



Araştırma Makalesi

Öneri Algoritmalarının Film Önerme Problemi Üzerinde Karşılaştırılması: MovieLens Örneği

Muhammed Bilal Bozkurt¹, Çiğdem İnan Acı^{*1}

¹Mersin Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Mersin, Türkiye

ÖZ

Anahtar Kelimeler:

Öneri Sistemleri
Yapay Zeka
Veri Madenciliği
MovieLens

Öneri sistemleri kullanıcılara en kısa yoldan istedikleri öğeye ulaşmasını sağlamayı amaçlar. Kullanıcının geçmiş davranışları, öğelerin içerikleri ve daha birçok parametrenin dâhil edildiği birçok farklı öneri yaklaşımı bulunmaktadır. Bunlar kullanıcı memnuniyeti, hız, performans gibi alanlarda farklı sonuçlar sunmaktadır. Kullanıcıya yapılan önerinin en iyi ve en kısa yoldan yapılması kullanıcının memnuniyetini artırır. Bu sebeple günümüzde birçok şirket, öneri algoritmalarını performans ve doğruluk bakımından geliştirmenin yollarını aramaktadır. Bu çalışmada MovieLens veri kümesini kullanılarak üç farklı öneri sistemi geliştirilmiş performansları karşılaştırılmıştır.

Comparison of Recommendation Algorithms on the Movie Recommendation Problem: The MovieLens Example

Keywords:

Recommendation Systems
Artificial Intelligence
Data Mining
MovieLens

ABSTRACT

Recommendation algorithms enables users to access the specific item which they might be looking for through shortest way. There are many recommendation algorithms which include parameters such as user's past actions, contents of items and further additional parameters could be included. Mentioned algorithms provide varying results in regard to user satisfaction, speed and performance. A suggestion made in most effective way improves the user satisfaction. Therefore many companies today are seeking ways to further develop these recommendation algorithms, in aspects of performance and reliability. In this study, three different recommendation systems were developed using MovieLens dataset and their performances were compared.

*Sorumlu Yazar

*(caci@mersin.edu.tr) ORCID ID 0000-0002-0028-9890
(mbilalbozkurt@gmail.com) ORCID ID 0000-0003-1224-9535

e-ISSN: 2717-8579

Geliş Tarihi: 15/03/2021; Kabul Tarihi: 09/06/2021

Bilgisayar Bilimleri ve Teknolojileri Dergisi

1. GİRİŞ

Öneri sistemleri, bilgi elde etme çalışmalarıyla birlikte ortaya çıkmış bir kavram olup ancak 1990'ların ortasından itibaren ayrı bir araştırma ve çalışma alanı haline gelmiştir. Bu sistemler kullanıcının herhangi bir çabası olmadan onun kişisel tercihlerine (beğenme ve beğenmeme) veya özelliklerine (yaş, cinsiyet bölge vb.) göre uygun öğeler tavsiye eder. Öneri sistemlerini arama motorlarından ayıran bu temel durum kullanıcı memnuniyetini artırmaktadır. Çalışma ve kullanım alanı oldukça geniş olmakla birlikte müzik, film, kitap, alışveriş gibi birçok çeşitli alanda uygulaması mevcuttur. Ancak ne yazık ki farklı alanlardaki farklı sorunlar sebebiyle geniş kapsamlı bir öneri sistemi geliştirmek güçtür.

Amazon (URL-1) ve Netflix gibi şirketlerin de kullanıcıları ile etkileşime geçerken kullandıkları öneri sistemleri birçok alanda uygulanabilir durumdadır. Öneri sistemleri kullanıcı memnuniyetini artırdığı gibi uygulayıcıların kullanıcıları elde tutma oranını da (Tablo 1) artırmaktadır (Kumar, Yadav, Singh ve Gupta, 2015).

Tablo 1. Önerilerin şirkete katkıları

Netflix	İzlenen filmlerin %75'i önerilenler arasından seçiliyor.
Google News	Öneriler %38 daha fazla tıklanıyor.
Amazon	Satışların %35'i önerilenler arasından seçiliyor.

