T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GENETİK ALGORİTMA İLE AĞIRLIKLANDIRILMIŞ HİBRİT BİR FİLM ÖNERİ SİSTEMİ

Hamit SEYREK

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2020

Doç. Dr. İ. Berkan AYDİLEK danışmanlığında Hamit SEYREK'in hazırladığı "**Genetik Algoritma ile Ağırlıklandırılmış Hibrit bir Film Öneri Sistemi"** konulu bu çalışma "14.01.2020" tarihinde aşağıdaki jüri tarafından oy birliği ile Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Ana Bilim Dalı'nda YÜKSEK LİSANS tezi olarak kabul edilmiştir.

İmza

Danışman : Doç. Dr. İ. Berkan AYDİLEK ...……

Üye : Dr. Öğr. Üyesi Abdülkadir GÜMÜŞÇÜ …….

Üye : Dr. Öğr. Üyesi Abdurrahman ÖZBEYAZ …….

Bu tezin Elektrik Elektronik Mühendisliği Ana Bilim Dalında Yapıldığını ve Enstitümüz Kurallarına Göre Düzenlendiğini Onaylarım.

Doç. Dr. İsmail HİLALİ
Enstitü Müdürü

Not: Bu Tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı 5486 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

Sayfa No

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
ŞEKİLLER DİZİNİ iv
ÇİZELGELER DİZİNİ v
SİMGELER ve KISALTMALAR DİZİNİ vi
1. GİRİŞ 1
1.1. Çalışmanın Önemi ve Amacı 4
2. ÖNCEKİ ÇALIŞMALAR 7
3. MATERYAL ve YÖNTEM 15
3.1. Öneri Sistemleri 15
3.1.1. İşbirlikçi filtreleme 16
3.1.1.1. Benzerlik yöntemleri 19
3.1.1.1.1. Kosinüs benzerliği 20
3.1.1.1.2. Pearson korelasyonu 21
3.1.1.1.3. Ortalama kare farkı benzerliği 23
3.1.1.2. Komşuluk seçimi 25
3.1.1.3. Tahmin hesaplama 26
3.1.1.3.1. Basit ortalama 27
3.1.1.3.2. Ağırlıklandırılmış ortalama 28
3.1.1.3.3. Ayarlanabilir ağırlıklandırılmış ortalama 28
3.1.1.3.4. Standart sapmalı ayarlanabilir ağırlıklandırılmış ortalama 29
3.1.1.4. İşbirlikçi filtreleme algoritması örneği 30
3.1.2. İçerik tabanlı filtreleme 32
3.1.3. Hibrit filtreleme 36
3.2. Öneri Sistemlerindeki Önemli Faktörler 38
3.3. Öneri Sistemleri Araştırmaları 40
3.4. Öneri Sistemleri Uygulamaları 42
3.4.1. Amazon.com çevrimiçi mağazası 44
3.4.2. Netflix.com çevrimiçi film sitesi 45
3.4.3. N11.com çevrimiçi alışveriş sitesi 46
3.4.4. Gittigidiyor.com çevrimiçi alışveriş sitesi 47
3.5. *k*-NN Algoritması 48
3.6. Kullanılan Değerlendirme Ölçütleri 49
3.6.1. Ortalama mutlak hata 49
3.6.2. Hata karelerinin ortalamasının karekökü 50
3.7. Genetik Algoritma 50
3.8. Kullanılan Veri Seti 55
3.9. Önerilen Yöntemler 60
3.9.1. Önerilen içerik tabanlı filtreleme yöntemi 60
3.9.2. Önerilen işbirlikçi filtreleme yöntemi 65
3.9.3. Önerilen hibrit filtreleme yöntemi 67
4. ARAŞTIRMA BULGULARI ve TARTIŞMA 70
4.1. Yapılan Deneylerin Performansları 70
4.1.1. Önerilen içerik tabanlı filtreleme yöntemi deneyleri 70
4.1.2. Önerilen işbirlikçi filtreleme yöntemi deneyleri 73
4.1.3. Önerilen hibrit filtreleme yöntemi için yapılan deneyler 77
4.2. Genetik Algoritma ile Yapılan Optimizasyon Deneyleri 79
4.3. Önerilen Hibrit Film Öneri Sistemi Değerlendirmesi 86
5. SONUÇ ve ÖNERİLER 88
KAYNAKLAR 90
ÖZGEÇMİŞ 95

ÖZET

Yüksek Lisans Tezi

Genetik Algoritma ile Ağırlıklandırılmış Hibrit Bir Film Öneri Sistemi

Hamit SEYREK

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Ana Bilim Dalı

Danışman: Doc. Dr. İ. Berkan AYDİLEK
Yıl: 2020, Sayfa: 95

Günümüzde internet ortamında hemen hemen ihtiyaç olunan tüm bilgiler, ürünler veya hizmetler bulunmaktadır. Fakat bazen aranılan bilgi, ürün veya hizmet geç bulunmakta veya hiç bulunamamaktadır. Çünkü aynı veriler farklı yerlerde yayınlanabilmekte ve bunların bazıları yanlış, kalitesiz veya sahte olabilmektedir. Bir gün içerisinde web ortamına eklenen, güncellenen veya silinen veri miktarı haberdar olunabilecek verinin çok ötesindedir. Bu veri yığınlarının içerisinde kullanıcının ihtiyacını bulabilmesini kolaylaştırmak için öneri sistemleri geliştirilmektedir. Öneri sistemleri kullanıcılara aradıklarının bulmak konusunda yardımcı olduğu gibi kullanıcıların ihtiyaçlarını analiz edip onlara ilgilerini çekebilecek yeni ürünler de önermektedir. Öneri sistemleri sadece kullanıcılara değil aynı zamanda çevrimiçi mağazaların müşterilerine daha iyi hizmet vermesi hedeflenerek firmaya olan bağlılığı arttırmaktadır. Böylece firmalar daha fazla ürün satışı yaparak daha fazla kâr elde etmesi sağlanmaktadır.

Çevrimiçi ortamda aranılan bilgi, ürün veya hizmet arama motorlarının izin verdiği kadarıyla bulunabilmektedir. Yakın gelecekte öneri sistemlerinin arama motorlarının yerini alacağı düşünülmektedir. Artık kullanıcı kendisine lazım olanı aramayacak çünkü sistem kullanıcıya lazım olanı gerekli olduğu zamanda önerebilecektir.

Bu çalışma, öneri sistemlerine genel bir bakış sunmaktadır. Öneri sistemleri yöntemlerinden işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit filtreleme tezde detaylı bir şekilde açıklanmaktadır. Ayrıca konunun öneminin daha iyi anlaşılması için internet ortamında öneri sistemlerini başarılı bir şekilde uygulamakta olan bazı çevrimiçi mağazaların kullandıkları öneri sistemleri hakkında bilgiler verilmektedir.

Bu tez kapsamında MovieLens 100K veri seti üzerinde uygulanmış içerik tabanlı filtreleme ile işbirlikçi filtreleme algoritmalarından elde edilen tahminler bazı varsayımlar sonucu oluşturulmuş bir birleştirme algoritması sayesinde bir araya getirilerek hibrit bir film öneri sistemi oluşturulmaktadır. Kullanılan birleştirme algoritmasındaki katsayılar Genetik Algoritma ile hesaplanmaktadır. Elde edilen deneysel sonuçlar önerilen hibrit film öneri sisteminin daha düşük hata oranlarına sahip olduğunu göstermektedir.

ANAHTAR KELİMELER: Hibrit filtreleme, işbirlikçi filtreleme içerik tabanlı filtreleme, genetik algoritma.

ABSTRACT

MSc Thesis

A Hybrid Film Recommendation System Weighted with Genetic Algorithm

Hamit SEYREK

Harran University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronic Engineering

Supervisor: Assoc. Prof. Dr. İ. Berkan AYDİLEK
Year: 2020, Pages: 95

Today, almost all the information, products or services that are needed on the internet are available. However, sometimes the information, product or service required is late or absent. Because the same data can be published in many different places and some of them may be wrong, poor quality or fake. The amount of data added, updated or deleted on the web in a day is far beyond the data that can be informed. Recommendation systems are developed in order to make it easier for the user to find their needs in these data stacks. Recommendation systems not only help users find what they are looking for, but also recommend new products that can analyze their needs and attract their attention. Recommendation systems increase loyalty to the company by targeting not only the users but also the online stores to serve their customers better. Thus, it is ensured that companies make more profit by selling more products.

Information searched online can be found as far as the product or service search engines allow. It is thought that recommendation systems will replace search engines in the near future. The user will no longer be able to search for what is needed because the system will be able to offer the user what is needed.

This study provides an overview of recommendation systems. Collaborative filtering, content-based filtering and hybrid filtering, which are among the recommendation systems methods, are explained in detail in the thesis. In addition, information is provided about the recommendation systems used by some online stores that are successfully implementing recommendation systems in the internet environment in order to understand the importance of the issue better.

Within the scope of this thesis, a hybrid film recommendation system is formed by combining the predictions obtained from content-based filtering and collaborative filtering algorithms applied on the MovieLens 100K data set by means of a merge algorithm which is formed as a result of some assumptions. In the combining algorithm used, the weights are calculated with Genetic Algorithm. Experimental results show that the proposed hybrid approach has lower error rates.

KEY WORDS: Hybrid filtering, collaborative filtering, content-based filtering, genetic algorithm

TEŞEKKÜR

Bilgi ve deneyimini hiçbir zaman esirgemeyen danışmanım sayın Doç. Dr. İ. Berkan AYDİLEK'e, tezimi hazırlarken destekleriyle bana yardımcı olan yüreğimde büyüttüğüm aileme ve tüm dostlarıma sonsuz teşekkürlerimi sunuyorum.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1. İşbirlikçi filtreleme..... **Error! Bookmark not defined.**
Şekil 3.2. İçerik tabanlı filtreleme..... 32
Şekil 3.3. Amazon öneri sistemi görüntüsü (Amazon.com, 2019) 44
Şekil 3.4. Netflix web sitesi (Netflix, 2019) 45
Şekil 3.5. N11.com çevrimiçi mağazası (N11.com, 2019) 46
Şekil 3.6. Gittigidiyor.com çevrimiçi mağazası (Gittigidiyor.com, 2019) 47
Şekil 3.7. *k*-EYK algoritması çalışma şekli (k=4) 48
Şekil 3.8. Genetik algoritma akış şeması 53
Şekil 3.9. Türler için oluşturulmuş kelime bulutu..... 55
Şekil 3.10. Türlerin filmler üzerindeki dağılımı 56
Şekil 3.11. Yıllara göre film dağılımları 59
Şekil 3.12. İçerik tabanlı filtreleme algoritması akış şeması 61
Şekil 3.13. Türlerin yıllara göre dağılımı..... 63
Şekil 3.14. Anahtar kelimeler için oluşturulmuş kelime bulutu..... 63
Şekil 3.15. İşbirlikçi filtreleme algoritması akış şeması 65
Şekil 3.16. Hibrit film öneri algoritması akış şeması..... 67
Şekil 3.17. Birleştirme-1 algoritması 68
Şekil 3.18. Birleştirme-2 algoritması 69
Şekil 4.1. Genetik algoritma optimizasyonu ile elde edilen OMH değişimi 80
Şekil 4.2. Genetik algoritma & birleştirme-1 algoritması için OMH değişimi..... 81
Şekil 4.3. Genetik algoritma & birleştirme-2 algoritması için OMH değişimi..... 83
Şekil 4.4. Genetik algoritma & birleştirme-2 algoritması için OMH değişimi (10 nesil) 85

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1. Kullanıcı-ürün matrisi 30
Çizelge 3.2: Ürün-özellik matrisi 34
Çizelge 3.3. Öneri sistemleri sınıflandırmalarının araştırmaları (Adomavicius ve Tuzhilin, 2005) 41
Çizelge 3.4. Öneri sistemleri uygulamaları (Schaffer, Konstan ve Reidl, 1999) 42
Çizelge 3.5. ratings.csv dosyasından ilk 20 satır 56
Çizelge 3.6. movies.csv dosyasından ilk 20 satır 57
Çizelge 3.7. tags.csv dosyasından ilk 20 satır 58
Çizelge 3.8. Tür listesi 62
Çizelge 3.9. “Toy Story (1995)” filmi için tür vektörü 62
Çizelge 4.1. Tür özniteliği kullanılarak oluşturulmuş benzerlik matrisinden bir kesit 71
Çizelge 4.2. Öznitelik hata oranları 71
Çizelge 4.3. Öznitelik kombinasyonları hata oranları 72
Çizelge 4.4. Film tabanlı işbirlikçi filtreleme deney sonuçları 74
Çizelge 4.5. Kullanıcı tabanlı işbirlikçi filtreleme deney sonuçları 74
Çizelge 4.6. Eşit katsayı kullanılan hibrit filtreleme yöntemleri 77
Çizelge 4.7. Genetik algoritma ile optimize edilmiş hibrit yaklaşımlar için OMH hata oranları 81
Çizelge 4.8. Genetik algoritma & birleştirme-1 algoritması OMH hata oranları 82
Çizelge 4.9. Genetik algoritma & birleştirme-2 algoritması OMH hata oranları 84
Çizelge 4.10. HF-10 algoritması için OMH hata oranları 85
Çizelge 4.11. HF-10 algoritması değerlendirilmesi 87

SİMGELER ve KISALTMALAR DİZİNİ

AO	:Ağırlıklandırılmış Ortalama (Weighted Average)
AAO	:Ayarlanabilir Ağırlıklandırılmış Ortalama (Adjusted Weighted Average)
ARGE	:Araştırma ve Geliştirme
BD	:Beklenti Dağılımı (Expectation Maksimation)
BK	:Bilgi Kazancı (Information Gain)
BO	:Basit Ortalama (Simple Average)
DVM	:Destek Vektör Makinaları (Support Vector Machine)
GA	:Genetik Algoritma (Genetic Algorithm)
HKOK	:Hata Karelerinin Ortalamasının Karekökü (Root Mean Square Error)
KB	:Kosinüs Benzerliği (Cosine Similarity)
KCB	:Kişisel Cep Bilgisayarı (Personal Digital Assistant)
k-EYK	:*k* En Yakın Komşu (*k* Nearest Neighbor)
NOMA	:Negatif Olmayan Matris Ayrışımı
OKF	:Ortalama Kare Farkı (Mean Squared Difference)
OMH	:Ortalama Mutlak Hata
PK	:Pearson Korelasyonu (Pearson Correlation)
SAAO	:Standart Sapmalı Ayarlanabilir Ağırlıklandırılmış Ortalama (Standart Deviation Adjusted Weighted Average)
TDA	:Tekil Değer Ayrıştırma (Singular Value Decomposition)
TF-TDF	:Terim Frekansı - Ters Doküman Frekansı (Term Frequency - Inverse Document Frequency)

1. GİRİŞ

Öneri sistemleri internette bulunan ürün veya bilgi için kullandığımız arama stilini bir anda değiştirmiş ve kolaylaştırmıştır (Miller ve ark., 2003; Kumar ve ark., 2015). Normalde bir ürün alırken çevremizdeki insanların fikirlerini alırız. Fakat bazen fikrini aldığımız kişiler bizimle aynı zevklere sahip olmayabilirler. Öneri sistemleri ürün önerirken ortak zevkler açısından kullanıcıya en çok benzeyen kişilerin tercihlerini analiz edip olası ihtiyaçları tahmin ederek bu sıkıntıya çözüm getirmektedir.

Teknolojinin hızla gelişmesi ve internetin hızla yaygınlaşması hayatın birçok alanında kolaylık ve yenilik sağlamıştır. Artık seyahat planlarken, bilet alırken, bir ürün alırken veya kitap okurken sürekli internetin sunduğu hizmetler kullanılmaktadır. Bu hizmetlere internet ortamından hızlı bir şekilde erişebiliyor olsak da verinin fazlalığı ve çeşitliliği, ürün seçimi konusunda zorlanmamıza neden olabilmektedir. Bazen beğenilmeyecek ya da ihtiyacı görmeyecek bir ürün alınarak zarar edilebilmektedir. Bazen de bir ürün aranırken çok vakit harcanabilmektedir. Böyle bir durumda tavsiyelere ihtiyaç duyulmaktadır. Böyle bir ihtiyaç çevrimiçi mağazalar başta olmak üzere bu tür hizmet veren tüm firmaları öneri sistemlerini kullanmaya mecbur bırakmıştır.

