gerçek tavsiyeler ile uyduğunu göstermektedir. Geri çağırma değeri yüksek olması ise, gerçekten tavsiye edilmesi gereken filmlerin, sistemin ürettiği tavsiye listesinde yüksek oranda yer alması demektir. Özetle kesinlik, sistem tarafından tüm tavsiye edilen filmler içerisinde ne kadarı gerçekten tavsiye edilmesi gereken film olduğunu ölçerken, geri çağırma gerçekten tavsiye edilmesi gereken filmlerin ne kadarının tavsiye olarak döndürüldüğünü ölçer. F1 skor ise bu iki değer harmonik ortalamasıdır. Sonuç olarak, DAERec diğer yöntemlere göre daha isabetli sonuçlar üretmektedir.

Eğitim başarısı değerlendirilirken kullanılan hata metriklerinde MovieLens1M için DAERec yüksek hata (Şekil 5.2 (c) ve (d)) vermesine karşın, isabetli tavsiyeler üretmesi şu şekilde açıklanabilir: Tahmin hatası hesaplanırken tüm sonuçların matematiksel farkları tek tek hesaplanır. Örneğin 5 olması gerekirken 3,6 tahmin edildi ise 1,4 fark hatadır. Ancak tavsiye üretilirken 3,6 değeri 3,5 üstünde olduğu için 5 oy değeri ile aynı kümededir ve tavsiye olarak kullanıcıya sunulabilir. Şekil 5.10 ile 5.17 arasındaki grafikler test kümesi üzerinde gerçekleştirilen tavsiye doğrulukları göstermektedir. Test kümeleri üzerinde kayda değer bir başarı artışı sağlanmıştır. Bu başarı artışı, eğitim safhasında veri seyrekliği oldukça yüksek olan MovieLens1m veri kümesinde diğer yöntemlere göre yüksek ama rekabetçi sayılacak derecede OMH/KOKH hatasına sahip DAERec yönteminin, model genellemesini daha iyi yaptığını göstermektedir.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında tavsiye sistemlerinin veri seyrekliği ve ölçeklenebilirlik problemleri üzerinde çalışılmış ve çözüm olarak yeni bir Derin Otomatik Kodlayıcı tabanlı model önerilmiştir. Önerilen modelde ağın ürünlerin tür bilgilerini de öğrenerek daha iyi tahminler üretmesi ve kullanıcıya daha uygun tavsiyelerde bulunulması amaçlanmıştır.

İlk olarak veri setinde bulunan oy ve tür bilgileri ön işlemden geçirilerek G tür oy matrisi oluşturulmuştur. Bu matriste tür bilgisini belirten 1 biti yerine oy değeri yazılmıştır. Bu sayede kullanıcının değerlendirme davranışında kullanıcının ilgi alanına giren film türlerinin öğrenilmesi hedeflenmiştir.

İkinci olarak, tavsiye sisteminin eğitim ve test başarısını önemli ölçüde etkileyen hiperparametrelerin iki aşamada analizleri yapılmıştır. İlk aşamada, ağın girişine sadece oylar verilmiş, film tür bilgileri ağa verilmemiştir. İkinci aşamada, iki boyutlu Tür-Oy matrisi ağa verilmiş, bu ağ için uygun parametre değerleri araştırılmıştır. Her iki model için de, araştırma konusu olan parametrelerin en uygun değerleri Movielens100k veri kümesi üzerinde yapılan deneyler ile elde edilmiştir. Yapılan iki boyutlu parametre analizlerinde elde edilen parametre seçimleri, daha önce tür bilgileri eklenmeden yapılan analizler sonucunda elde edilenler ile aynıdır. Bu hiperparametreler ReLU aktivasyon fonksiyonu, Adagrad optimizasyon algoritması, 0,001 öğrenme adımı ve 0,01 ağırlık kaybı ve kodlama gizli katmanında 20, dar boğaz katmanında 10 ve kod çözücü katmanında 20 nöron sayısı olarak belirlenmiştir. Ancak, tek boyutlu parametre analizlerinde OMH ortalama olarak 0,9 civarındayken 2 boyutlu parametre analizlerinde 0,2 civarındadır. Aynı parametre seçimleri ile aynı veri kümesi üzerindeki hatalar arasındaki fark, tür bilgilerinin öğrenmeye olan olumlu katkısını göstermektedir.