Öneri sistemleri, gerek kullanıcılar arasındaki benzerlikleri, gerek içerikler arasındaki benzerlikleri gerekse her ikisini bir arada kullanarak öneride bulunabilmektedir (Wang, Peng, Jin, Sha, ve Wang, 2016). Bu yöntemler sırasıyla işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit filtreleme olarak isimlendirilmektedir.

Bu çalışmada MovieLens veri kümesini kullanarak üç farklı öneri sistemi geliştirilmiş ve performansları değerlendirilmiştir. Bu sayede yöntemler arasındaki hız farkı ve sonuçların kabul edilebilir olup olmadığı tartışılmıştır.

2. ÖNCEKİ ÇALIŞMALAR

Literatürde öneri sistemlerine yönelik birçok çalışma yapılmıştır. Var olan sistemleri geliştirmeyi amaçlayan çalışmalar olduğu gibi, kullanıcıların en çok tercih ettiği ve memnun kaldığı yöntemleri arayan çalışmalar da bulunmaktadır.

2011 yılında Bobadilla ve ark. optimal benzerlik fonksiyonunu bulabilmek için genetik bir algoritma

sundular. Çalışma sonucunda elde ettikleri benzerlik fonksiyonları daha iyi sonuç vermiştir. Fonksiyonları, korelasyondan %42 daha hızlı çalışmaktadır (Bobadilla, Ortega, Hernando ve Alcalá, 2011).

Ekstrand ve ark. tarafından 2014'te ürün-ürün işbirlikçi Filtreleme (İF), kullanıcı-kullanıcı İF ve Tekil Değer Ayrışımı (TDA) yöntemleri ile öneri sistemleri geliştirilmiş ve sonuçlarını karşılaştırabilmek için 582 kişi ile anket çalışması yapılmıştır. Bu anketin sonucuna göre TDA ve ürün-ürün İF yaklaşımları kullanıcı-kullanıcı İF yaklaşımından daha çok tercih edilmiştir. Ürün-ürün ve kullanıcı-kullanıcı arasında yapılan karşılaştırmada 201 kişinin %71,6'sı ürün-ürünü; ürün-ürün ve TDA arasında yapılan karşılaştırmada 198 kişinin %51'i ürün-ürünü; TDA ve kullanıcı-kullanıcı arasında yapılan karşılaştırmada 183 kişinin %74,3'ü TDA'yi tercih etmiştir (Ekstrand, Harper, Willemsen ve Konstan, 2014).

MovieLens veri kümesini kullanarak kullanıcı ve öğe tabanlıdan farklı bir yöntem test etmek isteyen Zhang ve ark. etiket tabanlı bir öneri sistemi geliştirmişlerdir. Öncelikle etiket tabanında en yakın komşuyu bulup daha sonra İF kullanarak kişiselleştirilmiş önerilerini sunmuşlardır. Ancak aldıkları sonuç kullanıcı tabanlı ve öğe tabanlı yaklaşımların yanında başarılı olamamıştır (Zhang, Peng, Sun ve Liu, 2014).

2014 yılında Wang ve ark. kümeleme algoritması ve boyut indirgeme tekniğini birleştirerek hibrit model tabanlı işbirlikçi filtreleme yaklaşımı denemiştir. Bu yaklaşımda en yakın komşuyu bulmak için genetik algoritmalar da kullanılmıştır. Yaptıkları çalışma sonucunda var olan kümeleme tabanlı İF'lerden daha isabetli öneriler aldıklarını belirtmişlerdir (Wang, Yu, Feng ve Wang, 2014).