Öneri sistemleri aslında hayatın içinde kendine bir yer edinmiş durumdadır (Özgöbek ve Erdur, 2015). Amazon.com'dan daha önce bir kitap satın aldıysanız veya Netflix'te bir film izlediyseniz, muhtemelen bir öneri sistemi kullanmışsınızdır (Miller ve ark., 2003).

Öneri sistemleri, işbirlikçi filtreleme hakkında 1990'ların ortalarında ilk makalelerin yayınlanmasından beri literatürde ve çevrimiçi dünyada önemli bir araştırma konusu olmuştur. İlk öneri sistemi modeli Goldberg ve arkadaşları tarafından 1992 yılında ortaya atılan işbirlikçi filtreleme modelidir. Hem endüstri hem de akademik çevrelerde son yıllarda öneri sistemlerini geliştirmek için yeni yöntemler

denenmektedir. Çok sayıda probleme sahip olması, çevrimiçi ortamdaki veri yığınlarının kullanıcıyı zor durumda bırakması ve iş hayatındaki rekabet nedeniyle öneri sistemleri sürekli önem kazanmaktadır. (Adomavicius ve Tuzhilin, 2005).

Öneri sistemleri kullanıcılara çaba gerektirmeyen yardımlarda bulunurken, kullanıcının kişisel özniteliklerinden faydalanmaktadır. Öneri sistemleri bunu yaparken bazen de kullanıcılara kendilerinin bile farkında olmadığı yeni ihtiyaçları için ürün önererek onları şaşırtmakta ve yeni şeyler bulmalarına yardımcı olmaktadır (Özgöbek ve Erdur, 2015).

Öneri sistemleri, kullanıcıların kendilerinin girmesini istediği veriler (ürünler üzerinde oylama, puan verme, derecelendirme vb.) veya kullanıcı hakkında başka platformlardan (sosyal medya, daha önceki tercihleri vb.) elde ettiği veriler doğrultusunda kullanıcıyla ilişkilendirilmiş veriler sunmak üzere çalışmaktadır (Adomavicius ve Tuzhilin, 2005).

Öneri sistemleri kullanıcılara birçok ürün ve hizmet içerisinden en uygununu bulabilmesine yardımcı olduğu gibi satıcıların daha kolay ve daha çok ürün satmasına olanak sağlamaktadır. Büyük firmaların (Amazon, Netflix, N11, E-Bay vb.) çoğu daha fazla ürün satışı yapabilmek, daha fazla kullanıcıya ulaşabilmek veya kullanıcının sistemde daha fazla vakit geçirmesini sağlamak için öneri sistemlerini kullanmaktadırlar (Stern, Herbrich ve Graepel, 2009; Shani ve Gunawardana, 2011). Öneri sistemlerini kendi sistemlerine entegre etmeyi başaramamış firmalar sektörde kalıcı olamamaktadır (Anıl ve Akçayol, 2017).

Öneri sistemlerinin öncelikli amacı kullanıcının herhangi bir çabası olmadan, kişisel özniteliklerine göre olası ihtiyaçlarını tespit edip uygun ürünler (film, müzik, tatil, eşya vb.) tavsiye etmesidir. Öneri sistemleri internet ortamında olan çok fazla veriyle uğraşılmasına gerek bırakmadan sadece ihtiyaç olunan verileri gösterir. Çevrimiçi ortamdaki büyük verileri sadece ihtiyaçlar seviyesine indirger. Aynı zamanda alınacak hizmeti seçme konusunda da karar destek sistemleri gibi davranıp yardımcı olur. Diğer amaçlar kısaca şu şekilde sıralanabilir:

- Sistemi kullanan kişi sayısını arttırmak.
- Sistemi kullanan kişilerin sisteme tekrar tekrar girmesini sağlamak.
- İşlem sayısını veya satışları arttırarak kârlılığa katkıda bulunmak.
- Verimli ürün araması sağlamak.
- Ürün seçimlerinde yardımcı olmak.

Yukarıdaki amaçlar düşünüldüğünde bir kısmının müşteri için olduğu ve bir kısmının ise çevrimiçi hizmet sunan sistemlere hitap ettiği görülmektedir.

Öneri sistemleri pazarlamacılara ve tedarikçilere ürünlerini daha etkili bir şekilde müşteriye ulaştırma konusunda eşi görülmemiş fırsatlar sunarken, müşterilerin çok fazla seçenek arasından ve reklam karmaşasından kurtulmasını sağlar (Lekakos ve Giaglis, 2004).

Öneri sistemleri genel olarak kullanıcının büyük verilerle uğraşmasını engellediği ve şirketlere daha fazla kârlılık sağladığı için hem kullanıcı hem de e-ticaret sistemi için farklı faydalar sağlamaktadır.

Kullanıcı açısından:

- Kişiye özel servis alma
- İlgi veya ihtiyaçların bulunma süresini azaltma
- Yeni ürünler keşfetme

Servis sağlayıcı açısından

- Satış artışı
- Sistemde daha fazla vakit geçirme
- Kâr marjı yüksek ürünleri ön plana çıkarma
- İtibar artışı ve firmaya bağımlılık

Çevrimiçi ortamda artan veri yığınlarının kullanıcılar üzerindeki olumsuz etkileri ve öneri sistemlerinin bütün bu faydaları düşünüldüğünde ileriki zamanlarda öneri sistemlerinin daha da yaygınlaşması kaçınılmazdır.

Zeki optimizasyon teknikleri genelde Yapay Sinir Ağları (YSA), Genetik Algoritma (GA), Destek Vektör Makineleri (DVM) vb. gibi yoğun matematiksel işlemler kullanan ve *k* en yakın komşu algoritması (*k*-EYK) gibi nispeten daha az matematiksel işlemler gerektiren yapay zekâ algoritmalarının optimizasyon temelleri ile problem çözümleri üzerine yapılan uygulamalardır (Tan ve Bora, 2017).

GA doğadan esinlenerek ortaya çıkmış bir algoritmadır. Doğada bulunan doğal seleksiyon yani en iyinin hayatta kalması olayının algoritmik olarak ifade edilmesidir. Genetik algoritmada en iyi olan çözümler bir sonraki çözüm uzayına aktarılarak başarı oranı sürekli bir şekilde arttırılmaya çalışılır.

K en yakın komşu algoritması öğelerin birbirine olan uzaklıklarına göre birbirine yakın olan nesneleri bulmaya yarayan algoritmadır. *k*-EYK algoritması öneri sistemlerinde en çok kullanılan algoritmadır (Milli, 2013). Bu algoritmaya göre önceden hesaplanmış olan öğeler (kullanıcı veya ürün) arası uzaklıklara göre bir öğeye en yakın k tane komşu öğe seçilir. Bu en yakın komşuların tercihleri doğrultusunda aktif kullanıcının tercihleri tahmin edilmeye çalışılır. Bu amaçla tez kapsamında kullanılan öneri sitemleri yöntemlerinden İçerik Tabanlı Filtreleme (İTF), İşbirlikçi Filtreleme (İF), *K* En Yakın Komşu Algoritması ve Genetik Algoritma detaylı bir şekilde açıklanmaktadır. Bunlarla beraber kullanılan benzerlik ölçme yöntemleri ve tahmin etme yöntemleri açıklanmaktadır.

1.1. Çalışmanın Önemi ve Amacı

Gün geçtikçe internete bağlanan cihaz sayısındaki artış ile doğru orantılı olarak çevrimiçi verinin boyutu artmaktadır. Veri depolama merkezlerinin artması ve veri depolama birimlerinin ucuzlaması, kaydedilen verilerin muazzam bir şekilde artmasına neden olmaktadır (Sağıroğlu ve Koç, 2017). Depolanan verilerin bu kadar artması ile gerek internet ortamında gerekse veri depolarında aradığını bulmak giderek zorlaşmaktadır.

Bu büyük veri yığınları içinde kullanıcılar ihtiyaçlarını nerede ve nasıl bulacaklarını bilememektedir. Bu problemi çözmek ihtiyaçtan çok zorunluluk haline

geldiğinden öneri sistemleri araştırmaları giderek artmaktadır (Lekakos ve Caravelas, 2008; Anıl ve Akçayol, 2017).

Amazon, Netflix, eBay, N11, Gittigidiyor vb. çevrimiçi sektörünün önde gelen firmaları öneri sistemlerini profesyonel bir şekilde kullanmaktadırlar (Porter, 2007; Stern, Herbrich ve Graepel, 2009; Shani ve Gunawardana, 2011; MacKenzie, 2012; N11 Kurumsal, 2014; Gomez-Uribe ve Hunt, 2016; Netflix, 2016; Aimgroup, 2016; Gittigidiyor Kurumsal, 2017).

Öneri sistemleri kullanıcıların farklı özniteliklerini değerlendirerek erişilen içerik ile kullanıcı arasında bir ilişki kurmaktadır. Daha sonra önerilecek olan içerik (ürün, görsel, makale, film, video vb.) belirli kriterlere göre değerlendirilip, kullanıcıya özgü biçimde önerilmektedir. Bu sayede kişi internetteki veri yığınları içerisinde aradığını daha kolay bulabilmektedir (Anıl ve Akçayol, 2017).

Kullanıcı sadece aradığını değil bazen de kendisine lazım olabilecek içerikleri daha aramadan bile bulabilmektedir. Örneğin 3 aylık bebek bezi siparişi veren bir kullanıcıya 4 ay sonra kullanıcı daha arama yapmadan 7 aylık bebekler için bebek bezi önerilebilmektedir.

Bu tez çalışmasında kullanımı giderek yaygınlaşmakta olan öneri sistemleri teknolojileri için çok elverişli olan ve sıklıkla kullanılan film sektöründe bir uygulama yapılmıştır. Öneri sistemleri yöntemlerinden İşbirlikçi Filtreleme ile İçerik Tabanlı Filtreleme bir arada kullanılarak hibrit bir yaklaşım sunulmuştur. Hâlihazırda işbirlikçi filtreleme ve içerik tabanlı filtreleme birçok sistemde kullanılmaktadır. Her sistemde olduğu gibi öneri sistemleri de geliştirilmeye ihtiyaç duymaktadır. Yeni bir hibrit birleştirme yöntemi önerilerek daha düşük hata oranlarına sahip bir öneri sistemi tasarlanmaktadır.

Yapılan tez çalışmasının bundan sonraki kısımlarında sırasıyla Önceki Çalışmalar bölümünde daha önce yapılmış bazı öneri sistemlerinden bahsedilmekte; Materyal ve Yöntem bölümünde öneri sistemleri, kullanılan değerlendirme metrikleri,

önerilen yöntemler detaylı bir şekilde anlatılmakta ve öneri sistemlerini kullanan bazı çevrimiçi mağaza örnekleri verilmekte; Araştırma Bulguları ve Tartışma bölümünde önerilen yöntemler yapılan deneyler karşılaştırılmıştır. Tezin son bölümü olan Sonuçlar ve Öneriler bölümünde yapılan deneyler sonucunda elde edilen sonuçlar verilmiş ve ileride yapılabilecek çalışmalardan bahsedilmiştir.

2. ÖNCEKİ ÇALIŞMALAR

Öneri sistemlerinin başarısının ölçülmesindeki belirsizlik, farklı sektörlerin ihtiyaçlarının farklı olması ve katlanarak artan çevrimiçi verinin yeni yöntemlere olan ihtiyacı gibi nedenlerden dolayı akademik çevrelerde ve iş hayatında birçok öneri sistemi önerilmiştir (Burke, 2002). Halihazırda kullanılan öneri sistemleri literatürde aktif bir araştırma alanı olmakla beraber, birçok firmanın da ARGE (Araştırma ve Geliştirme) çalışmalarına konu olmuştur (Stern, Herbrich ve Graepel, 2009).

Mukherjee ve ark. (2001), kullanıcı tercihlerini yakalayan web tabanlı bir film öneri sistemi geliştirmişlerdir. Bu öneri sistemi (Movies2Go) yanıt süresini azaltmak için film bilgilerini de önbelleğe alarak çalışmaktadır. Filmlerin oyuncu, yönetmen gibi özniteliklerinin benzerlik hesaplamasındaki ağırlıklarını bulmak için oylama temelli sıralama prosedürünü işleyen öğrenmeye dayalı bir yaklaşımla birleştirilir. Bu sayede eğer kullanıcı için yönetmen yapım yılından daha öncelikliyse yönetmen özelliğinin ağırlığı yapım yılından daha yüksek bir değer alınabilmektedir. Yeni kullanıcı sorununa çözüm için sisteme ilk defa giren kullanıcıya olası en çok tercih edilen birkaç filmi oylaması sunulmuştur. Yapılan web tabanlı film öneri sisteminde önerilen filmlerden sonra kullanıcılardan geri dönüş yapmaları istenmiştir. Çalışma sonucunda filmler için kullanılan özniteliklerin ağırlıklandırılmasının doğruluk oranına etki ettiği gösterilmiştir.

Öneri sistemlerinin doğruluk oranları gibi kullanıldıkları cihazlar da çok önemlidir. Miller ve ark. (2003), Kişisel Cep Bilgisayarı (KCB) için geliştirdikleri bir film öneri sistemi ile taşınabilir cihazlarda öneri sistemlerinin etkinliğini incelemişlerdir. Yaptıkları uygulama MovieLens veri setini kullanmaktadır. İki senaryo üzerinden geliştirilen sistemde ilk senaryo; şu an gösterimde olan filmler içinde tavsiye isteyen kullanıcılar içindir. Bu senaryoda film önerilirken filmin oynatıldığı sinema salonlarının kullanıcıya olan uzaklıkları hesaba katılarak algoritma geliştirilmiştir. İkinci senaryo; bir film kiralamak veya satın almak isteyen kullanıcılar içindir. Bu senaryoda filmlerin yayınlanma tarihlerine göre (film ne kadar yeni ise

ağırlık oranı daha fazladır) öncelik tanınarak önerilerde bulunulmuştur. Araştırma sonucunda kullanıcıların taşınabilir bir cihaz üzerinden alınan önerileri daha çok dikkate aldıkları ve artık bilgisayarlarda film önerisi aramayı azalttıkları görülmüştür.

Zhang ve Koren (2007), soğuk başlangıç problemine bir çözüm getirmek için içerik tabanlı filtrelemede kullanmak amacıyla oluşturulan kullanıcı profili için Bayes hiyerarşik modelini önermişlerdir. İçerik tabanlı filtreleme için kullanıcı profili oluşturulması önemli bir sorundur, özellikle az sayıda değerlendirme yapmış bir kullanıcıya kişisel önerilerde bulunmanın anahtarıdır. Milyonlarca kullanıcıya hizmet veren bir sistemin, bir Bayes hiyerarşik modelini kullanıp diğer kullanıcılardan bilgi alarak, yeni bir kullanıcı veya daha az geribildirim alan bir kullanıcı için daha iyi bir kullanıcı profili oluşturulabileceği söylenmiştir. Bu çalışmada Beklenti Dağılımı (BD) algoritması tabanlı öğrenme yaklaşımının Bayes hiyerarşik doğrusal modelleri için zayıflığı incelemiştir ve Modified EM adı verilen daha iyi bir öğrenme tekniği önerilmiştir. Önerilen tekniğin teorik olarak standart BD algoritmasından daha hızlı ve daha verimli olduğu ispatlanmıştır. MovieLens ve Netflix veri setleri üzerinde yapılan deneyler, verilerde seyreklik sorununun yaşandığı durumlarda önerilen tekniğin daha başarılı olduğunu göstermiştir.

Son zamanlarda yapay zekâ topluluğu üyeleri arasında, yapay zekânın internet bilgi depolarında veri madenciliği, bilgi arama gibi alanlarda etkinliğinin artmasının verimi arttıracağı düşünülmektedir. Christakou ve Stafylopatis (2007), yaptıkları "Yapay Sinir Ağları Tabanlı Hibrit bir Film Öneri Sistemi" isimli makalede yapay sinir ağlarının film öneri sistemlerinde başarılı sonuçlar verebileceğini göstermişlerdir. MovieLens veri seti kullanılarak önerilen hibrit yaklaşımda içerik tabanlı filtreleme ve işbirlikçi filtreleme yaklaşımlarını yapay sinir ağları ile ayrı ayrı gerçekleştirdikten sonra ortaya çıkan öneriler birleştirilmiştir. İçerik tabanlı filtrelemede film türü, kişileri (aktör senarist, yönetmen vb.) ve özeti üç giriş sinir ağı olarak kullanılmıştır. Yapay sinir ağları modelinde film puanı 4-5 olanlar başarılı, 1-3 olanlar başarısız olacak şekilde ikili bir sonuç kullanılmıştır. İşbirlikçi filtrelemede benzerlikler Pearson Korelasyonu (PK) ile hesaplanmıştır. Bulunan benzerlikler daha sonra yapılan tahminleme aşamasında kullanılmış olup 1-3 arası puanı olanlar bir gurup, 4-5 arası

puanı olanlar bir gurup kabul edilmiştir. Önerilen hibrit yaklaşımın çıktısı olan 5 film birden farklı adımla seçilmiş olsalar bile genel olarak 2 tanesi içerik tabanlı filtrelemeden, 2 tanesi işbirlikçi filtrelemeden ve 1 tanesi de içerik kriterleri dikkate alınarak içerik tabanlı filtrelemeden alınmıştır. Önerilen hibrit yaklaşım başka makale çalışmaları ile karşılaştırılmış ve başarılı sonuçlar almışlardır.