Elde edilen parametre değerleri ile eğitilen DAERec modeline ait tahmin performansı OMH ve KOKH değerlendirme kriterleri ile ölçülmüştür. Movielens100k veri kümesi için en iyi OMH ve KOKH değerleri %60 eğitim verisi üzerinde sırasıyla 0,635 ve 0,789'dur. Movielens1m veri kümesi için en iyi OMH ve KOKH değerleri %90 eğitim verisi üzerinde sırasıyla 0,656 ve 0,833'tür.

Önerilen modele ait en iyi 5,10,15 ve 20 tavsiye için performansı kesinlik, geri çağırma ve F1 skoru değerlendirme kriterleri ile ölçülmüştür ve literatürdeki on bir farklı yöntem ile kıyaslanmıştır. DAERec, kullanıcıya sunulan tavsiye aşamasında diğer yöntemlere göre üstün bir başarı göstermiştir. Kesinlik, geri çağırma ve F1 skoru

değerlendirme kriterlerinin 1'e yakın olması makbuldür. 1 değerini alması %100 doğru tavsiye verildiği anlamına gelmektedir. Literatür yöntemlerinin kesinlik ve geri çağırma değerleri 0,25-0,3 arasında iken önerilen DAERec yönteminin kesinlik ve geri çağırma değerleri 0,52-0,77 arasındadır. Elde edilen bu sonuçlar kullanıcıya tavsiye edilecek filmlerden en az %50'sinin kullanıcının ilgi alanında olacağı anlamına gelmektedir.

DAERec yönteminin Movielens100k veri kümesi üstündeki başarısını Movielens1m üstünde de sağlaması veri seyrekliği problemini aştığının göstergesidir.

Eğitim grafikleri incelendiğinde kısa bir süre sonra yok denilebilecek kadar az değişiklik olduğu gözlenmiştir. Ağın eğitim süresinin kısa olması, büyük veri kümeleri üstünde bile ölçeklenebilirlik problemini aştığını göstermektedir.

Önerilen DAERec yönteminin tavsiye sistemlerinin problemlerine çözüm niteliğinde olması, kullanıcıya sunulacak ürün tavsiyelerinde ürün türlerinin öğrenilmesinin faydalı olduğunu göstermektedir.

Veri kümesinin boyutu büyüdüğünde oy matrisindeki boşluk arttığı için ağın öğrenme başarısı düşmektedir. Tür bilgisinin öğrenilmesi ile artan başarı daha fazla bilginin öğrenilmesi ile artabilir. Bu durumda önerilen yöntem ile birlikte içerik tabanlı yöntemler kullanılarak oluşturulan hibrit bir tavsiye sisteminin daha yüksek başarı elde etmesi beklenmektedir.

Veri girişinin iki boyutlu hale getirilerek resme benzetilmesi ve ağın eğitimi için kullanılan bilgisayarlı görme algoritmasının başarılı sonuçlar elde edilmesi farklı bilgisayarlı görme algoritmalarının tavsiye sistemlerine entegre edilerek iyi sonuçlar elde edebileceği beklentisini oluşturmuştur.

7. KAYNAKLAR

- Aydın, A., & Surer, E. (2020). *Derin Peki, stirmeli Peki, stirmeli Ögrenmede Özellik Çıkarma için Atari Oyunlarında Üretici Çeki, smeli Çeki, smeli Ağların Kullanımı Using Generative Adversarial Nets on Atari Games for Feature Extraction in Deep Reinforcement Learning.*
- Baldi, P. (2012). *Autoencoders, Unsupervised Learning, and Deep Architectures* (Vol. 27). JMLR Workshop and Conference Proceedings. <http://proceedings.mlr.press/v27/baldi12a.html>
- Barron, J. T. (2017). Continuously Differentiable Exponential Linear Units. *ArXiv*. <http://arxiv.org/abs/1704.07483>
- Bell, R. M., & Koren, Y. (2007). *Improved Neighborhood-based Collaborative Filtering.*
- Bengio, Y. (2009). *Learning Deep Architectures for AI.* Retrieved January 8, 2021, from <http://www.iro.umontreal.ca/~bengioy>
- Breese, J. S., Heckerman, D., & Kadie, C. (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering.* <https://doi.org/10.5555/2074094.2074100>
- B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *Undefined*, 110(4), 31–36. <https://doi.org/10.5120/19308-0760>
- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). *FAST AND ACCURATE DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS).*
- Da'u, A., Salim, N., Rabi, I., & Osman, A. (2020a). Recommendation system exploiting aspect-based opinion mining with deep learning method. *Information Sciences*, 512, 1279–1292. <https://doi.org/10.1016/j.ins.2019.10.038>
- Da'u, A., Salim, N., Rabi, I., & Osman, A. (2020b). Weighted aspect-based opinion mining using deep learning for recommender system. *Expert Systems with Applications*, 140, 112871. <https://doi.org/10.1016/J.ESWA.2019.112871>
- Doğan, F., & Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*, 10(2), 409–445. <https://doi.org/10.24012/dumf.411130>
- Dong, B., Zhu, Y., Li, L., & Wu, X. (2020). Hybrid Collaborative Recommendation via Dual-Autoencoder. *IEEE Access*, 8, 46030–46040. <https://doi.org/10.1109/ACCESS.2020.2979255>