Ekstrand ve ark. tarafından MovieLens veri kümesi kullanılarak yapılan çalışmada ortalama reytinge göre, ürün-ürün İF'e göre ve TDA ile 3 farklı öneri sistemi geliştirilmiş olup kullanıcıların bu 3 algoritma arasında seçim şansı sunulmuştur. Toplam 3005 kullanıcı arasından 748 kullanıcı en az bir kez öneri sistemleri arasında geçiş yapmış olup bu kullanıcıların nihai tercihleri şu şekildedir: %45,5 TDA, %39 ürün-ürün (Ekstrand, Kluver, Harper, & Konstan, 2015).

Ahuja ve ark. MovieLens veri kümesinde K-Means Clustering ve K En Yakın Komşu algoritmalarını kullanarak geliştirdikleri sistemin var olan teknikten daha iyi sonuç verdiğini belirtmiştir (Ahuja, Solanki ve Nayyar, 2019).

Geçmişte yapılan bu çalışmaların çoğunda İF ve hibrit yaklaşımın kullanıldığını görüyoruz. Yalnızca İF ya da yalnızca içerik tabanlı filtreleme kullanılmamasının sebebi bu iki yaklaşımın da kendi içinde ayrı ayrı problemleri olmasıdır.

3. ÖNERİ SİSTEMLERİ

Her geçen gün daha da büyüyen çevrimiçi ortamlarda kullanıcıların aradıkları ürünü bulmalarının zorlaşması sonucu ortaya çıkan öneri sistemleri, şirketlerin kârını artırdığı gibi kullanıcıların da memnuniyetini artırmaktadır. Örneğin, film izlemek isteyen bir kullanıcı dakikalarca izleyebileceği bir film aramakta, bulunduğu filmleri ise yönetmen, oyuncu gibi çeşitli parametreler ile filtrelemekte ve belki de içerik olarak fazlasıyla beğenebileceği bir filmi göz ardı edebilmektedir. Öneri sistemleri kullanıcıya beğenebileceği ancak daha önce hiçbir oyuncusunun adını duymadığı filmler önerebilir.

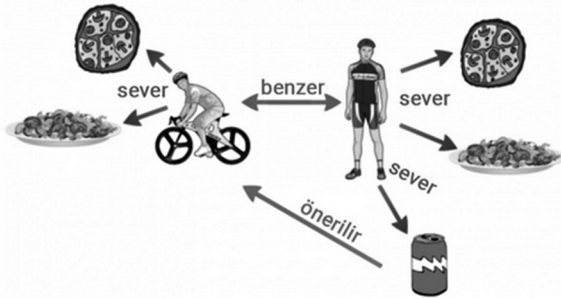
3.1. Öneri Sistemlerinde Öneri Yaklaşımları

Bir öneri sistemi geliştirebilmek için birçok farklı yaklaşım bulunmaktadır. Bu yaklaşımlar kullanıcılar arasındaki benzerlikleri, içerikler arasındaki benzerlikleri veya bu iki yöntemin hibrit hallerini kullanarak öneri sunar. Bu yöntemlere sırasıyla İF, içerik tabanlı filtreleme ve hibrit filtreleme denmektedir. Öneri sistemleri genel olarak bu üç yöntem üzerinden çalışılmaktadır.

3.1.1. İşbirlikçi Filtreleme (İF)

İF, çok genel bir ifadeyle kullanıcının diğer kullanıcılarla olan benzerliklerini temel alarak önerilen bir algoritmadır. A ve B kullanıcılarının izlediği ve oyladığı filmler arasında bir ilişki kurulabilir ve bu kullanıcılardan birisine diğerinin izleyip beğendiği bir film önerilebilir.

Şekil-1'de iki bisikletçinin de pizza ve salata sevdiğini görüyoruz. Bunlara ek olarak sağdaki bisikletçi ayrıca kutu içecek de sevmektedir. Bu sebeple soldaki bisikletçinin de kutu içecek sevdiğini varsayıp ona kutu içecek önerebiliriz. Bu iki bisikletçi arasındaki ilişki ikisinin de pizza ve salata sevmesi yani ortak zevklere sahip olması dolayısıyla kurulmuştur.