Lekakos ve Caravelas (2008), web tabanlı bir film öneri sistemi (MORE) tasarlamışlardır. Bu çalışmada içerik tabanlı ve işbirlikçi filtreleme yöntemleri bir araya getirilerek hibrit bir öneri sistemi önerilmiştir. İçerik tabanlı filtrelemede filmler için "oyuncular, yönetmenler, yazarlar ve yapımcılar" öznitelikleri kullanılmıştır. İşbirlikçi filtreleme ürün (film) tabanlı benzerlikler üzerine kurulmuştur. Sistem aktif olarak çalışırken performans sorunları yaşanmaması için yoğun matematiksel işlemler gerektiren benzerlik hesaplamaları işbirlikçi filtreleme kısmı için Pearson Korelasyonu ve içerik tabanlı filtreleme için Kosinüs Benzerliği (KB) hesaplamaları önceden yapılmıştır. Yeni bir kullanıcı sisteme kayıt olur olmaz profilini oluşturmak için bir dizi değerlendirme yapması istenmektedir. Yapılan çalışma sonucunda Ortalama Mutlak Hata (OMH) değerleri karşılaştırılmış ve işbirlikçi filtrelemenin hala en başarılı filtreleme yöntemlerinden biri olduğu kanıtlanmıştır.

Yakın gelecekte popülerliği artacağı düşünülen en önemli teknolojilerden biri olan Semantik web teknolojileri akademik ve endüstriyel çevrelerde önemli bir yere sahiptir. Semantik web teknolojilerinin yaygınlaşması sonucu çevrimiçi ortamda bulunan veriler üzerinde gerçek insanlar tarafından yapılan işlemler artık bilgisayarlar tarafından yapılacak olup veriler arasındaki ilişkiler tespit edilip yine bilgisayarlar tarafından bu veriler üzerinde yeni ilişkiler ortaya çıkarılabilecektir. Bu sayede televizyon kanalları izleyicilerine kişiselleştirilmiş televizyon izleme tecrübeleri, kişiye özel reklamlar ve doğrudan TV ekranından çalışabilecek istemciler kullanılarak kişileştirilmiş TV programı öneri sistemleri sunulabilecektir. Battal (2009), Türkiye Radyo ve Televizyon Kurumu'nun (TRT) yayın akışı için semantik web tabanlı bir program öneri sistemi geliştirilmiştir. Bu sayede izleyicilere kişiselleştirilmiş program akışları önerilebilecektir. Yapılan çalışma ile semantik web tabanlı bir öneri sisteminin tv kanallarında başarılı olabileceği gösterilmiştir.

Hemen hemen herkesin internete erişim imkânı bulduğu günümüzde, internette yararlı bilgiler bulmak giderek zorlaşmaktadır. Web öneri sistemleri, kullanıcılara sunulan bilgi hacminin çok büyük olduğu bu karmaşık bilgi alanında kararlar almalarında yardımcı olmaktadır. Bu sistemlerde genelde kullanıcının alışkanlıkları, beğenileri veya ihtiyaçları doğrultusunda profili oluşturulmaktadır. Bu konuda birçok öneri sistemi geliştirilmiştir. Bununla birlikte her sistemin kendine özgü problemleri veya kısıtlamaları mevcuttur. Göksedef ve Gündüz Öğüdücü (2010), bir hibrit öneri sistemi geliştirmişlerdir. Bu sistem veri madenciliğine dayanan birkaç öneri tekniğinin sonuçlarının birleştirilmesidir. Farklı öneri tekniklerinin hibrit bir şekilde kullanılmasının geleneksel yöntemlerden daha iyi sonuçlar verdiği görülmüştür. Bu hibrit yöntemlerin öneri sistemlerinin tahmin doğruluğunu nasıl etkilediğine ilişkin ayrıntılı bir karşılaştırmalı değerlendirme yapılmıştır. Bu çalışma siteminde hibrit sistemlerin diğer öneri sistemlerinden daha başarılı olduğu sonucuna varılmıştır.

Uluyağmur (2012), yaptığı "Hibrit Film Öneri Sistemi" isimli tez çalışmasında film içeriği, oyuncu, yönetmen ve tür bilgileri kullanılarak oluşturulmuş içerik tabanlı filtreleme ile kullanıcı puanlamasını baz alan bir işbirlikçi filtreleme bir araya getirilmiştir. İçerik tabanlı filtreleme algoritmasında özniteliklerin ağırlıklandırılması için kullanıcılardan alınmış puanlandırmalar kullanılmıştır. İşbirlikçi filtreleme de dolaysız matris ayrıştırma kullanılmıştır. İçerik tabanlı filtrelemeden gelen tahmin ve işbirlikçi filtrelemeden gelen tahmin doğrusal bir modele göre birleştirilmiştir. Oluşturulan hibrit öneri sistemi işbirlikçi filtreleme ve içerik tabanlı filtreleme ile oluşturulan öneri sistemine göre daha başarılı sonuçlar elde edilmiştir.

Milli (2013), yaptığı "Film Öneri Sistemleri İçin Hibrit Bir Yöntem Geliştirilmesi" isimli tez çalışmasında film öneri sistemlerindeki gelişmelere dikkat çekmek için yeni bir öneri yaklaşımı sunmuştur. Önerilen içerik tabanlı filtreleme yaklaşımında filmler için kullanılan öznitelik kümesinin (tür, yıl, yönetmen, aktör, yapımcı, maliyet) tüm elemanları kullanılmamıştır. Doğruluktan fazla ödün vermeden tahminleme hızını arttırmak için kullanıcıların film tercihlerinde en çok rol oynayan "tür" özniteliği tek kullanılmıştır. Benzerlik hesaplamasında Pearson korelasyon ve kosinüs benzerlik ölçme yöntemleri kullanılmıştır. Kullanıcıların pozitif benzerlikleri

gibi negatif benzerlikleri de hesaba katılmıştır. Oluşturulan hibrit yöntemde ilk önce işbirlikçi filtreleme yaklaşımı ile bir sonuç elde edilmiştir. Daha sonra içerik tabanlı filtreleme ile oluşturulan kullanıcı profili ve ürün profili arasındaki benzerlik hesaplaması ve bazı matematiksel işlemler sonucunda bulunan bias değeri işbirlikçi filtreleme ile bulunan sonuca eklenmiştir. Yapılan deneysel karşılaştırmalar sonucunda önerilen hibrit modelin geleneksel algoritmalardan daha iyi sonuç verdiği ispatlanmıştır.

Şimşekli ve ark. (2013), çevrimiçi servislerde ayrışım tabanlı öneri sistemlerinin nasıl kullanılacağı ve Negatif Olmayan Matris Ayrışımı (NOMA) tabanlı bir sistemde seçilecek model parametrelerinin neye göre olacağı üzerine çalışma yapmışlardır. NOMA modeli ayrışım tabanlı öneri sistemlerinde, çok sık kullanılan modellerden biridir. Bu çalışmada bir yemek sipariş servisinden elde edilen verilerden bazıları rastgele olarak silinmiştir. Daha sonra NOMA modeli ile bu eksik veriler tahmin edilmeye çalışılmıştır. Böylelikle Noma modelinin eksik verileri tahmin etme de ne kadar başarılı olduğu gösterilmiştir

İşbirlikçi filtreleme yöntemi zamanla değişebilen özelliklere cevap vermekte yetersiz kalmaktadır. Çünkü bu filtreleme modelinde daha önceki beğeniler, zevkler zamana göre şekillenmez. Halbuki kullanıcının beğenileri, zevkleri zaman içerisinde değişiklik gösterebilmektedir. Bu sebeple öneri sistemlerinde "zamansal değişiklik (temporal dynamics)" önemli bir kavram haline gelmektedir. Bakır ve Albayrak (2014), kullanıcıların zamana bağlı olarak değişebilen özelliklere göre kişileştirilmiş öneriler yapmak için yeni bir yöntem geliştirmişlerdir. Geliştirilen bu yeni yöntemde geleneksel kullanıcı tabanlı İşbirlikçi filtrelemeden farklı olarak kullanıcıların ürünlere verdikleri puanlar tarihlere göre değerlendirilip sonraki tarihler için yeni puanlar tahmin edilmeye çalışılmıştır. Öneri sistemlerinde bir ürünün puanı, bu puanın verildiği tarihin güncelliğine göre önem kazanmaktadır. Sistemdeki bir ürünün eski bir tarihte verilen puanı çok önemli olmayıp yakın tarihte verilen puanı daha önemli kabul edilmektedir. Bu sebeple puanların yaşları hesaplanmış ve puan yaşı yükseldikçe değerinin düştüğüne karar verilmiştir. Bu çalışma, puanların yaşlarının dikkate alınarak verilen tavsiyelerin doğruluk oranlarının daha yüksek olduğunu göstermiştir.

Kumar ve ark. (2015), MOVREC adlı bir öneri sistemi önermişlerdir. Kullanıcılar tarafından sağlanan bilgileri kullanıp, bunları analiz ederek kullanıcılara önerilerde bulunan işbirlikçi filtreleme yaklaşımına dayanan MOVREC sistemi web tabanlı PHP programlama dili ile tasarlanmıştır. Yeni kullanıcı sorunu sisteme ilk girişte aktör, yönetmen, tür, yıl, reyting vb. bir dizi özellik seçtirilerek kısmen çözülmüştür. Farklı niteliklerin farklı ağırlıkları vardır. MOVREC sisteminde üretilebilecek en uygun önerilerin önceki kullanıcıların filmlere yaptığı değerlendirmelere dayanması gerektiği tespit edilmiş, bu nedenle de değerlendirme niteliğine diğer özelliklerden daha fazla önem verilmiştir. Sistemi kullanmak isteyen kullanıcı bir özelliğe göre arama yapmak istediğinde veri tabanından uyuşan filmler getirilirmiştir. Film sayısı yirmiden az ise tüm ilişkili derecelendirmeler sıralanmış şekilde; yirmiden fazla ise farklı niteliklerin kümülatif ağırlığına göre çalışan K-means algoritması kullanılarak filtrelenen ilk yirmi film gösterilmiştir.

Günümüzde benzer içeriklere sahip çok sayıda haber kaynağı bulunmaktadır. Bu kadar çok kaynak arasından kısa bir zaman içerisinde sadece ihtiyaç duyulan makalelerin bulunup okunması neredeyse imkânsızdır. Çevrimiçi ortamda ulaşılan haber kaynağının gerçekten ihtiyaç duyulan veriyi içinde barındırıyor olması ve bunun okuyucu tarafından teyit edilmesi gerekmektedir. Taşçı (2015), "İçerik Tabanlı Medya Takip ve Haber Tavsiye Sistemi" isimli tez çalışması ile çevrimiçi haber kaynaklarından elde edilen dokümanları değişik veri işleme adımlarına tabi tutarak okuyucuların tercihlerine veya özelliklerine göre sınıflandırma ve filtreleme işlemleri uygulayarak kişiselleştirilmiş bir haber öneri sistemi yapmayı amaçlamıştır. Bu tezde öneri sistemlerinin başlıca yöntemlerinden olan içerik tabanlı filtreleme kullanılmıştır.

Aygün (2016), "Genetik Algoritma Kullanılarak İçerik Tabanlı Kitap Tavsiye Sistemi Geliştirilmesi" isimli tez çalışmasında önerdiği kitap öneri sisteminin tahmin etme oranları çok iyi sonuçlar vermiştir. Sistem kullanıcının daha önceden okuduğu veya beğendiği kitapları göz önünde bulundurarak kullanıcıya yeni kitaplar önermektedir. Uygulama bu yönüyle kişiye özgü tavsiyeler üretmektedir. Bu sayede kullanıcı sadece popüler olan veya başkalarının önerdiği kitapları değil kendisinin okuma profiliyle örtüşen kitaplarla buluşabilmektedir. Bu kitap öneri sistemi

kullanıcının daha önceki tercih ve beğenilerini kullanarak kişinin profilini çıkarmaktadır. Daha sonra bu profille örtüşen önerilerde bulunulmaktadır. Kitap önerileri haber, müzik, film gibi çabuk tüketilen öneriler olmayıp, daha uzun zaman aralıklarıyla değerlendirilen ve zamana göre çok değişiklik göstermeyen tavsiyelerdendir.

Çevrimiçi cihaz sayısında artış ve büyük veri depolarının çoğalması ile web üzerinde toplanan bilgi miktarı da büyük bir artış göstermektedir. Bu verilerin büyük çoğunluğu sosyal ağlarda üretilmektedir. Bu durum karmaşık veri yığınlarına sebep olduğu gibi faydalı birçok verinin de gerçek hedefine ulaşmamasına neden olmaktadır. Bazen de karşılaşılan bu veriler zihinsel yorgunluğa neden olabilmekte ve yanlış yönlendirmelere sebep olabilmektedirler. Bütün bu sebeplerden ötürü sosyal ağlar üzerinde faydalı olabilecek doğru bilginin nasıl elde edileceği konusunda birçok çalışma yapılmaktadır. Sevli ve Küçüksille (2016), yapmış oldukları çalışma ile sosyal ağlardaki analiz ve öneri sistemlerindeki sorunlardan yola çıkarak, kullanıcının ilgi alanlarını dinamik olarak analiz edip kullanıcının profiline en yakın, takip edilebilecek diğer kullanıcıları tavsiye edebilecek bir sistem geliştirmişlerdir. Uygulama alanı olarak çok hızlı bir şekilde büyüyen Twitter tercih edilmiştir. Yapılan öneri sisteminin geliştirilmesinde Türkçe doğal dil işleme, tabu arama yaklaşımı, büyük veri analiz teknikleri ve şartlı olasılık teoremi kullanılmıştır. Geliştirilen sisteme "Tweetlyzer" ismi verilmiştir. Tweetlyzer uygulaması çevrimiçi ortamdaki tüm kullanıcıların kullanılması için yapılmıştır. Sistemi test etmek için Twitter kullanan aktif yüz kişilik bir kullanıcı gurubu belirlenmiştir. Testler bu gurup üzerinde gerçekleştirilmiştir. Kullanıcılar tarafından yapılan geri bildirimler neticesinde sistemin %86 oranında bir başarı elde ettiği görülmüştür.

Laveti ve ark. (2016), hibrit benzerlik metrik modeli kullanan bir öneri sistemi önermişlerdir. Bu modelde Pearson korelasyon, log olasılık ve Tanimoto katsayısı gibi iki veya daha fazla geleneksel benzerlik ölçümünü birleştirmişlerdir. Bu öneri sisteminin değerlendirilmesinde 100 bin değerlendirme, 1 milyon değerlendirme ve 10 milyon değerlendirmeden oluşan üç farklı MovieLens veri seti kullanılmıştır. Hibrit benzerlik metrik modeli kullanan bu öneri sistemi ile yukarıda belirtilen veri setlerinin

gerçek zamanlı yürütülmesinden çıkarımlar yapılmıştır. Yapılan deneyler ile önerilen sistemin geleneksel öneri sistemlerinden daha iyi sonuçlar verdiği görülmüştür. Bu çalışma öneri sistemlerinin daha verimli çalışması için giderek daha karmaşık bir hal aldığını ve yoğun matematiksel işlemler gerektirdiğini ispatlar niteliktedir.

Öneri sistemlerinin geliştirilmesinde kullanılan tekniklerden biri kümelemedir. Kümeleme aynı kümelerdeki nesnelerin, diğer kümelerdekilerden daha çok benzer olacak şekilde gruplanma işlemidir. Ahuja ve ark. (2019), yaptıkları çalışmada K-Means Kümeleme ve K En Yakın Komşu algoritmaları kullanılan bir film öneri sistemi sunmuşlardır. MovieLens veri seti üzerinde oluşturulan farklı kümeler ile çalışma yapılmıştır. Küme sayısının azalmasının HKOK hata oranını düşürdüğü tespit edilmiştir. Önerilen yöntemin HKOK değerinin, mevcut teknik ile aynı değere ulaştığında daha fazla küme içerdiği de tespit edilmiştir. Çalışmada önerilen yöntemin mevcut yöntemden daha başarılı olduğu ampirik kanıtlanmıştır.