- Dong, B., Zhu, Y., Li, L., & Wu, X. (2021). Hybrid collaborative recommendation of co-embedded item attributes and graph features. *Neurocomputing*, 442, 307–316. <https://doi.org/10.1016/J.NEUCOM.2021.01.129>
- Duchi JDUCHI, J., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization * Elad Hazan. In *Journal of Machine Learning Research* (Vol. 12, Issue 61). <http://jmlr.org/papers/v12/duchi11a.html>
- Georgiev, K., & Nakov, P. (2013). *A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines*. PMLR. <http://proceedings.mlr.press/v28/georgiev13.html>
- Graves, A. (2014). *Generating Sequences With Recurrent Neural Networks*.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>
- Jiang, J., Li, W., Dong, A., Gou, Q., & Luo, X. (2020). A Fast Deep AutoEncoder for high-dimensional and sparse matrices in recommender systems. *Neurocomputing*, 412, 381–391. <https://doi.org/10.1016/j.neucom.2020.06.109>
- Kantepe, E., & Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı DANIŞMAN Hasan ERDAL, B. (2019). *DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÜRÜN ÖNERME SİSTEMİ TASARIMI MARMARA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ*.
- Kingma, D. P., & Lei Ba, J. (2015). *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*.
- Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–434. <https://doi.org/10.1145/1401890.1401944>
- Krizhevsky, A., & Hinton, G. E. (2011). *Using Very Deep Autoencoders for Content-Based Image Retrieval*. Retrieved January 9, 2021, from <http://www.i6doc.com/en/livre/?GCOI=28001100817300>.
- Kuchaiev, O., & Ginsburg, B. (2017a). Training Deep AutoEncoders for Collaborative Filtering. *ArXiv*. <http://arxiv.org/abs/1708.01715>
- Kuchaiev, O., & Ginsburg, B. (2017b). *Training Deep AutoEncoders for Collaborative Filtering*. <http://arxiv.org/abs/1708.01715>

- Kurt Prof Mehmet Önder Efe, F. D. (2018). *EVRIŞİMLİ SİNİR AĞLARINDA HİPER PARAMETRELERİN ETKİSİNİN İNCELENMESİ ANALYSIS OF THE EFFECTS OF HYPERPARAMETERS IN CONVOLUTIONAL NEURAL NETWORKS*.
- Lee, D. D., & Seung, H. S. (2001). *Algorithms for Non-negative Matrix Factorization*.
- Liang, H., & Baldwin, T. (2015). A probabilistic rating auto-encoder for personalized recommender systems. *International Conference on Information and Knowledge Management, Proceedings, 19-23-Oct-2015, 1863–1866*.
<https://doi.org/10.1145/2806416.2806633>
- Liu, F., Chen, Z., Ding, Y., Yang, S., & Zhang, T. (2021). A face attribute based recommendation system via integrating denoising autoencoder and hash coding. *Computers and Electrical Engineering, 90*.
<https://doi.org/10.1016/j.compeleceng.2021.107020>
- Liu, J., Dolan, P., & Pedersen, E. R. (2009). *Personalized News Recommendation Based on Click Behavior*. <http://news.google.com>,
- Lyudchik, O. (2016). *Outlier detection using autoencoders*.
- Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. *Proceedings of the 8th International Conference on Intelligent User Interfaces - IUI '03*. <https://doi.org/10.1145/604045>
- Nassar, N., Jafar, A., & Rahhal, Y. (2020). A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems, 187, 104811*. <https://doi.org/10.1016/j.knosys.2019.06.019>
- Pan, Y., He, F., Yan, X., & Li, H. (2021). A synchronized heterogeneous autoencoder with feature-level and label-level knowledge distillation for the recommendation. *Engineering Applications of Artificial Intelligence, 106*.
<https://doi.org/10.1016/j.engappai.2021.104494>
- Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications, 83, 300–313*. <https://doi.org/10.1016/j.eswa.2017.04.046>
- R, K., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications, 144, 113054*.
<https://doi.org/10.1016/j.eswa.2019.113054>