Şekil 1. İF'nin görselleştirilmesi

İF, hafıza tabanlı, model tabanlı ve hibrit olmak üzere kendi içinde üçe ayrılır. Model tabanlı yaklaşımda istatistiksel yöntemlerle veya makine öğrenmesiyle boş reytingler tahmin edilir. Hafıza

tabanlı yöntemde ise kullanıcılar arasındaki benzerlikler bulunarak reyting tahmini yapılır.

3.1.1.1. Hafıza tabanlı yaklaşım

Hafıza tabanlı yaklaşımda çözülmesi gereken problem benzerliğine başvurulacak en yakın komşunun nasıl bulunacağıdır. Bu konuda Kosinüs benzerliği sıklıkla kullanılmaktadır. Kosinüs benzerliği, iki kullanıcı vektörünün arasındaki açının kosinüsünü hesaplayarak (Denklem 1) bulunur (Xia, Zhang, ve Li, 2015).

r_u : u kullanıcısının oyu,
 $r_{u'}$: u' kullanıcısının oyu olmak üzere;

$$\text{benzerlik}(u, u') = \frac{r_u \cdot r_{u'}}{|r_u| \cdot |r_{u'}|} = \sum_i \frac{r_{ui} r_{u'i}}{\sqrt{\sum_i r_{ui}^2} \sqrt{\sum_i r_{u'i}^2}} \quad (1)$$

u-kullanıcısının i-ögesi için vereceği oyu diğer tüm kullanıcıların i-ögesi için verdiği oyların ağırlıklı toplamı kullanılarak tahmin edilebilir. Formül hesaplanırken kullanılacak ağırlık ise diğer kullanıcılar ile u-kullanıcısı arasındaki benzerlik olacaktır (Denklem 2).

$r_{u'i}$: u' kullanıcısının i filmi için oyu olmak üzere;

$$\hat{r}_{ui} = \sum_{u'} \text{benzerlik}(u, u') r_{u'i} \quad (2)$$

Son olarak, bulunan sonuç diğer tüm kullanıcıların reytinglerine bölünerek normalize edilir (Denklem 3).

$$\hat{r}_{ui} = \frac{\sum_{u'} \text{sim}(u, u') r_{u'i}}{\sum_{u'} |\text{sim}(u, u')|} \quad (3)$$

3.1.1.2. Model tabanlı yaklaşım

Model tabanlı yaklaşımda kullanılan model, kullanıcının daha önce oylamadığı öğeler için makine öğrenmesi ile reyting tahmini yapar (Şekil 2).

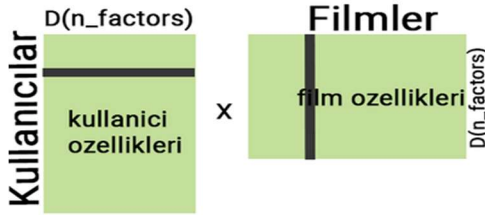


Şekil 2. Model tabanlı yaklaşımda yöntemler

Model tabanlı yaklaşım matris faktörizasyonunu kullanır. Matris faktörizasyonunun arkasındaki ana fikir kullanıcıların tutum ve tercihlerinin az sayıda gizli faktör ile belirlenmesidir. Şekil-3'teki matrisin ayrıştırılmasından sonra, satırlarda ifade edilen vektörlerin, sütunlarda ifade edilen kullanıcı vektörleriyle çarpımı, herhangi bir kullanıcının herhangi bir film için vereceği tahmini reytingi ifade edecektir (Şekil-4) (URL-2).

	Cennet Mahallesi	Arka Sokaklar	Yedi Numara
Bilal	4.5		5.0
Burak		4.5	4.0

Şekil 3. Film-Kullanıcı Matrisi



Şekil 4. Ayrıştırma sonrası durum

Şekil-3'e uygulanan ayrıştırmanın ardından Şekil-4 elde edilir. $D(n_factors)$ parametresi isteğe bağlı olarak değiştirilebilir. Örneğin, $n_factors=3$ olarak alındığında film özelliklerinde "Ne kadar bilim kurgu", "ne kadar yeni", "ne kadar animasyon içeriyor" gibi faktörleri temsil edebilir.