Çevrimiçi tavsiye sistemleri, e-ticaretin ayrılmaz bir parçasıdır. Farklı yaklaşımları takip eden çok sayıda algoritma vardır. Patra ve Ganguly (2019), Tekil Değer Ayrıştırma (TDA) algoritmasını *k*-EYK algoritması kullanarak iyileştirmeye çalışmışlardır. TDA, boyut azaltma teknikleri kullanarak filmlerin altında yatan bazı gizli kalıpların anlaşılmasını kolaylaştırmaktadır. Ancak hesaplama ve kaynak açısından düşük performans göstermektedir. Bu çalışmada MovieLens 1M veri seti üzerinde TDA ile ilgili birkaç ön işleme algoritması arasında karşılaştırmalı bir çalışma yapılmıştır. Öneri sistemlerinin sahip olduğu seyreklik sorununa bir çözüm olarak önerilen TDA, *k*-EYK algoritması sayesinde daha da geliştirilmeye çalışılmıştır. K-EYK tabanlı yaklaşım, kullanıcıların en yakın *k* komşularını bulmak için kullanılmış ve değerlendirmeleri daha sonra eksik değerleri bulmak için kullanılmıştır. Eksik değerler benzer kullanıcıların ortalaması kullanılarak hesaplandığında ve benzerlik hesaplamasında Öklid kullanıldığında *k*-EYK tabanlı TDA'nın en iyi performansı gösterdiğini tespit edilmiştir.

3. MATERYAL ve YÖNTEM

3.1. Öneri Sistemleri

Öneri sistemleri akıllı bir bilgi filtreleme yöntemidir. Kişiselleştirilmiş bir filtreleme motoru olan öneri sistemleri bilgiyi ararken harcanan süreyi kısaltarak karar verme sürecini azaltır. Öneri sistemleri kişilerin profillerini oluştururken iki farklı yolla bilgi toplar. Bunlar açık ve kapalı işlemlerdir. Açık işlemler kullanıcının sistem (uygulama, web sitesi, tv programları vb.) üzerindeki tercihlerinden (değerlendirme, yorum vb.) oluşmaktadır. Kapalı işlemler ise kullanıcının sistemde geçirdiği zaman dilimindeki davranışları veya geçmiş işlemlerinden oluşmaktadır. Öneri sistemleri bu iki yöntem ile elde edilen verileri işleyerek kullanıcı ya da müşteri için kişiselleştirilmiş tavsiyelerde bulunur.

Öneri sistemleri kullanılmaya başlandığı ilk zamanlarda çok daha basit bir yapıdaydı. Sadece belli başlı noktalara dikkat edilmekteydi. Bunlar önerilecek ürünün daha önce tercih edilip edilmediği veya en çok tercih edilen ürün olma özelliği gibi kriterlerdi (Adomavicius ve Tuzhilin, 2005).

Öneri sistemleri günümüzde çok daha karmaşıklaşmış olup yapay zekâ, makine öğrenmesi veya sınıflandırma algoritmaları kullanılarak kullanıcının birden fazla özelliği göz önüne alınıp kullanıcı ile önerilecek ürün (müzik, film, makale, eşya vb.) arasında ilişki kurmaya çalışmaktadır (Shani ve Gunawardana, 2011)

Şüphesiz öneri sistemlerinin kullanıcıyı memnun etme ve sistemde daha fazla zaman geçirip yeni ürünler alması üzerinde çok etkili olduğu bellidir. Müşteri, sistem kendisine aradığı şeyi bulmada yardımcı olduğu için tercih ederken; ticari siteler ise kullanıcı sayısını ve doğal olarak satışı arttırıp daha fazla kâr elde etmek için öneri sistemlerini tercih etmektedirler.

Bir öneri sistemi üç öğeden oluşmaktadır. Bunlar kullanılan veri seti, öneri motoru ve kullanıcı arayüzüdür.

- Veri setleri daha önceden hazırlanmış olan, kullanıcının davranışlarından veya tercihlerinden oluşan bilgi yığınıdır. Bu bilgiler açık veya kapalı işlemler ile elde edilmektedir.
- Öneri motoru veri seti üzerinden gerçekleştirilen işlemler sonucu kullanıcıya öneriler yapılmasını sağlayan işlemler dizinidir. Öneri motoru benzerlik hesaplama, komşuluk bulma ve tahmin yapma aşaması olarak üç aşamadan oluşmaktadır.
- Arayüz ise önerileri kullanıcılara göstermek ve onlardan bu konuda geri bildirim almak için kullanılan uygulamalardır.

Öneri sistemleri kullandıkları öneri yaklaşımına göre üç başlık altında toplanabilir. Bu yaklaşımlar işbirlikçi filtreleme, içerik tabanlı filtreleme ve bu ikisinin farklı şekillerde bir araya getirildikleri hibrit filtrelemelerdir (Adomavicius ve Tuzhilin, 2005).

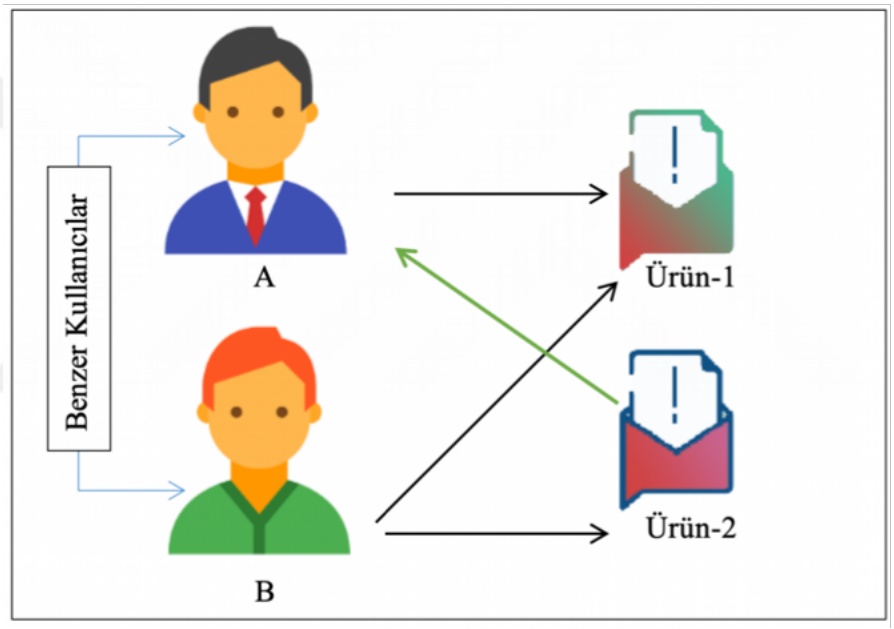
3.1.1. İşbirlikçi filtreleme

En eski yöntem olmasına rağmen en çok kullanılan yöntemdir (Taşçı, 2015). Bu filtreleme türünde öğeler (kullanıcı veya ürün) arasında benzerlik hesaplamaları yapılır. Bu benzerlikler yardımı ile öneriler yapılır. Eğer iki kullanıcı aynı ürünü beğenmiş ise muhtemelen birinci kullanıcının beğendiği ve ikinci kullanıcının henüz görmediği ürün ikinci kullanıcı tarafından da beğenilecektir prensibi ile çalışmaktadır (Su ve Khoshgoftaar, 2009). Aktif kullanıcının daha önce izlediği ve yüksek puan vererek değerlendirdiği filmleri izleyip beğenen başka bir kullanıcının aktif kullanıcıya benzer olduğu kabul edilir. Daha sonra benzerlik katsayılarına göre izlemediği filmler önerilir.

İşbirlikçi filtreleme 1990'ların başlarından beri kullanılmaktadır (Adomavicius ve Tuzhilin, 2005). İşbirlikçi filtreleme önerilerini kullanıcının tercih ettiği ürünlere

dayandırdığından güncelliğini sürekli korumaktadır (Taşçı, 2015). İşbirlikçi filtrelemenin, belirli kriterlerin (komşu sayısı ve değerlendirilmiş ürün sayısı) karşılanması şartıyla içerik tabanlı filtrelemeden daha doğru sonuçlar elde ettiği ampirik olarak gösterilmiştir (Alspector ve ark., 1997; Basu ve ark., 1998).

İşbirlikçi filtrelemenin çalışma mantığı Şekil 3.1.'de görülmektedir. Şekil 3.1.'de A ve B kullanıcıları, Ürün-1'i satın almıştır. Bu iki kullanıcı arasında bir benzerlik olduğunu gösterir. B kullanıcısının satın almış olduğu Ürün-2, henüz bu ürünü almamış olan A kullanıcısına önerilebilir.



Şekil 3.1. İşbirlikçi filtreleme

İşbirlikçi filtreleme iki farklı şekilde uygulanabilmektedir. Bunlar kullanıcı tabanlı ve ürün (film, müzik, eşya vb.) tabanlı işbirlikçi filtrelemedir. İşbirlikçi filtrelemede benzerlik hesaplamaları [kullanıcı X ürün] matrisi üzerinde yapılmaktadır. Kullanıcı tabanlı işbirlikçi filtrelemede benzerlik hesaplaması matrisin satırları (kullanıcılar) arasında yapılırken, ürün tabanlı işbirlikçi filtrelemede matrisin sütunları (ürünler) arasında yapılır (Sarwar ve ark., 2001).

İşbirlikçi filtreleme bazı sorunlara sahiptir. Bunlar başlıca soğuk başlangıç problemi, işlem maliyeti, shilling ataklar (sahte profil saldırıları) ve profil darlığı sorunlarıdır.

İşbirlikçi filtreleme yeni kullanıcılara ilk tavsiyelerde bulunmakta zorlanır. Bu problem soğuk başlangıç (Cold Start) problemi olarak tanımlanmaktadır (Taşçı, 2015). Sisteme yeni katılmış olan bir kullanıcı henüz herhangi bir ürün hakkında bir değerlendirmede bulunmamıştır. Bu durum kullanıcı tarafında yaşanan bir sıkıntıdır. Soğuk başlangıç problemi yeni ürünlerde de yaşanmaktadır. Yeni olan bir ürünün henüz bir alıcısı olmadığı için benzerlik hesaplamaları yapılamaz (Taşçı, 2015). Dolayısıyla bu ürünler de herhangi bir kullanıcıya önerilemez. Bu çevrimiçi mağaza tarafında yaşanan bir dezavantajdır.

İşbirlikçi filtreleme ürün ve müşteri yelpazesi genişledikçe daha karmaşık hesaplamalara ihtiyaç duymaktadır. Bu durum sistemin istenilen sürede istenilen öneriyi yapamamasına neden olmaktadır (Taşçı, 2015). Bu problem için Boyut Azaltma (Dimension Reduction) teknikleri önerilmiştir (Wang ve ark., 2014).

İşbirlikçi filtreleme algoritmaları shilling ataklar, profil enjeksiyon atakları veya sahte profil saldırıları olarak adlandırılan saldırılara karşı savunmasızdırlar. Sahte kişi veya şirket profili üzerinden istenilen (kâr marjı yüksek ürün) ürünün popülaritesi artırılarak ön plana alınabilirken istenmeyen (kâr marjı düşük ürün) ürünler için düşük değerlendirmeler yapılarak popülaritesi düşürülebilir (Lam ve Riedl, 2004; Bilge, Batmaz ve Polat, 2016). İşbirlikçi filtreleme sistemleri genel olarak herkese açık sistemler olduklarından bu tarz kötü niyetli profillere engel olma konusunda yetersiz kalmaktadırlar. Yapılan çalışmalar ile ortaya atılan algoritmalar sayesinde bu sahte profiller tespit edilmeye çalışılmaktadır. Bu sayede sistem bu saldırılardan kısmen korunabilmektedir. Fakat tüm bu algoritmaların henüz tam olarak tüm saldırıları tespit etme konusunda başarılı oldukları söylenemez (Lam ve Riedl, 2004).

İşbirlikçi filtrelemenin bir diğer dezavantajı kullanıcı profillerinin dar olmasıdır. Bunun ana nedeni çevrimiçi sistemlerin kullanıcılardan yeterli geri bildirim

alamamasıdır. Kullanıcı profili dar olduğunda sınıflandırma yapmak zor olacağından öneri sistemleri başka alanlardan yararlanmaktadırlar. Öneri sistemleri bu tür durumlarda kullanıcının demografik özelliklerini kullanarak önerilerde bulunabilir. Demografik filtrelemede kullanıcıların cinsiyet, yaş, eğitim, ırk, meslek ve yaşadıkları şehir gibi demografik bilgiler kullanılmaktadır. Bu yöntem sayesinde oluşturulan profillere göre önerilerde bulunulmaktadır. Demografik filtrelemenin en büyük dezavantajı kullanılan çoğu özniteliğin (cinsiyet, meslek, doğum yeri vb.) zamanla değişmemesidir. Değişmeyen bu özniteliklerin aksine kullanıcıların ilgi alanları veya ihtiyaçları zamanla değişebilmektedir (Taşçı, 2015).

İşbirlikçi filtreleme algoritması üç aşamadan oluşmaktadır. Bu aşamalar sırasıyla işlenerek en sonunda işbirlikçi filtreleme algoritmasının çıktısı (tahminini) elde edilmektedir. Bu aşamalar şunlardır: (Lekakos ve Caravelas, 2008)

- Benzerlik Hesaplama
- Komşuluk Seçimi
- Tahmin Hesaplama

İşbirlikçi filtreleme sistemleri bu aşamaları sırayla işlemektedir. Birinci aşama olan benzerlik hesaplama aşamasında işbirlikçi filtreleme sisteminin türüne göre ya ürünler arasındaki (ürün tabanlı işbirlikçi filtreleme) ya da kullanıcılar arasındaki (kullanıcı tabanlı işbirlikçi filtreleme) benzerlikler hesaplanır. Daha sonra k-EYK, DVM, YSA vb. yapay zekâda kullanılan sınıflandırma algoritmalardan birisi ile benzerlik oranları doğrultusunda bu öğeye en yakın komşular belirlenir. Son aşamada ise bulunan komşular sayesinde algoritma tahminler üretir. Her aşamada birden çok yöntem bulunmaktadır (Milli, 2013). İşbirlikçi filtreleme sistemlerinin aşamaları sırasıyla açıklanmaktadır.

3.1.1.1. Benzerlik yöntemleri

İlk aşama olan benzerlik hesaplamada bulunacak olan benzerlik oranları bundan sonraki aşamalar için çok önemlidir. Bu aşamada kendisi için tahmin yapmak

istediğimiz öğe (kullanıcı veya ürün) ile diğer öğeler arasında hesaplanan benzerlikler bu öğelerin tahmin yapma aşamasında ne derece etkili olacağını belirlemektedir. Benzerlik oranı en yüksek olan öğe aktif öğe için yapılacak tahminde en etkili rolü oynamaktadır.

Kullanıcı-Ürün matrisi üzerinde oluşturulmuş kullanıcı vektörleri sayesinde kullanıcılar arasındaki, ürün vektörleri sayesinde de ürünler arasındaki benzerlik hesaplamaları yapılmaktadır (Adomavicius ve Tuzhilin, 2005).

Öneri sisteminin türüne göre kullanıcılar veya ürünler arasındaki benzerlikler sim(a,b) hesaplanırken literatürde birçok yöntem kullanılmıştır (Breese ve ark., 1998; Sarwar ve ark., 2001). Bu tez kapsamında Kosinüs Benzerliği, Pearson Korelasyonu ve Ortalama Kare Farkı (OKF) benzerlik hesaplama yöntemleri sırasıyla açıklanmaktadır.

3.1.1.1.1. Kosinüs benzerliği

Kosinüs benzerliğinde iki öğenin benzerliği hesaplanırken ilk olarak her bir öğe kendisini oluşturan öznitelik (features) türünde bir vektör olarak tanımlanır. Bu çok boyutlu düzlemde iki vektör arasındaki açının kosinüs değeri hesaplanır (Salton ve McGill, 1983).