- Rashed, A., Grabocka, J., & Schmidt-Thieme, L. (2019). Attribute-aware non-linear co-embeddings of graph features. *RecSys 2019 - 13th ACM Conference on Recommender Systems*, 314–321. <https://doi.org/10.1145/3298689.3346999>
- Salakhutdinov, R., & Mnih, A. (2007). *Probabilistic Matrix Factorization*.
- Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using markov chain Monte Carlo. *Proceedings of the 25th International Conference on Machine Learning*, 880–887. <https://doi.org/10.1145/1390156.1390267>
- Sedhain, S., Menon, A. K., Sanner, S., Xie, L., & Nicta. (2015). *AutoRec: Autoencoders Meet Collaborative Filtering*. <https://doi.org/10.1145/2740908.2742726>
- Simon, Seymour. (1997). *The brain : our nervous system*.
- Strub, F., Gaudel, R., & Mary, J. (2016). Hybrid recommender system based on autoencoders. *ACM International Conference Proceeding Series, 15-September-2016*, 11–16. <https://doi.org/10.1145/2988450.2988456>
- Strub, F., Mary, J., & Mary, J. (2015). *Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs*. <https://hal.inria.fr/hal-01256422v1>
- Sun, Y., & Zhang, Y. (2018). Conversational recommender system. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, 235–244. <https://doi.org/10.1145/3209978.3210002>
- Tilahun, B., Awono, C., & Batchakui, B. (2017). A Survey of State-of-the-art: Deep Learning Methods on Recommender System. *International Journal of Computer Applications*, 162(10), 17–22. <https://doi.org/10.5120/ijca2017913361>
- Tran, D. H., Hussain, Z., Zhang, W. E., Khoa, N. L. D., Tran, N. H., & Sheng, Q. Z. (2018). Deep Autoencoder for Recommender Systems: Parameter Influence Analysis. *ACIS 2018 - 29th Australasian Conference on Information Systems*. <https://doi.org/10.48550/arxiv.1901.00415>
- TÜRK, F. (2019). Machine Learning of Kidney Tumors and Diagnosis and Classification by Deep Learning Methods. *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*, 11(3), 802–812. <https://doi.org/10.29137/umagd.640667>
- Wang, H., Wang, N., & Yeung, D. Y. (2015). Collaborative deep learning for recommender systems. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. <https://doi.org/10.1145/2783258.2783273>

- Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69, 1339–1351. <https://doi.org/10.1016/j.eswa.2016.09.040>
- Wu, H., Zhang, Z., Yue, K., Zhang, B., He, J., & Sun, L. (2018). Dual-regularized matrix factorization with deep neural networks for recommender systems. *Knowledge-Based Systems*, 145, 46–58. <https://doi.org/10.1016/J.KNOSYS.2018.01.003>
- Zeiler, M. D. (2012). *ADADELTA: An Adaptive Learning Rate Method*. <http://arxiv.org/abs/1212.5701>
- Zhang, S., Yao, L., Xu, X., Wang, S., & Zhu, L. (2017). Hybrid Collaborative Recommendation via Semi-AutoEncoder. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10634 LNCS, 185–193. <https://doi.org/10.48550/arxiv.1706.04453>
- Zhang, W., Du, Y., Yoshida, T., & Yang, Y. (2019a). DeepRec: A deep neural network approach to recommendation with item embedding and weighted loss function. *Information Sciences*, 470, 121–140. <https://doi.org/10.1016/j.ins.2018.08.039>
- Zhang, W., Du, Y., Yoshida, T., & Yang, Y. (2019b). DeepRec: A deep neural network approach to recommendation with item embedding and weighted loss function. *Information Sciences*, 470, 121–140. <https://doi.org/10.1016/j.ins.2018.08.039>
- Zhu, Y., Wu, X., Qiang, J., Yuan, Y., & Li, Y. (2021). Representation learning with collaborative autoencoder for personalized recommendation. *Expert Systems with Applications*, 186. <https://doi.org/10.1016/j.eswa.2021.115825>
- Zhuang, F., Zhang, Z., Qian, M., Shi, C., Xie, X., & He, Q. (2017). Representation learning via Dual-Autoencoder for recommendation. *Neural Networks*, 90, 83–89. <https://doi.org/10.1016/j.neunet.2017.03.009>

URL-1 <https://pytorch.org/docs/stable/nn.html#non-linear-activations-weighted-sum-nonlinearity> (Erişim Tarihi : 15.06.2022)