3.1.2. İçerik Tabanlı Filtreleme

İçerik tabanlı filtrelemede kullanıcının daha önce tercih ettiği öğelerin içerikleri incelenir ve bunlara uygun yeni bir içerik kullanıcıya sunulur. Bu yöntem öğelerin birçok özelliğinin (yönetmen, oyuncular, konu vs.) bilinmesini gerektirir.

3.1.3. Hibrit filtreleme

Hibrit filtreleme işbirlikçi ve içerik tabanlı filtrelemelerin dezavantajlarından kurtulabilmek için ikisinin bir arada kullanılmasıyla geliştirilen yöntemdir. Örneğin İF yönteminde sisteme yeni bir öğe eklendiğinde hiç oylanmamış olması ve dolayısıyla kimseye önerilemeyecek olması gibi bir dezavantajı varken içerik tabanlı filtrelemenin kullanıcıya asla kullanıcının zevkleri dışında bir içerik önerememe gibi bir dezavantajı vardır. Bunların üstesinden ikisinin bir arada kullanılmasıyla gelinmektedir.

3.1.4. Popülarite Tabanlı Öneri Sistemleri

Popülarite tabanlı öneri sistemleri kullanıcıların oylamasından bağımsız olarak bir öğenin "erişme" sayısına göre öneriler sunan sistemler için kullanılan terimdir. Bu tür sistemlerde sunulan öneri her kullanıcı için aynıdır. Örneğin popülarite tabanlı film öneri sistemlerinde oylamaya göre değil kaç kullanıcının o filmi izlediğine göre sıralama yapılır. Bu sistemin sorunlarından birisi yeni eklenen filmlerin belirli bir süre zarfında önerilemeyecek olmasıdır.

3.2. Öneri Sistemlerinin Zorlukları

Bir kullanıcıya zevkine göre öneride bulunma fikri ilk bakışta kolay gözükebilir, ancak kullanıcının zevkine uygun doğru öğeleri bulabilmek sadece kullanıcı hakkında bilgi sahibi olmakla çözülememektedir. Buna ek olarak öğeler hakkında, genel bağlam hakkında da bilgi sahibi olmayı ve bunları kullanarak karmaşık hesaplamalara yapmayı gerektirmektedir.

Yapılan bir öneri kullanıcının zevklerine uygun olmasına rağmen kültürüne uygun olmayabilir. Kültürüne uygun olmasına rağmen kullanıcının görüşleri filmin yönetmeninin görüşleriyle zıt düşüyor olabilir. Devam eden paragraflarda öneri sistemlerinin zorlukları sunulmuştur:

3.2.1. Soğuk Başlangıç (Cold Start) Problemi

Soğuk başlangıç ya da ilk oylayıcı sorunu İF'de sisteme yeni eklenen bir öğenin daha önce hiçbir kullanıcı tarafından oylanmamış olması ve bu sebeple hiç kimseye önerilemeyecek olmasından kaynaklanan bir problemdir (Elahi, Ricci ve Rubens, 2016). Aynı durum, sisteme yeni giren bir kullanıcı için de geçerlidir. Kullanıcı ilk geri bildirimini yapana kadar kendisine bir öneri sunulamayacaktır. İçerik tabanlı sistemde ise kullanıcıların oyları veya öğelerin puanları yerine öğenin özellikleri ile hesaplama yapıldığından böyle bir sorun oluşmamaktadır.

3.2.2. Seyrek Veri (Sparsity) Problemi

İF'nin problemlerinden birisi olan seyrek veri problemi sistemde yeterince puanlama olmadığında meydana gelir. Bir matris halinde tutulan kullanıcı/puan yapısındaki boşluklar sistemin verimliliğini düşürür. Bu sorun genellikle sistemde çok öğe az kullanıcı olduğunda meydana gelir.