Burada kullanıcı tarafından verilen oylamalar çok boyutlu uzayda (film sayısı kadar boyut) bir vektör olarak düşünülmektedir. Bu vektör kullanıcının öznitelik kümesi olarak da ifade edilebilir. Daha sonra oluşturulmuş olan bu iki kullanıcı vektörü arasındaki kosinüs değeri bulunmaktadır. Bulunan değer iki öğe arasındaki kosinüs benzerlik oranıdır.

a ve *c* kullanıcıları arasındaki kosinüs benzerliği Denklem (3.1) ile ifade edilebilir:

$$\text{sim}(a, c) = \cos(\vec{a}, \vec{c}) = \frac{\vec{a}.\vec{c}}{||\vec{a}||.||\vec{c}||} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2}\sqrt{\sum_i^n b_i^2}} \quad (3.1)$$

Denklem (3.1)'de $\vec{a}$ vektörü *a* kullanıcısının değerlendirdiği filmlere verdiği oylamalardan oluşan vektör; $\vec{c}$ vektörü c kullanıcısının değerlendirdiği filmlere verdiği oylamalardan oluşan vektördür. $|\vec{a}|$, *a* vektörünün uzunluğunu, $|\vec{c}|$, i vektörünün uzunluğunu ifade etmektedir. Bir vektörün uzunluğu sahip olduğu tüm elemanların (a_i, b_i) karelerinin toplamının karekökü alınarak hesaplanmaktadır.

Kosinüs benzerliğinde iki vektör arasındaki açı ne kadar küçük ise bu iki vektörün arasındaki kosinüs değeri 1'e o kadar yakın olur. 1 değeri iki vektörün aynı olduğunu ve aradaki açının 0 olduğunu gösterir.

Hem içerik tabanlı hem de işbirlikçi filtreleme yaklaşımları benzerlik hesaplamasında aynı kosinüs hesaplamasını kullanmaktadırlar. İçerik tabanlı filtrelemede, filmleri tanımlayan öznitelik vektörleri arasında bir benzerlik hesaplaması yapılırken; işbirlikçi filtrelemede, kullanıcı tarafından yapılan oylamalardan oluşan vektörler arasında bir benzerlik hesaplaması yapılmaktadır. (Adomavicius ve Tuzhilin, 2005).

3.1.1.1.2. Pearson korelasyonu

Pearson korelasyon katsayısı (Pearson Correlation Coefficient) merkezlenmiş kosinüs benzerliği olarak da adlandırılabilir. Öneri sistemlerinde kullanıcılar veya ürünler arasındaki benzerlik kosinüs benzerliğine göre hesaplanırken ürünlerin toplam aldıkları puanların aritmetik ortalaması veya kullanıcının yapmış olduğu değerlendirmelerin aritmetik ortalaması dikkate alınmaz.

Pearson korelasyon katsayısı ile kullanıcıların veya ürünlerin benzerlikleri hesaplanırken var olan değerlendirmelerden kullanıcının veya ürünün ortalamasını çıkarmak suretiyle hesaplanmaktadır. Kullanıcılar veya ürünler için nasıl uygulanacağı şu şekilde açıklanabilir:

Kullanıcı Benzerliği: Her kullanıcı ürünlere aynı bakış açısıyla bakmadığı için yapmış oldukları değerlendirmeler farklı olmaktadır. Bazı kullanıcılar tüm derecelendirmeleri (0-5) kullanarak daha özenli değerlendirme yaparken bazı kullanıcılar ise beğendiği her ürün için en yüksek değerlendirmeyi yapabilmektedirler. Kimi kullanıcılar ise sürekli düşük değerlendirme yapma eğilimindedirler. Pearson korelasyonu benzerliği bu tür olumsuzlukları en aza indirebilmek için kullanıcıların kendi değerlendirme ortalamalarını alıp bunları hesaba katarak daha doğru sonuçlar elde etmeyi amaçlamaktadır (Ding ve ark., 2006).

$$\mathrm{sim}(a,b) = \frac{\sum_{i\in I}(R_{a,i}-R_a).(R_{b,i}-R_b)}{\sqrt{\sum_{i\in I}(R_{a,i}-R_a)2}.\sqrt{\sum_{i\in I}(R_{b,i}-R_b)2}} \qquad (3.2)$$

Denklem (3.2)'de $R_{a,i}$, a kullanıcısının i ürününe yaptığı değerlendirmeyi; R_a, a kullanıcısının yaptığı bütün değerlendirmelerin aritmetik ortalamasını; $R_{b,i}$, b kullanıcısının i ürününe yaptığı değerlendirmeyi; R_b ise b kullanıcısının yaptığı bütün değerlendirmelerin aritmetik ortalamasını ifade etmektedir. I kümesi hem a hem de b kullanıcılarının ikisinin de ortak değerlendirdiği ürünlerin kümesidir.

Ürün Benzerliği: Pearson korelasyon katsayısı yerine kosinüs benzerliği tercih edilmiş bir benzerlik bulma yönteminde çoğunluk tarafından düşük oylamayla değerlendirilmiş bir film kullanıcının komşuları tarafından yüksek puanlı olarak değerlendirilmiş ise o film yüksek ihtimal ile ilk sıralarda önerilecektir. Ama genele bakıldığında o filmin aldığı değerlendirmelerin ortalaması düşük olabilir veya kalitesiz bir film olabilir. Bu tarz olumsuzlukların önüne geçebilmek için pearson korelasyon katsayısı kullanılmaktadır.

$$\mathrm{sim}(i,j) = \frac{\sum_{i\in I}(R_{a,i}-R_i).(R_{a,j}-R_j)}{\sqrt{\sum_{a\in A}(R_{a,i}-R_i)2}.\sqrt{\sum_{a\in A}(R_{a,j}-R_j)2}} \qquad (3.3)$$

Denklem (3.3)'te $R_{a,i}$, i ürününe a kullanıcısı tarafından yapılan değerlendirmeyi; R_i, i ürününe kullanıcılar tarafından yapılan toplam değerlendirmelerin aritmetik ortalamasını; $R_{a,j}$, j ürününe a kullanıcısı tarafından yapılan değerlendirmeyi; R_j, j

ürününe kullanıcılar tarafından yapılan toplam değerlendirmelerin aritmetik ortalamasını ifade etmektedir. *A* hem *i* hem de *j* ürünlerini değerlendiren kullanıcıların kümesini ifade etmektedir.

Pearson benzerliğinde benzerlik -1 ile 1 arasında çıkmaktadır. 1'e ne kadar yakın ise o kadar benzer -1'e ne kadar yakın ise aradaki benzerlik o kadar az demektir.

3.1.1.1.3. Ortalama kare farkı benzerliği

Ortalama kare farkı benzerliği hesaplanırken öncelikle kullanıcılar veya ürünler arasındaki ortalama kare farkı (Mean Squared Difference) bulunmaktadır. Bulunan ortalama kare farkının tersi alınarak benzerlikler bulunmaktadır.

u ve *v* kullanıcıları arasındaki ortalama kare farkı Denklem (3.4) ile ifade edilebilir:

$$\mathrm{okf}(u,v) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2 \qquad (3.4)$$

Denklem (3.4)'te r_{ui}, *u* kullanıcısının i ürününe yaptığı değerlendirmeyi; r_{vi}, *v* kullanıcısının i ürününe yaptığı değerlendirmeyi göstermektedir. Bu denklem ile elde edilen okf(*u,v*), *u* ve *v* kullanıcıları arasında ortalama kare farkıdır.

Eğer oluşturduğumuz işbirlikçi filtreleme ürün tabanlı bir işbirlikçi filtreleme ise ürünler arasında bulunmasını istediğimiz ortalama kare farkı Denklem (3.5) ile ifade edilmektedir:

$$\mathrm{okf}(i,j) = \frac{1}{|U_{ij}|} \cdot \sum_{u \in U_{ij}} (r_{ui} - r_{uj})^2 \qquad (3.5)$$

Denklem (3.5)'te r_{ui}, *i* ürününe *u* kullanıcısı tarafından yapılan değerlendirmeyi ve r_{uj}, *j* ürününe *u* kullanıcısı tarafından yapılan değerlendirmeyi göstermektedir. $U_{i,j}$,

i ve *j* ürünlerini değerlendirmiş olan kullanıcıların kesişim kümesidir. Bu denklem ile *i* ve *j* ürünleri arasındaki ortalama kare farkı bulunmuş olur.

Aralarındaki ortalama kare farkı bulunmuş kullanıcı veya ürünler için ortalama kare farkı benzerliği Denklem (3.6) ve Denklem (3.7) ile hesaplanmaktadır (Surprise, 2015);

Kullanıcı benzerliği; $\text{okf_sim}(u, v) = \frac{1}{\text{okf}(u,v)+1}$ (3.6)

Ürün benzerliği; $\text{okf_sim}(i, j) = \frac{1}{\text{okf}(i,j)+1}$ (3.7)

Denklem (3.6) ile *u* ve *v* kullanıcıları arasındaki, Denklem (3.7) ile de *i* ve *j* ürünleri arasındaki ortalama kare farkı benzerliği hesaplanmaktadır.

İşbirlikçi filtreleme de kullanılmakta olan bu benzerlik bulma yöntemleri (kosinüs benzerliği, pearson korelasyonu, ortalama kare farkı) için yapılan matematiksel işlemler kullanıcı-ürün matrisi üzerinde yapılmaktadır. Eğer kullanıcı-ürün matrisi kısa zamanda büyük oranda değişmiyorsa benzerlikler bir defa yapılır ve hızlı bir şekilde sisteme giren kullanıcılara önerilerde bulunulabilir (Adomavicius ve Tuzhilin, 2005). Netflix gibi güçlü sunuculara sahip firmalar aktif kullanıcı ile diğer kullanıcılar arasındaki benzerlikleri gerçek zamanlı olarak hesaplayabilmektedirler.

İşbirlikçi filtrelemede kullanılan bu benzerlik bulma yöntemleri içerik tabanlı filtrelemede de kullanılabilir. İki filtrelemede aynı benzerlik hesaplama yöntemleri kullanılabilmesine karşın benzerlikleri bulunmak istenen vektörler (kullanıcı veya ürün vektörü) farklı değişkenlerden oluşmaktadır. İşbirlikçi filtrelemede aradaki benzerliği hesaplanan vektörler yapılan değerlendirmelerden oluşurken içerik tabanlı filtrelemede bu vektörler kullanıcı (cinsiyet, meslek, doğum yeri vb.) veya ürünlerin (renk, kalite, imalat yılı, yazar, yönetmen, oyuncu vb.) özniteliklerinden oluşmaktadır.

Literatürde benzerlik bulmak için çok yöntem kullanılmış ve karşılaştırılmıştır. Daha önce yapılmış deneysel çalışmalar kullanıcı tabanlı öneri sistemlerinde pearson

korelasyon katsayısı yönteminin diğer yöntemlerden daha başarılı olduğunu göstermektedir (Herlocker ve ark., 1999). Bu tez kapsamında yapılan deneylerde kullanıcı tabanlı ve ürün tabanlı öneri sistemlerinde Ortalama kare farkı benzerlik bulma yönteminin diğer yöntemlerden daha iyi sonuçlar verdiği ampirik olarak kanıtlanmıştır.

3.1.1.2. Komşuluk seçimi

Öneri sistemlerinde bir öğeye (kullanıcı veya ürün) en yakın komşular bir önceki aşamada bulunan benzerlik oranlarına göre bulunmaktadır. Bulunan bu komşular sayesinde kullanıcıya yeni önerilerde bulunmak için tahmin yapma aşamasına geçilmektedir.

İşbirlikçi filtrelemede kullanıcılar veya ürünler arasındaki benzerliklere göre işlem yapılmaktadır. Kullanıcı tabanlı işbirlikçi filtrelemede benzer kullanıcılar benzer ürünlere ilgi duyar mantığı kullanılmaktadır. Kullanıcı tabanlı işbirlikçi filtrelemede aktif kullanıcıya en yakın kullanıcılar bulunduktan sonra bu kullanıcıların en çok beğendiği ürünler aktif kullanıcıya önerilir. Burada önemli olan seçilen komşulardır. Çünkü aktif kullanıcıya benzerliği bulunan tüm kullanıcılar hesaplamaya dahil edilmemektedir. Benzerlik oranlarına göre belli sayıda veya oranda kullanıcı seçilmektedir. Bu işleme komşuluk seçimi denilmektedir.

Ürün tabanlı işbirlikçi filtrelemede ürünlerin birbirleri ile olan benzerlikleri hesaplanmaktadır. Daha sonra ürün benzerliklerine göre kullanıcıya ürünler önerilmektedir. Burada seçilen ürüne en yakın komşular yine benzerlik oranlarına göre seçilmekte olup tüm benzer ürünler hesaplamaya dahil edilmemektedir.

Komşuluk seçiminde kullanılan iki yöntem vardır (Gong, 2010). Birinci yöntem komşu seçerken öğeye olan benzerlik oranı belli bir eşik değerin üstünde olanların komşu seçilmesi, ikinci yöntem ise öğeye en yakın k adet komşu seçilmesidir (Gong, 2010).

Komşuluk seçiminde kullanılan bu yöntemler literatürde araştırılmış ve bazı problemler tespit edilmiştir (Anand ve Mobaher, 2005, Herlocker, 1999). Birinci yöntemde eşik değerinin yüksek seçilmesi bazı öğeler için komşu bulunamamasına veya çok az sayıda komşu bulunmasına neden olabilmektedir. Bu probleme Filtre Balonu (Filter Bubbles) denilmektedir . Filtre Balonu problemi kullanıcıya önerilecek ürünlerin belli bir küme içerisinde kısıtlı kalması olarak tanımlanmaktadır (Dooms, 2013). Kısıtlı kalmış bir öneri havuzunda sürekli aynı ürünler önerileceği için sistemin cazibesini kaybetmesine aynı zamanda da müşterilerini kaybetmesine neden olabilmektedir. Eşik değerin çok düşük tutulması öğeye benzerliği az olan öğelerin de tahmin yapma aşamasında etkili olmasına neden olmaktadır. Bir komşuluk seçim yöntemi olan *k*-EYK algoritmasında da benzer sorunlar yaşanabilmektedir. *k* komşu sayısının düşük bir değer seçilmesi aktif kullanıcıya yüksek benzerliği bulunan kullanıcıların hesaplamaya dahil edilmemesine neden olabileceği gibi *k* komşu sayısının yüksek bir değer seçilmesi de kullanıcıya benzerliği az olan kullanıcıların hesaplamaya dahil edilmesine neden olabilmektedir. Bu durum yapılan önerilen kalitesinin düşmesine neden olabilmektedir. İkinci yöntemde *k* değerinin düşük seçilmesi Filtre Balonu (Filter Bubbles) problemine neden olmamaktadır (Jannach ve ark., 2011).

3.1.1.3. Tahmin hesaplama

İşbirlikçi filtrelemede öğenin komşuları seçildikten sonra son aşamada artık aktif kullanıcıya ürün önerme yapılmaktadır. Bu aşamada aktif kullanıcının henüz değerlendirmediği ürünler için yapabileceği değerlendirmeler tahmin edilmeye çalışılır. Ürün değerlendirme tahmini yapılırken bir önceki aşamada seçilen komşuların yapmış oldukları değerlendirmeler doğrultusunda hesaplama yapılmaktadır. Bu hesaplamalar yapılırken komşuların ağırlıkları benzerlik oranlarına göre seçilip işleme koyulmaktadır. Önerilecek ürün için seçilen komşuların değerlendirmeleri bir takım tahmin yapma hesaplamalarına tabi tutularak tahmini değerlendirme hesaplanmaktadır (Herlocker ve ark., 1999).

Literatürde birçok tahmin hesaplama yöntemi bulunmakla beraber, en yaygın olarak kullanılan dört adet tahmin hesaplama yöntemi Basit Ortalama (BO), Ağırlıklandırılmış Ortalama (AO), Ayarlanabilir Ağırlıklandırılmış Ortalama (AAO) ve Standart Sapmalı Ayarlanabilir Ağırlıklandırılmış Ortalama (SAAO) yöntemleridir (Milli, 2013).

3.1.1.3.1. Basit ortalama

Basit ortalama (Simple Average) yönteminde bir önceki aşamada seçilen komşuların yapmış oldukları değerlendirmelerinin aritmetik ortalaması alınarak aktif kullanıcının yapacağı değerlendirme tahmin edilmektedir. Ürün önerisi yapılacak bir sistemde yapılan hesaplamalar sonucunda en yüksek değerlendirme puanlarına sahip ürünler aktif kullanıcıya önerilmektedir.

Basit ortalama yöntemi yoğun matematiksel işlemler gerektirmediği için hızlı çalışmaktadır. Bu yüzden zaman sorunu olan problemlerde kullanılabilir. Zamandan kazanç sağlamasına karşın hata oranı diğer hesaplamalara göre genelde daha yüksek çıkmaktadır. Basit ortalama yönteminde tahmin hesabı Denklem (3.8) ile ifade edilebilir:

$$R_{a,i} = \frac{\sum_{u \in U}^{n} R_{u,i}}{n} \tag{3.8}$$

Denklem (3.8)'de U kümesi a kullanıcısının komşuluk aşamasında seçilmiş komşularının kümesi ve n bu kümenin eleman sayısıdır. $R_{u,i}$, u kullanıcısının i ürünü için yapmış olduğu değerlendirmeyi ifade etmektedir. Denklem (3.8)'den anlaşılacağı üzere a kullanıcısının i ürünü için yapabileceği tahmini değerlendirmeyi bulmak için sistem a kullanıcısının seçilmiş komşularının i ürünü için yapmış oldukları tüm değerlendirmelerinin aritmetik ortalamasını hesaplanmaktadır.

3.1.1.3.2. Ağırlıklandırılmış ortalama

İşbirlikçi filtreleme sistemlerinde kullanıcının yapacağı değerlendirmeyi tahmin etme aşamasında en çok kullanılan yöntem ağırlıklandırılmış ortalama (Weighted Average) yöntemidir (Adomavicius ve Tuzhilin, 2005).

Basit ortalamanın aksine birinci aşamada bulunan benzerlikler hesaplamaya dahil edilmektedirler. Bu benzerlikler tahmin hesaplamaları yapılırken komşuların yapmış oldukları değerlendirmeler ile çarpılarak komşuların tahmin hesaplamasındaki ağırlıkları hesaba katılmış olur. Benzerliği en yüksek olan kullanıcının tahmin yapma aşamasında daha etkili olması yapılan tahminin doğruluk oranının daha yüksek olmasını sağlamaktadır.

$$R_{a,i} = \frac{\sum_{u \in U}^{n} R_{u,i}.sim(a,u)}{\sum_{u \in U}^{n} \text{sim(a,u)}} \quad (3.9)$$

Denklem (3.9)'da *U* kümesi aktif kullanıcının *i* ürününü değerlendirmiş komşularının kümesi ve *n* bu kümenin eleman sayısıdır. $R_{u,i}$, *u* kullanıcısının *i* ürünü için yapmış olduğu değerlendirmedir. *sim(a,u)*, *a* ile *u* kullanıcısının benzerlik oranıdır. Yapılan işlem sonucu olan $R_{a,i}$, *a* kullanıcısının daha önceden değerlendirmediği *i* ürünü için yapacağı tahmini değerlendirmedir.

3.1.1.3.3. Ayarlanabilir ağırlıklandırılmış ortalama

Pearson korelasyonu hesaplanırken hesaba katılan kullanıcıların yapmış oldukları tüm değerlendirmelerin aritmetik ortalaması ayarlanabilir ağırlıklandırılmış ortalamada da (Adjusted Weighted Average) hesaba katılarak daha doğru sonuçlar elde edilmeye çalışılmaktadır (Adomavicius ve Tuzhilin, 2005; Ding ve ark., 2006). Ağırlıklandırılmış ortalamada yapılan hesaplama işleminden farklı olarak komşuların yapmış oldukları değerlendirmelerden kendi ortalamaları çıkarılmaktadır. Daha sonra yine ağırlıklandırılmış ortalamada olduğu gibi benzerlik oranıyla çarpılıp benzerlik oranlarının toplamına bölünmektedir. Hesaplama aşamasında çıkarılmış olan komşuların ortalamalarının yerine tahmin yapılmak istenen aktif kullanıcının

ortalaması hesaba dahil edilmektedir. Ayarlanabilir ağırlıklandırılmış ortalama ile tahmin hesaplama işlemi Denklem (3.10) ile ifade edilebilir:

$$R_{a,i} = R_a + \frac{\sum_{u \in U}^{n} (R_{u,i} - R_u).sim(a,u)}{\sum_{u \in U}^{n} \text{sim(a,u)}} \quad (3.10)$$

Denklem (3.10)'da *U* kümesi aktif kullanıcının *i* ürününü değerlendirmiş komşularının kümesi ve *n* bu kümenin eleman sayısıdır. $R_{u,i}$, *u* kullanıcısının i ürünü için yapmış olduğu değerlendirmedir. R_a, *a* kullanıcısının, R_u ise *u* kullanıcısının yapmış olduğu tüm değerlendirmelerin aritmetik ortalamasıdır. *sim(a,u), a* ile *u* kullanıcılarının benzerlik katsayısıdır. Yapılan işlem sonucu olan $R_{a,i}$, *a* kullanıcısının daha önceden değerlendirmediği i ürünü için yapacağı tahmini değerlendirmedir.

3.1.1.3.4. Standart sapmalı ayarlanabilir ağırlıklandırılmış ortalama

Standart sapma değeri, yapılan değerlendirmelerin birbirinden ne kadar uzak olduklarının göstergesidir. Eğer standart sapma değeri küçük ise yapılan değerlendirmeler aritmetik ortalamaya yakındır. Aksi durumda aritmetik ortalamadan uzak noktalardadır.

$$\text{Standart Sapma } (\sigma a) = \frac{\sum (R_{a,i} - R_a)^2}{\text{N}} \quad (3.11)$$

Denklem (3.11)'de σa, *a* kullanıcısının standart sapmasını göstermektedir. $R_{a,i}$, *a* kullanıcısının *i* ürününe yaptığı değerlendirmeyi ve R_a, *a* kullanıcısının yapmış olduğu tüm değerlendirmelerin aritmetik ortalamasını ifade etmektedir. *N* ise *a* kullanıcısının toplam değerlendirdiği ürünlerin sayısıdır.

Denklem (3.11)'den anlaşılacağı üzere AAO tahmin hesaplama yönteminde eğer aktif kullanıcının standart sapması yüksek ise tahmin edilen değerlendirme gerçek değerden yüksek çıkabilmekte, aksi durumda da düşük çıkabilmektedir. Bunun önüne geçmek için aktif kullanıcının standart sapması hesaplamaya dahil edilerek SAAO

yöntemi ile yapılan tahminde doğruluk oranı iyileştirilmeye çalışılmaktadır (Herlocker ve ark., 1999).

$$R_{a,i} = R_a + \frac{\sum_{u\in U}^{n}(R_{u,i}-R_u).sim(a,u).\sigma a}{\sum_{u\in U}^{n}\mathrm{sim(a,u)}} \quad (3.12)$$

Denklem (3.12)'den anlaşılacağı üzere ayarlanabilir ağırlıklandırılmış ortalama yöntemine aktif kullanıcının standart sapması dahil edilerek standart sapmalı ayarlanabilir ağırlıklandırılmış ortalama (Standart Deviation Adjusted Weighted Average) denklemi elde edilmiştir.

3.1.1.4. İşbirlikçi filtreleme algoritması örneği

İşbirlikçi filtreleme sisteminin tüm aşamaları önceki bölümlerde anlatılmıştır. Bu bölümde basit bir örnek üzerinde bu aşamalar gösterilmektedir. İşbirlikçi filtrelemede ilk yapılan işlem kullanıcı ve ürünlerin bir matris haline getirilmesidir. Satırlar kullanıcıları, sütunlar ise ürünleri temsil etmektedir. Matrisin her hücresinde [kullanıcı, ürün] kullanıcının ilgili ürün için yaptığı değerlendirme bulunmaktadır. Çizelge 3.1.'de 4 kullanıcı ve 4 ürün bulunan bir veri seti için değerlendirme matrisi oluşturulmuştur. Matriste bulunan boş hücreler oylama yapılmadığını göstermektedir. Gerçek verilerde genelde boş hücreler daha çoktur. Çünkü hiçbir sistem kullanıcılardan yeterli geri bildirimi alamamaktadır. Verilen örneğin gerçeğe daha çok benzemesi amacıyla bazı hücreler boş bırakılmıştır.

Çizelge 3.1. Kullanıcı-ürün matrisi

Kullanıcı / Ürün	A ürünü	B ürünü	C ürünü	D ürünü
d1 kullanıcısı	1		5	4
d2 kullanıcısı	2	3	4	
d3 kullanıcısı	4	5		2
d4 kullanıcısı			4	5

Kullanıcı tabanlı bir işbirlikçi filtreleme algoritmasında ilk adım kullanıcıları tanımlayan vektörlerin oluşturulmasıdır. Değerlendirilmemiş ürünler için birçok hesaplama yapılabilir. Bazı algoritmalarda kullanıcının yaptığı değerlendirmelerin ortalaması alınırken bazılarında ile ürün için yapılmış değerlendirmelerin ortalaması alınmaktadır. Verilen örnekte kullanıcı-ürün matrisinde bulunan boş hücreler için 0 "Sıfır" değeri kabul edilerek hesaplama yapılmaktadır.

İlk aşama olan benzerlik bulma aşaması için oluşturulan kullanıcı vektörleri şu şekilde tanımlanabilir:

- d1 Kullanıcısı için oluşturulan vektör: $\overrightarrow{d1} = < 1,0,5,4 >$
- d2 Kullanıcısı için oluşturulan vektör: $\overrightarrow{d2} = < 2,3,4,0 >$
- d3 Kullanıcısı için oluşturulan vektör: $\overrightarrow{d3} = < 4,5,0,2 >$
- d4 Kullanıcısı için oluşturulan vektör: $\overrightarrow{d4} = < 0,0,4,5 >$

Bu vektörler arasındaki benzerlikler daha önceki bölümlerde anlatılan Kosinüs Benzerliği, Pearson Korelasyonu ve Ortalama Kare Farkı benzerlik bulma yöntemleri ile hesaplanabilir. Verilen bu örnekte Çizelge 3.1.'de görülen matriste yer alan tüm kullanıcılar için oluşturulmuş vektörler arasındaki benzerlikler Denklem (3.1)'te gösterilen kosinüs benzerliğine göre hesaplanmıştır. Bulunan benzerlik oranları şu şekildedir:

- sim (d1,d2)= 0,63
- sim (d1,d3)= 0,28
- sim (d1,d4)= 0,96
- sim (d2,d3)= 0,64
- sim (d2,d4)= 0,46
- sim (d3,d4)= 0,22

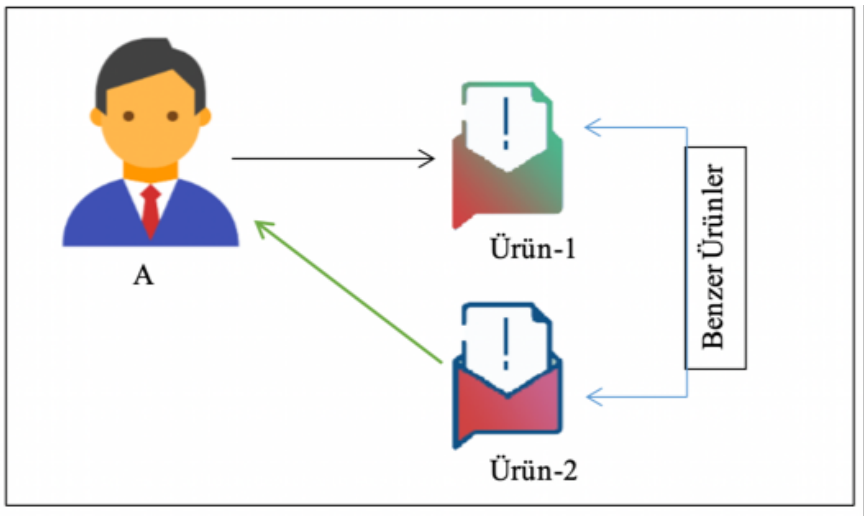
Yukarıda kosinüs benzerliğine göre yapılmış hesaplama sonucunda bulunan benzerlikler görülmektedir. İkinci aşama olan komşuluk seçiminde hesaplanan benzerlik oranlarına göre komşular belirlenir. Komşu belirlenme aşamasında Bayes,

DVM veya *k*-EYK algoritmaları gibi sınıflandırma algoritmaları kullanılmaktadır. Verilen örnekte *K* en yakın komşu algoritmasına göre komşuluklar bulunmaktadır. *K* değerini "1" aldığımız takdirde her kullanıcıya en çok benzeyen bir başka kullanıcı seçilmektedir. d1 kullanıcısı için yapılacak komşuluk seçiminde, d1 kullanıcısına en çok benzeyen d2 kullanıcısı komşu olarak alınmaktadır.

Komşular bulunduktan sonra yapılması gereken işlem üçüncü aşama olan tahmin hesaplama aşamasıdır. Bu aşamada seçilen tahmin hesaplama yöntemine göre bir tahmin yapılabilir. Verilen örnek için yapılan hesaplamalarda tek komşu olduğu için sadece en yakın komşunun beğenileri önerilmektedir. Böyle bir durumda d2 kullanıcısı tarafından 3 puan ile değerlendirilmiş B ürününü d1 kullanıcısının da 3 puan vererek değerlendirmesi tahmin edilmektedir.

3.1.2. İçerik tabanlı filtreleme

Aktif kullanıcının değerlendirdiği ürünlerin (film, müzik, eşya vb.) öznitelikleri, bir kullanıcı profili oluşturmak için kullanılmaktadır. Tüm ürünler bu profille karşılaştırılır ve en benzerleri aktif kullanıcıya önerilmektedir (Lekakos ve Caravelas, 2008).



Şekil 3.2. İçerik tabanlı filtreleme

Şekil 3.2.'de görülen A kullanıcısı tarafından satın alınmış Ürün-1'in öznitelikleri sayesinde A kullanıcısının profili oluşturulmaktadır. Daha sonra Ürün-1 ile benzerliği bulunan Ürün-2 kullanıcının profiliyle uyuşacağı için A kullanıcısına önerilebilmektedir.

İçerik tabanlı filtreleme, öğeyi özelliklere göre tanımlamanın mümkün olduğu alanlarda uygulanabilmektedir. Buna karşın işbirlikçi filtreleme, yapılan değerlendirmeler yoluyla ifade edilen öznel "genel lezzet" kriterine dayandığı ve özelliklere yönelik analiz gerektirmediği için her türlü içeriğe uygulanabilmektedir (Lekakos ve Caravelas, 2008).

İçerik tabanlı filtreleme yöntemi çoğu zaman tatmin edici sonuçlar verse de bu her zaman böyle olmamaktadır. Örneğin haber öneri sistemlerinde kullanılan bir içerik tabanlı filtreleme çok verimli olmamaktadır. Çünkü kullanıcılara önerilecek haberler daha önce zaten okumuş oldukları haberlere benzeyeceği hatta aynı haberler olabileceği için kullanıcı bu durumdan sıkılabilir. Bu durum tüm haber türleri için geçerli değildir. Örneğin spor haberleri için bu filtreleme işe yarayabilir. Çünkü spor tutkunları sevdikleri bir futbol takımının kazandığı maçları farklı kişilerden farklı yorumlarla tekrar tekrar dinlemekten veya okumaktan hoşnut olabilmektedirler (Taşçı, 2015).

İşbirlikçi filtreleme kullanıcıların aktif hareketlerinden (değerlendirmelerinden) faydalanarak benzerlik hesaplamaları yaptığı için sürekli güncellenen bir benzerlik oranına sahiptir. Bu durum sistemin sık sık güncellenmesi gerektiği anlamına gelmektedir. İçerik tabanlı filtreleme buna nispeten daha az aralıklarla benzerlik hesaplamalarına ihtiyaç duymaktadır. Çünkü içerik tabanlı filtreleme sistemi benzerlik hesaplamaları için, ürünlerin daha az değişen özniteliklerinden faydalanmaktadır. Örneğin, bir kitabın öznitelikleri yazar, tür, dil vb. olurken bir film için öznitelikler tür, yıl, oyuncu, yönetmen vb. olabilmektedir. Bu öznitelikler düşünüldüğünde sürekli bir değişime uğramadıkları anlaşılmaktadır. Böyle bir durumda içerik tabanlı filtreleme ile yapılmış bir öneri sisteminin sık aralıklarla benzerlik hesaplamasına ihtiyaç duymayacağı anlaşılmaktadır.

İçerik tabanlı filtreleme sistemlerinde [ürün x öznitelik] matrisi kullanılmaktadır (Lu ve ark., 2015). Çizelge 3.2.'de içerik tabanlı filtreleme kullanılan bir film öneri sistemi için oluşturulmuş [film x öznitelik] matrisinden bir kesit görülmektedir. Matris oluşturulurken MovieLens veri setindeki gerçek bilgiler kullanılmıştır.