3.2.3. Ölçeklenebilirlik Problemi

Öneri sistemleri, aynı anda birden çok kullanıcıya hizmet vermeyi hedefler. Aynı anda binlerce kullanıcıya hizmet vermesine ek olarak sistemde milyonları aşkın kullanıcı ve öğe bulunabilir. Böyle bir sistemin verimli çalışabilmesi için sistemi bu yükü taşıyabilecek donanım ve

yazılımla çalıştırılması gerekir. Ölçeklenebilirlik, tüm öneri sistemleri için zorlayıcıdır.

3.2.4. Gri Koyun Problemi

Yine işbirlikçi filtrelemenin sorunlarından birisi olan gri koyun sorunu sistemdeki bazı kullanıcıların hiçbir kullanıcı veya grupta ortak ilgi alanı olmadığında meydana gelir. Ancak sistemdeki kullanıcı ve oylama sayısı arttığında bu sorunla karşılaşma ihtimali azalmaktadır.

3.2.5. Komşu Geçişkenliği Problemi

Öneri sistemlerinin veritabanı çok seyrek olduğunda yani yeterli oylama olmadığında komşu geçişkenliği meydana gelebilir. Öyle ki aynı zevk ve beğeniye sahip iki kullanıcı, sistemde aynı öğeleri hiç oylamamış olabilirler. Bu durumda öneri sisteminin bu iki kullanıcı arasında ilişki kurması mümkün değildir.

3.3. Veri Kümeleri ve Büyük Veri

Öneri sistemlerinin en büyük ihtiyacı kullanıcılar ve onların oylamalarıdır. Bu büyük veriyi uzun emekler sonucu toplayabilir ya da hazır veri kümeleri kullanarak sistemi kısa sürede çalıştırabilir ve test edebiliriz. Piyasada gerçek veriler bulunduğu gibi algoritmik olarak oluşturulmuş veriler de bulunmaktadır.

3.3.1. Veri Kümesi

Bu çalışmada MovieLens veri kümesi kullanılmıştır. MovieLens, 2010 yılında piyasaya sürülmüş olan bir film öneri sistemi projesinin veri kümesidir. 100.000 ve 1 milyon satır boyutlu veri kümeleri kullanıcıların yaş, cinsiyet, meslek, bölge gibi temel verilerini de içermektedir. 10 milyonluk veri kümesi ise bu bilgileri içermeyip sadece oylama verisini içerir. Şekil 5'te MovieLens veri kümesinin ilk beş kaydı görülmektedir.

movieId	title	genres
0	1 Toy Story (1995)	Adventure Animation Children Comedy Fantasy
1	2 Jumanji (1995)	Adventure Children Fantasy
2	3 Grumpier Old Men (1995)	Comedy Romance
3	4 Waiting to Exhale (1995)	Comedy Drama Romance
4	5 Father of the Bride Part II (1995)	Comedy

userId	movieId	rating	timestamp
0	1	4.0	964982703
1	1	3	964981247
2	1	6	964982224
3	1	47	964983815
4	1	50	964982931

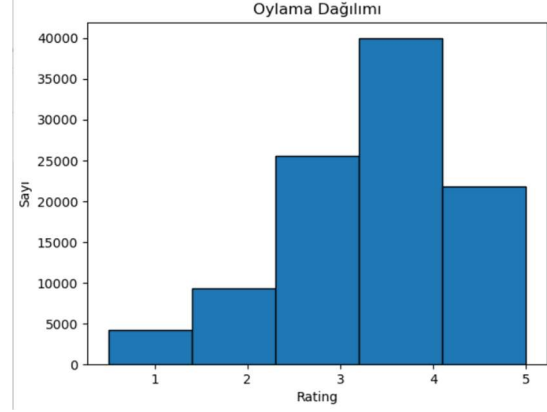
Toplam Film Sayısı: 5
Toplam Kullanıcı Sayısı: 1