Çizelge 3.2: Ürün-özellik matrisi

Film / Öznitelik	Tür	Aktör	Yıl	Yönetmen
Sudden Death	Aksiyon	Jean-Claude Van Damme	1995	Peter Hyams
Nixon	Drama	Anthony Hopkins	1995	Oliver Stone
Four Rooms	Komedi	Quentin Tarantino	1995	Quentin Tarantino
Boomerang	Komedi/Romantik	Eddie Murphy	1992	Reginald Hudlin

İçerik tabanlı bir film öneri sistemi [film x öznitelik] matrisinin satırları arasındaki benzerlikleri hesaplamaktadır. Daha sonra bir kullanıcının geçmişte beğendiği filmlere benzerliği en yüksek olanlar bulunup kullanıcıya önerilmektedir.

Ürünler (film, müzik, makale vb.) arasındaki benzerlikler hesaplanırken metin analizi, anahtar kelime benzerliği veya ürünün farklı öznitelikleri göz önüne alınmaktadır. Bu öznitelikler bir müzik parçası için şarkıcı, albüm adı, yılı vb. olurken bir araba için kilometre, yakıt türü, rengi vb. olabilmektedir.

İçerik tabanlı öneri sistemleri çoğunlukla metin tabanlı ürünler için tasarlanmıştır. Bu tür öneri sistemlerinde içerik çoğunlukla anahtar kelimelerle tanımlanmaktadır. Bir doküman için tasarlanan içerik tabanlı filtreleme algoritması dokümanı tanımlarken yazarı, yılı gibi özniteliklerin yanında metin analizi yaparak kelime frekansları veya anahtar kelime yoğunluğunu da kullanabilmektedir.

Metin tabanlı öneri sistemlerinde anahtar kelime ağırlıklarını çoğunlukla, terim frekansı / ters doküman frekansı (TF-TDF) ölçüsü ile hesaplanmaktadır. (Adomavicius ve Tuzhilin, 2005). Bununla birlikte Bilgi Kazancı (BK) metriği de sıklıkla kullanılmaktadır (Taşçı, 2015). Örneğin bir haber öneri sisteminde haberleri

tanımlamak için haberin içeriğinde bulunan her bir kelimenin TF-TDF ağırlığı bulunarak aralarındaki benzerlikler hesaplanmaktadır.

j belgesi için oluşturulmuş içerik vektörü (d_j) *i* belgesindeki kelimelerin ağırlıklarından oluşmaktadır.

$$içerik(d_j) = w_{i,j}, \dots, w_{k,j} \quad (3.13)$$

Denklem (3.13)'te $w_{i,j}$ *i* kelimesinin *j* belgesindeki ağırlığını ifade etmektedir. *j* belgesindeki *i* anahtar sözcüğü için TF-TDF ağırlığı Denklem (3.14)'te gösterildiği gibi ifade edilebilir:

$$w_{i,j} = TF_{i,j} \text{ x } IDF_i \quad (3.14)$$

İçerik tabanlı filtreleme kullanıcının tercihlerine dayandığından dolayı işbirlikçi filtreleme kadar karmaşık matematiksel hesaplamalara ihtiyaç duymamaktadır (Taşçı, 2015). İşbirlikçi filtrelemede yaşanan sahte profil saldırılarından etkilenmemektedir. İşbirlikçi filtrelemede bulunan yeni ürün sorunu içerik tabanlı filtrelemede yaşanmamaktadır. Çünkü bir ürün sisteme yeni eklenmiş olsa bile sahip olduğu öznitelikler sayesinde başka ürünlerle birtakım benzerliklere sahiptir. Bu benzerlikler sayesinde önerilebilmektedir.

İçerik tabanlı filtrelemenin de sahip olduğu bazı dezavantajlar vardır. Genel olarak içerik tabanlı filtreleme kullanmanın sahip olduğu dezavantajlar şu şekilde sıralanabilir:

- **Aşırı özelleşme sorunu:** Kullanıcılara sadece kendi tercih ettikleri ürünlerin sahip oldukları özniteliklere göre önerilerde bulunulduğundan bir süre sonra Filtre Bolunu problemi ortaya çıkabilmektedir. Bu problem kullanıcıya yeni şaşırtıcı önerilerde bulunmamasına neden olabilmektedir. Örneğin bir kişiye sürekli korku filmi önermek gibi. Sisteme rastgelelik katılarak çözümler üretilebilir (Adomavicius ve Tuzhilin, 2005).

- **Yeni kullanıcı sorunu:** Sisteme ilk defa girmiş bir kullanıcı henüz herhangi bir ürün hakkında bir değerlendirmede bulunmadığı için bu kullanıcıya ürün önerilemeyecektir (Taşçı, 2015). Bu yeni kullanıcı problemi hem işbirlikçi hem de içerik tabanlı filtrelemede bulunmaktadır.
- **Sınırlı içerik analizi:** Bir veri kümesi sadece belirtilmiş öznitelikler ile tanımlanabilmektedir. İki farklı öğe aynı özellikler ile tanımlanmış ise sistem ikisini ayırt edemez (Adomavicius ve Tuzhilin, 2005). Yıl ve tür bilgilerinin baz alındığı bir film öneri sisteminde aynı yılda çıkmış ve aynı türe sahip filmlerin tanım vektörleri aynı olmaktadır. Sistem öneri yaparken hangisini öne çıkarması gerektiğini ayırt edemeyecektir.
- **Nesnel olması:** Elementlerin nesnel bilgisine dayanır. Örneğin bir restoran öneri sistemi için, bir restoranın atmosferi, temizliği, personel davranışları dikkate alınmadan sadece restoran konum, bina, çalışan sayısı, yemek türleri vb. öznitelikleri üzerinden önerildiği için sorunlar yaşanabilmektedir (Chrıstakou ve Vrettos, 2007).

Bir film öneri uygulaması için içerik tabanlı filtreleme düşünüldüğünde, bir filmi aktif kullanıcıya önermek için içerik tabanlı öneri sistemi, aktif kullanıcının geçmişte değerlendirdiği filmler arasındaki ortak noktaları tespit eder (belirli aktörler, yönetmenler, anahtar kelimeler, konular, vb.). Daha sonra, aktif kullanıcının tercihleri doğrultusunda oluşturulan profille en çok benzerliği gösteren filmler önerilir (Adomavicius ve Tuzhilin, 2005).

3.1.3. Hibrit filtreleme

İşbirlikçi ve içerik tabanlı filtreleme, kullanıcıların ilgisini çekebilecek veya ihtiyaçları giderebilecek yeni ürünleri öngören öneri sistemlerinin ana yöntemleridir. Her yöntemin kendine özgü avantajları ve dezavantajları bulunmaktadır. Her biri ayrı ayrı bazı problemlere en uygun çözümler önerebilmektedir (Lekakos ve Caravelas, 2008).

Hibrit Filtreleme (HF) birbirlerinin eksikliklerini ortadan kaldırabilmesi için içerik tabanlı ve işbirlikçi filtrelemenin değişik şekillerde bir araya getirilmesiyle ortaya çıkarılmış yöntemlerdir (Taşçı, 2015). Hibrit filtrelemenin temel amacı herhangi bir bileşeninin sahip olduğu bir dezavantajı diğer bileşen yöntem sayesinde çözmektir.

İşbirlikçi ve içerik tabanlı filtrelemenin bir araya getirilmesinde farklı yöntemler kullanılmaktadır. Bu yöntemler şu şekilde sıralanabilir (Adomavicius ve Tuzhilin, 2005):

- İşbirlikçi ve içerik tabanlı filtreleme yöntemlerini veri seti üzerinde ayrı ayrı uygulamak ve elde edilen tahminleri belli oranlarda bir araya getirmek.
- Bazı içerik tabanlı filtreleme özelliklerini işbirlikçi bir yaklaşıma dahil etmek. Böyle bir yol izlenmesi durumunda işbirlikçi filtrelemenin sahip olduğu yeni ürün problemi içerik tabanlı filtreleme sayesinde kısmen çözülmektedir.
- İçerik tabanlı bir yaklaşıma bazı işbirlikçi filtreleme özelliklerini dahil etmek. Böyle bir durumda içerik tabanlı filtrelemenin sahip olduğu Filtre Balonu sorunu işbirlikçi filtreleme sayesinde kısmen çözülmektedir.
- Hem içerik tabanlı hem de işbirlikçi filtreleme özelliklerini içeren genel bir sistem oluşturmak.

Hibrit öneri sistemleri, yapılan önerilerin doğruluğunu iyileştirmek veya geleneksel öneri sistemlerinin sınırlamalarına (örneğin, yeni ürün, yeni kullanıcı vb. problemler) kısmen çözüm olmak için olay temelli akıl yürütme gibi bilgi temelli teknikler kullanmaktadır (Adomavicius ve Tuzhilin, 2005). Öneri sistemleri gibi bilgi tabanlı sistemlerin temel problemi, büyük verilere ihtiyaç duyulmasıdır. Bu durum birçok yapay zekâ uygulamasında sıkça yaşanan bir sorundur. Bilgi temelli öneri sistemleri, çoğunlukla makine tarafından okunabilen verilere sahip uygulama alanlarında kullanılmaktadır (Adomavicius ve Tuzhilin, 2005).

Birçok çalışma, hibrit yöntemlerin geleneksel yöntemlerden daha doğru öneriler sağladıklarını göstermek için hibrit yaklaşımın performansını, işbirlikçi ve içerik tabanlı filtreleme yöntemlerinin performanslarıyla karşılaştırmıştır. Yapılan bu karşılaştırmalar hibrit yöntemlerin daha başarılı sonuçlar verdiğini göstermektedir (Balabanovic ve Shoham, 1997; Pazzani, 1999; Soboroff ve Nıcholas, 1999; Burke, 2002; Melville ve ark., 2002; Milli, 2013; Laveti ve ark., 2016).

3.2. Öneri Sistemlerindeki Önemli Faktörler

Ayrışma (Wikipedia, 2018) – Kullanıcılar bazen birbirine daha az benzeyen ürünleri alma eğiliminde olabilmektedirler. Örneğin sürekli aynı sanatçıdan müzikleri öneren bir sistem yerine bazen de farklı sanatçılardan müzikler önermek sistemi daha verimli hale getirebilir. Burada ki asıl amaç öneri sistemlerinin daha önce haberdar olunmayan ürünleri bile önerebiliyor olmasıdır.

Önerme sürekliliği (Wikipedia, 2018) – Bazı alanlarda her seferinde yeni ürünler önermektense bazen de aynı ürünleri önermek kişinin ilgi alanlarına daha çok hitap edebilmektedir. Önerme sürekliliği her alanda aynı öneme sahip değildir. Örneğin bu durum spor haberleri için oluşturulan bir öneri sisteminde etkili olacaktır. Çünkü okuyucu sevdiği takımın maçları için olan haberleri farklı yorumculardan tekrar tekrar dinlemekten keyif alabilmektedir (Taşçı, 2015). Ama araba parçaları satan bir çevrimiçi alışveriş mağazası için verimli olmaz. Dikiz aynası satın almış bir kullanıcıya tekrar tekrar dikiz aynası modelleri göstermenin bir mantığı olmayacaktır. Araba aksesuarları ilgili ama farklı kategorilerden ürünler (silecek, buz kıracağı, parlatıcı vb.) göstermek satışlara olumlu yönde etki edebileceği gibi kullanıcının sisteme daha fazla güvenmesini de sağlayacaktır.

Gizlilik (Wikipedia, 2018) – Öneri sistemleri kullanıcıların kişisel bilgilerini kullandığı için kullanıcıya bu bilgilerin güvende olduğu hissini vermelidir. Demografik filtreleme tabanlı öneri sistemlerinde kullanıcıların demografik özellikleri (yaş, cinsiyet, ırk vb.) tutulduğu için gizlilik çok önemli olabilmektedir.

Kullanıcı demografisi (Wikipedia, 2018) – Kullanıcıların kimlikleri öneri memnuniyetlerini etkileyebilmektedir. Örneğin Müslüman toplumlarda yılbaşında Noel ürünleri önermek fazla verimli olmayacağı gibi sisteme karşı bir antipati oluşmasına neden olabilir. Veya bir kadına erkek ürünleri önermek satışlara olumlu yönde etki etmeyecektir.

Düzgünlük (Wikipedia, 2018) – Kullanıcı katılımı olan öneri sistemleri sahtekârlıklara karşı bazı önlemlere sahip olmalıdır. Bazen ürün satıcısı kullanıcı girişi yaparak kendi ürünlerine yüksek puanlı değerlendirmeler yapabilmektedir. Böyle bir durumda sistemin manipüle olması kaçınılmazdır.

Rastlantısallık (Wikipedia, 2018) – Rastlantısallık "önerilerin ne kadar şaşırtıcı olduğu"dur. Ayrılma ile benzeşiyor olsa da rastlantısallık biraz daha farklıdır. Sürekli korku filmi izleyen bir kullanıcıya beğenebileceği bir dram filmi önerisi ile kullanıcı hem şaşırtılabilir hem de sisteme olan güveni arttırılabilir.

Güvenilirlik (Wikipedia, 2018) – Güven vermeyen bir öneri sisteminin kullanıcı nezdinde bir değeri olmaz. Kullanıcılara öneri yöntemlerinin nasıl çalıştığı açıklanarak güven kazanılabilir. Örneğin kullanıcıya işbirlikçi filtreleme sistemi anlatılarak aslında yapılan önerilerin gerçek insanların yaptığı değerlendirmeler kullanılarak yapıldığı açıklanabilir. Böyle bir durumda kullanıcı yapılan önerilerin gerçek insan davranışlarından elde edildiğini bilecek ve sisteme daha çok güvenecektir.

Seyreklik – Öneri sistemlerindeki başlıca sorunlardan bir tanesi olan veri seyrekliği sorunu öneri sistemleri için daha yeni yöntemler geliştirilmesini gerektirmektedir. Çünkü veri setlerinde olan veri seyrekliği yüzünden kullanıcıların profilleri tam oluşturulamamaktadır. Bu yüzden de kişiselleştirilmiş bir deneyim sunmak zor olmaktadır. Tez kapsamında kullanılan veri setinde seyreklik %98,36'dır. Kullanıcı-Film matrisinin çok küçük bir bölümü doludur. Sistemlerde geri bildirimler otomatikleştirilebileceği gibi kullanıcının hareketlerinden veri toplanabilir veya sistemler cazip hale getirilerek kullanıcı geri bildirim yapmaya yöneltilebilir.

Etiketleme (Wikipedia, 2018) – İçerik tabanlı filtreleme kullanılan bir öneri sisteminde önerilecek ürünü tanımlamak için etiketler (öznitelikler) kullanılır. Bu etiketlemenin yanlış yapılması, yapılacak önerilerin kalitesinin düşük olmasına neden olabilmektedir. Online bir alışveriş sitesinde hediyelik eşya ve ihtiyaç diye etiketlenmiş ürünler birbirinden tamamen farklıdır. Bebek bezi bir ihtiyaç iken çiçek buketleri birer hediyedir. Kişiye hediye eşya önerisi diye bebek bezi önermek kişinin sisteme olan güveninin azalmasına neden olabileceği gibi sisteme tekrar girmemesine de neden olabilir.

Öneri sistemleri sahip olduğu bu faktörler ile aktif olarak birçok araştırmaya konu olmaktadır.

3.3. Öneri Sistemleri Araştırmaları

Öneri sistemlerinde kullanılan öneri yaklaşımları önceki bölümlerde işbirlikçi filtreleme yaklaşımı, içerik tabanlı filtreleme yaklaşımı ve hibrit filtreleme yaklaşımı olarak üç ana başlık altında incelenmiştir. Adomavicius ve Tuzhilin (2005), yaptıkları çalışmada öneri sistemlerinde değerlendirme tahmini için kullanılan öneri tekniklerini sezgisel tabanlı öneri tekniği ve model tabanlı öneri tekniği olarak iki başlıkta toplayabileceğimizi söylemişlerdir.

Sezgisel tabanlı öneri tekniği: Çevrimiçi öğrenme teknikleri kullanılmaktadır. Bu teknikte kullanıcı profilleri güncel tutularak, gerçek zamanlı öneriler yapılmaktadır.

Model tabanlı öneri tekniği: Çevrimdışı öğrenme teknikleri kullanılmaktadır. Bu modelin uygulanması performans sorunlarına daha açıktır. Bu yöntemin dezavantajı ikili (binary) çalışma mantığında olmasıdır. Yani kullanıcı belli bir ürün için iki seçeneğe sahiptir. Beğenme ve beğenmeme durumu dışında bir değerlendirme kullanılmaz (Pereira ve Hruschka, 2015).