Şekil 5. MovieLens veri kümesi

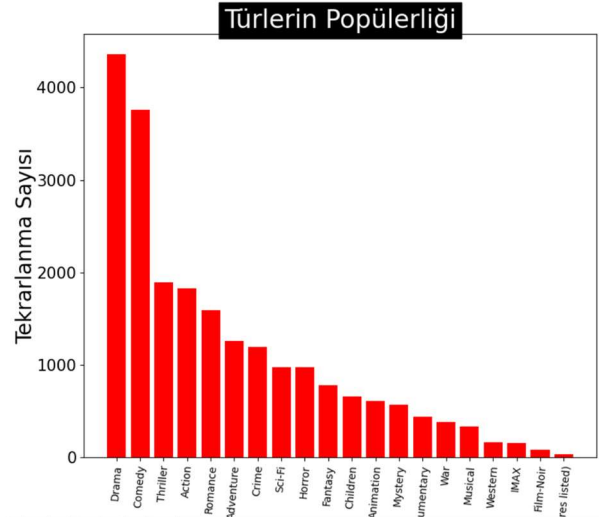
3.4. Öneri Sistemlerinin Geliştirilmesi

Bu çalışmada popülerite tabanlı, TDA ile işbirlikçi filtreleme ve ardından kosinüs benzerliği yöntemleri kullanılarak üç farklı öneri sistemi geliştirilecektir.

Geliştirilecek sistemler için MovieLens tarafından ücretsiz olarak sunulan 600 kullanıcı, 9.000 film ve 100.000 oylamaya sahip olan en küçük veri kümesi kullanılmıştır(URL-3). Bahsedilen veri kümesi, film başlığını, türünü ve kullanıcıların verdiği oylamaları içermektedir. Oylama ve tür dağılımı Şekil-6 ve Şekil-7'de gösterilmiştir.



Şekil 6. Seçilen veri kümesinin oylama dağılımı.



Şekil 7. Seçilen veri kümesinin kategori dağılımı.

4. SONUÇLAR

Çalışmamızda popülerite tabanlı, TDA ile işbirlikçi filtreleme tabanlı ve kosinüs benzerliği tabanlı olmak üzere üç farklı öneri sistemi geliştirilmiş ve sonuçlar karşılaştırılmıştır. Geliştirme ortamının teknik detayları Tablo 2'de verilmiştir.

Tablo 2. Geliştirme ortamı

CPU	Intel i7 6700HQ
GPU	Nvidia GeForce 950m
Dil	Python 3
İşletim Sistemi	Windows 10

MovieLens veri kümesindeki filmler kullanıcılar tarafından oylanmıştır. Veri kümesine yeni eklenen 611. kullanıcının oyladığı filmler Şekil-8'de görülmektedir.

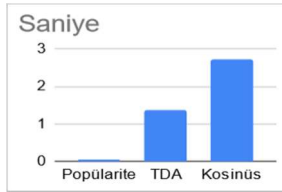
```

611. kullanıcı şu ana kadar toplam 18 filmi oyladı.
#1. A Quiet Place (2018) - 2.0
#2. I, Tonya (2017) - 4.0
#3. Arrival (2016) - 3.0
#4. Interstellar (2014) - 5.0
#5. Truman Show, The (1998) - 4.0
#6. Ray (2004) - 3.5
#7. The Martian (2015) - 4.5
#8. Léon: The Professional (a.k.a. The Professional) (Léon) (1994) - 5.0
#9. La La Land (2016) - 5.0
#10. Love Me If You Dare (Jeux d'enfants) (2003) - 4.0
#11. The Hateful Eight (2015) - 4.5
#12. Cast Away (2000) - 4.0
#13. Django Unchained (2012) - 4.5
#14. Fight Club (1999) - 5.0
#15. Lord of the Rings: The Two Towers, The (2002) - 5.0
#16. Lord of the Rings: The Return of the King, The (2003) - 5.0
#17. Lord of the Rings: The Fellowship of the Ring, The (2001) - 5.0
#18. Seven (a.k.a. Se7en) (1995) - 4.5

```

Şekil 8. Örnek bir kullanıcının oyları.

Geliştirilen sistemler örnek iki kullanıcı için (7. ve 611. kullanıcı) önerilerde bulunmuştur. Her bir algoritma her iki kullanıcı için de 30 adet öneri alınmıştır. Popülarite bazlı öneri sistemi beklendiği üzere her iki kullanıcı için de aynı sonucu vermiştir. Ortalama hesaplama süresi 0,0413 saniyedir. TDA İF sisteminin ortalama hesaplama süresi 1,37 saniye olarak hesaplanmıştır. Kosinüs benzerliği bazlı öneri sisteminin ortalama hesaplama süresi 2,73 saniye olarak hesaplanmıştır. Çalışma sürelerinin karşılaştırması Şekil-9'dan görülebilir.



Şekil 9. Sistemlerin çalışma süresi

611. Kullanıcıya bu öneriler hakkındaki düşünceleri sorulduğunda en iyi sonucu TDA yönteminin verdiğini, ardından sırasıyla kosinüs benzerliğini ve popülarite tabanlı yaklaşımı seçeceğini bildirmiştir. Çalışma sonucunda elde edilen sonuçlar (Şekil-10) İF'nin diğerlerine göre daha başarılı olduğunu göstermektedir.

	Popülarite Tabanlı	Matris Faktörizasyonu Tabanlı	Kosinüs Benzerliği Tabanlı
Kişisel Öneriler	HAYIR	EVET	EVET
Kullanıcı Özellikleri	HAYIR	EVET	EVET
Film Özellikleri	HAYIR	EVET	HAYIR
Yeni Filmleri Önerilebilir	HAYIR	HAYIR	HAYIR

Şekil 10. Sonuçlar

Özetle, TDA ile alınan sonuçlar ve çalışma hızı diğer yaklaşımlara göre daha iyi performans göstermiştir. Hiç animasyon filmi oylamamış olan 611. kullanıcıya TDA ile 4 adet animasyon filmi önerilmiş olması filmlerin içeriklerinin de

hesaplamaya dahil edildiğini göstermektedir. Kosinüs benzerliği ile bulunan en yakının komşu da önerilen bu filmleri izleyip yüksek bir puanla oylamıştır. Hibrit bir yaklaşım olarak TDA ile yapılan öneriler en yakın komşu ile karşılaştırılabilir ve en yakın komşu tarafından beğenilmeyen (düşük puan alan filmler) kullanıcıya önerilmez.

KAYNAKÇA

Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using K-Means clustering and K-Nearest Neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 263-268). IEEE.

Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-based systems*, 24(8), 1310-1316.

Ekstrand, M. D., Harper, F. M., Willemsen, M. C., & Konstan, J. A. (2014, October). User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems* (pp. 161-168).

Ekstrand, M. D., Kluver, D., Harper, F. M., & Konstan, J. A. (2015, September). Letting users choose recommender algorithms: An experimental study. In *Proceedings of the 9th ACM Conference on Recommender Systems* (pp. 11-18).

Elahi, M., Ricci, F., & Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20, 29-50.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009), 1-10.

Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. *International Journal of Computer Applications*, 124(3).

Wang, K., Peng, H., Jin, Y., Sha, C., & Wang, X. (2016). Local weighted matrix factorization for top-n recommendation with implicit feedback. *Data Science and Engineering*, 1(4), 252-264.

Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, 25(6), 667-675.

Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information Sciences, 307*, 39-52.

Zhang, J., Peng, Q., Sun, S., & Liu, C. (2014). Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Physica A: Statistical Mechanics and its Applications, 396*, 66-76.

URL-1: <https://www.amazon.science/the-history-of-amazons-recommendation-algorithm>
[Eriřim Tarihi: 12.03.2021]

URL-2: <https://developers.google.com/machine-learning/recommendation/collaborative/matrix> [Eriřim Tarihi: 12.03.2021]

URL-3: <https://grouplens.org/datasets/movielens/>
[Eriřim Tarihi: 12.03.2021]