Öneri sistemleri, işbirlikçi filtreleme teriminin ilk defa David Goldberg ve ark. tarafından 1992 yılında yapılan bir makalede kullanılmasından beri çok araştırmaya konu olmuştur. Çizelge 3.3.'de görüleceği üzere 2005'li yıllara daha gelmeden bile bu konuda birçok çalışma yapıldığı görülmektedir (Adomavicius ve Tuzhilin, 2005).

Çizelge 3.3.'de yapılan çalışmalar öneri yaklaşımına göre içerik tabanlı ve işbirlikçi filtreleme ve bu yaklaşımlarda kullanılan öneri tekniklerine göre sezgisel tabanlı ve model tabanlı olarak kategorize edilmiştir.

Çizelge 3.3. Öneri sistemleri sınıflandırmalarının araştırmaları (Adomavicius ve Tuzhilin, 2005)

Öneri Yaklaşımı	Öneri Tekniği	
	Sezgisel Tabanlı	Model Tabanlı
İçerik Tabanlı Filtreleme	Sık kullanılan teknikler: - TF-TDF (Bilgi Alımı) - Kümeleme Tipik araştırma örnekleri: - Lang 1995 - Balabanovic ve Shoham 1997 - Pazzani ve Billsus 1997	Sık kullanılan teknikler: - Bayes sınıflandırma - Kümeleme - Karar ağaçları - Yapay sinir ağları Tipik araştırma örnekleri: - Pazzani ve Billsus 1997 - Mooney ve Roy 1999 - Billsus ve Pazzani 1999, 2000 - Zhang ve ark. 2002
İşbirlikçi Filtreleme	Sık kullanılan teknikler: - En yakın komşu (kosinüs, korelasyon) - Kümeleme - Grafik teorisi Tipik araştırma örnekleri: - Resnick ve ark. 1994 - Hill ve ark. 1995 - Shardanand ve Maes 1995 - Breese ve ark. 1998 - Nakamura ve Abe 1998 - Aggarwal ve ark. 1999 - Delgado ve Ishii 1999 - Pennock ve Horwitz 1999 - Sarwar ve ark. 200 I	Sık kullanılan teknikler: - Bayes ağları - Kümeleme - Yapay sinir ağları - Doğrusal regression - Olasılıkçı Modeller Tipik araştırma örnekleri: - Billsus ve Pazzani 1998 - Breese ve ark. 1998 - Ungar ve Foster 1998 - Chien ve George 1999 - Getoor ve Sahami 1999 - Pennock ve Horwitz 1999 - Goldberg ve ark. 2001 - Kumar ve ark. 200l - Pavlov ve Pennock 2002 - Shani ve ark. 2002 - Yu ve ark. 2002, 2004

Çizelge 3.3. (**devam**)

Öneri Yaklaşımı	Öneri Tekniği	
	Sezgisel Tabanlı	Model Tabanlı
		- Hofmann 2003, 2004 - Marlin 2003 - Si ve Jin 2003
Hibrit Filtreleme	İçerik tabanlı ve işbirlikçi bileşenleri birleştirerek kullanma - Tahmini derecelendirmelerin doğrusal bileşimi - Çeşitli oylama şemaları - Bir bileşeni bir parçası olarak birleştirmek - Diğerinin sezgiselliği Tipik araştırma örnekleri: - Balabanovic ve Shoham 1997 - Claypool ve ark 1999 - Good ve ark 1999 - Pazzani 1999 - Billsus ve Pazzani 2000 - Tran ve Cohen 2000 - Melville ve ark 2002	İçerik tabanlı ve işbirlikçi bileşenleri birleştiren: - Bir bileşen modelin bir parçası olarak diğerine dahil edilmesi - Birleştirici bir model oluşturmak Tipik araştırma örnekleri: - Basu ve ark 1998 - Condliff ve ark 1999 - Soboroff ve Nicholas 1999 - Ansari ve ark 2000 - Popescu ve ark. 2001 - Schein ve ark 2002

3.4. Öneri Sistemleri Uygulamaları

Kişiselleştirme, Web'in geleceğidir ve endüstriyel uygulamalarda büyük başarılar elde etmiştir. Örneğin, Amazon ve Netflix gibi çevrimiçi mağazalar, kullanıcının geçmişteki tercihlerine veya davranışlarına göre ürünler veya hizmetler için özelleştirilmiş öneriler sunmaktadır (Zhang ve Koren, 2007).

Çizelge 3.4'de öneri sistemlerini kullanan bazı çevrimiçi mağazaların kullandıkları teknolojiler, öneri arayüzü ve öneri yapılan alanların başlıkları görülmektedir.

Çizelge 3.4. Öneri sistemleri uygulamaları (Schaffer, Konstan ve Reidl, 1999)

İş uygulamaları	**Öneri Arayüzü**	**Öneri Teknolojisi**	**Bulunan Öneriler**
Amazon.com			
Bunu Alanlar Bunu da Aldı	Benzer Öğeler	Öğe Tabanlı *Satın Alma İstatistikleri*	Organik Gezinme

Çizelge 3.4. (**devam**)

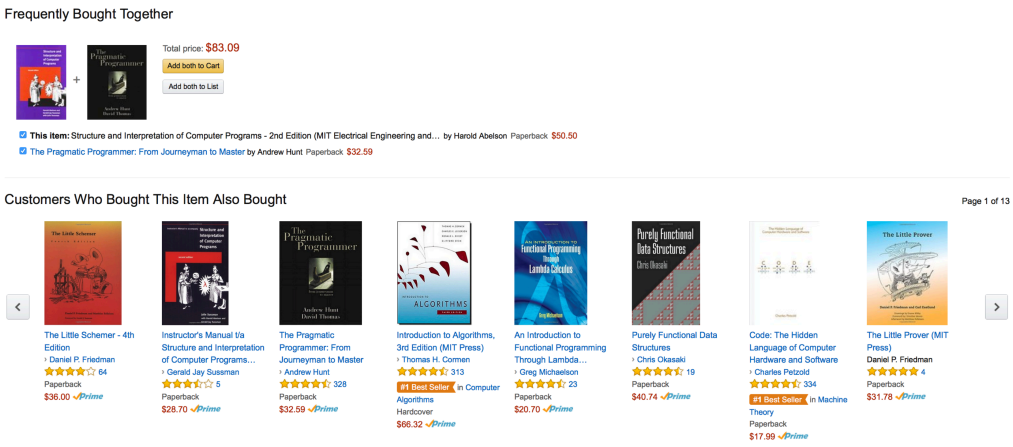
İş uygulamaları	**Öneri Arayüzü**	**Öneri Teknolojisi**	**Bulunan Öneriler**
İzleyici	Email	Öznitelik Tabanlı	Anahtar Kelimeler / Serbest Biçim
Amazon.com Sağlayıcısı	Email	Öznitelik Tabanlı	Oylamalar
Kitap Eşleştirmesi	Top N Listesi	Kullanıcı Tabanlı *Beğenilenler*	İstek Listesi
Kullanıcı Yorumları	Yorum Değerlendirmesi Ortalamaları	Toplam Değerlendirme *Beğenilen Metin*	Organik Gezinme
CDNOW			
Albüm Danışmanı	Benzer Öğeler için Top N Listesi	Öğe Tabanlı *Satın alma istatistikleri*	Organik Gezinme, Anahtar Kelimeler / Serbest Biçim
Benim CDNOW	Top N Listesi	Kullanıcı Tabanlı *Beğenilenler*	Organik Gezinme İstek Listesi
eBay			
Profiline Uygun Olanlar	Yorum Değerlendirmesi Ortalamaları	Toplam Değerlendirme *Beğenilen Metin*	Organik Gezinme
Levis			
Stilini Bul	Top N Listesi	Kullanıcı Tabanlı *Beğenilenler*	İstek Listesi
Moviefinder.com			
Eşleştirme	Benzer Öğeler	Öğe Tabanlı *Editör seçimi*	Bir Öğeye Git
Bizim Önerilerimiz	Satın Alınanların Değerlendirmesi için Top N Listesi	Kullanıcı Tabanlı *Toplam Değerlendirme Beğenileri*	Organik Gezinme, Anahtar Kelimeler / Serbest Biçim, Oylamalar
Reel.com			
Benzer Filmler	Benzer Öğeler	Öğe Tabanlı *Editör seçimi*	Organik Gezinme
Film Haritası	Tarama	Öznitelik Tabanlı *Editör seçimi*	Anahtar Kelimeler / Serbest Biçim

İnternet ortamında öneri sistemlerini başarılı bir şekilde sistemlerine entegre etmiş çevrimiçi mağazalar sanıldığından çok daha fazladır. Bu bölümde öneri sistemlerini başarılı bir şekilde sistemlerine entegre etmeyi başarmış Amazon.com, Netfix.com gibi bazı yabancı çevrimiçi mağazalar ile Türkiye'de faaliyet gösteren N11.com ve Gittigidiyor.com gibi çevrimiçi mağazaların kullandıkları öneri sistemleri ve bu sistemlerin faydaları anlatılmaktadır.

3.4.1. Amazon.com çevrimiçi mağazası

Amazon.com, gerçek zamanlı olarak sattığı hizmet ve ürünler üzerinde hibrit filtreleme sistemleri kullanarak müşterilerine önerilerde bulunmaktadır. Amazon.com'un öneri sistemi her bir kullanıcı için oluşturulmuş profiller sayesinde kişiselleştirilmiş öneriler sunabilmektedir. Amaç kullanıcıların kısa sürede aradıklarını bulmalarını sağlamaktır. Bunu yaparken daha çok ürün satışı sağlanmakta ve kârlılık arttırılmaktadır. Yapılan araştırmalara göre Amazan.com, gelirinin yüzde 35'ini öneri sistemleri üzerinden kazanmaktadır (MacKenzie, 2012).

Şekil 3.3.'de Amazon.com çevrimiçi mağazasının internet sayfasından (https://www.amazon.com) alınmış bir görüntü bulunmaktadır. Amazon.com "Customers Who Bought this Item Also Bought (Bu ürünü alan müşteriler bunları da aldı)" başlığı altında önerilerde bulunulmaktadır. Aynı zamanda "Frequently Bought Together (Sıkça Birlikte Alınanlar)" başlığı ile de öneriler yapılmaktadır.



Şekil 3.3. Amazon öneri sistemi görüntüsü (Amazon.com, 2019)

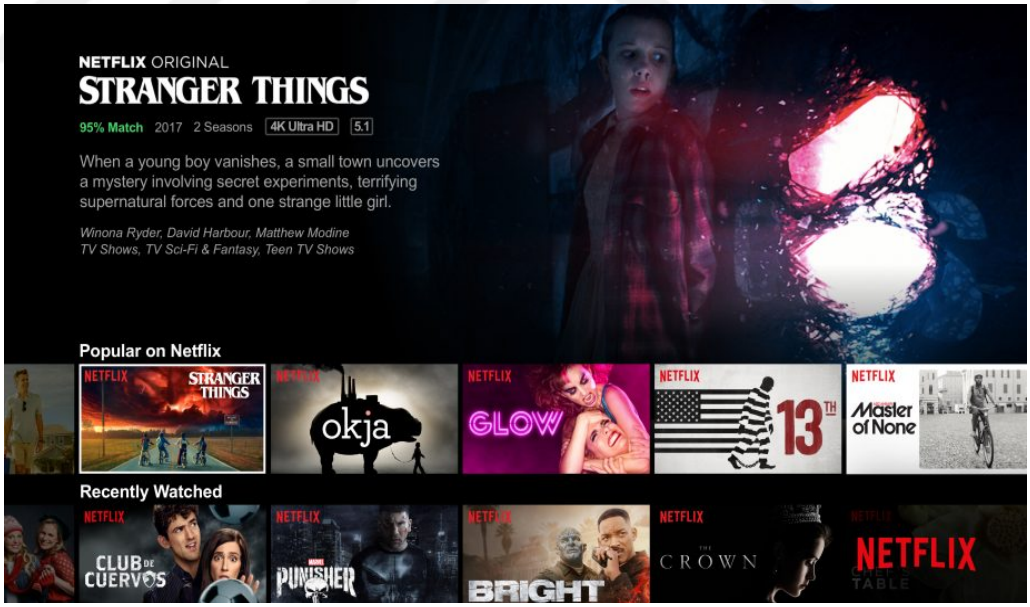
Amazonun kurucusu Jeff Bezos 1998'de yaptığı bir açıklamada: "Çevrimiçi dünyada işletmeler hem müşterilerin tercihlerini hem de satın alma geçmişlerini gözlemleyerek müşterilerle çok derin ilişkiler kurma fırsatına sahip oluyorlar. Böylece müşterinin kişiselleştirilmiş bilgilerini elde edebilir ve bu kişiselleştirilmiş bilgiyi kullanarak müşteriye daha özel hizmet verebilirler. Bunu yapabilirsek, daha yüksek

fiyatlar talep etmedikçe veya daha kötü hizmet sunmadıkça müşteriler rakip web sitelerine gitmeye gerek duymayacaklardır; çünkü kişiselleştirilmiş bir hizmet alabileceklerdir.” demiştir (Porter, 2007)

3.4.2. Netflix.com çevrimiçi film sitesi

Yapılan araştırmalara göre Netflix’te izlenen içeriklerin %75 i öneri sistemleri algoritmalarından gelmektedir (MacKenzie, 2012).

Tüketici araştırmaları, standart bir Netflix üyesinin, bir ya da iki sayfada 10-20 adet başlık okuyup (belki üç tanesinin içeriğine göz atarak) toplamda 60 ila 90 saniye vakit harcadıktan sonra hala ilgisini çekebilecek bir içerik bulmamışsa sistemi terk ettiğini göstermektedir. Netflix, bir üyesinin sistemi terk etmeden en fazla 90 saniyesi olduğunu ve en fazla 3 sayfaya baktığını bildiği için öneri sistemlerine çok önem vermektedir (Gomez-Uribe ve Hunt, 2016).



Şekil 3.4. Netflix web sitesi (Netflix, 2019)

Şekil 3.4.’de görüldüğü gibi Netflix “Popular on Netflix” başlığı altında müşterilerine önerilerde bulunmaktadır.

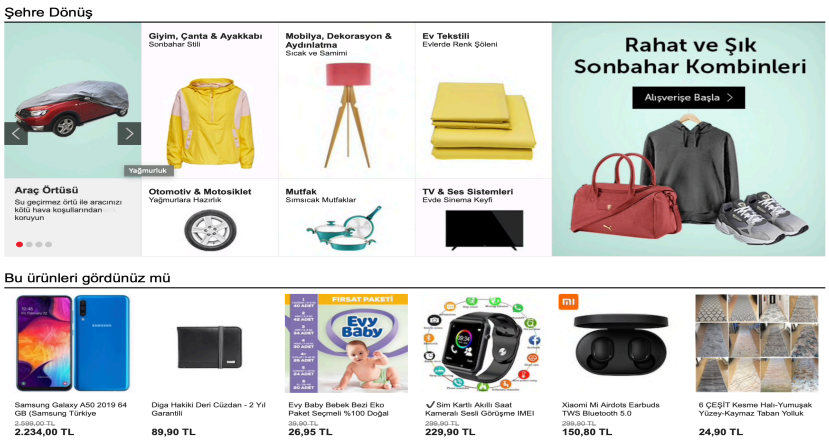
Netflix'in kullanıcıların ilgisini daima canlı tutmak ve onları sistemde daha uzun süre tutup bağlılıklarını arttırmak için öneri sistemlerine her yıl yaklaşık 1 milyar dolar harcadığı düşünülmektedir (Gomez-Uribe ve Hunt, 2016).

Netflix CineMatchSM adında kendi geliştirdiği bir öneri algoritması kullanmaktadır. Netflix 2006 yılında Netflix Prize adında bir yarışma düzenlemiştir. CineMatchSM adlı kendi algoritmasından %10 daha verimli çalışan bir algoritma öneren kişiye 1 milyon dolarlık bir ödül vereceğini açıklamıştır. 2009 yılında Gravity R&D'nin kurucu takımı (BellKor'un Pragmatik Kaosu) dünya genelinde 18 bin ekibin yarıştığı "Netflix Prize"da birinci olarak ödülü kazanmıştır (Netflix Prize, 2009; Gravity R&D, 2016)

Netflix kendi sitesinde şöyle bir tanımlama yapmaktadır: "İşimiz, ilginizi çekecek dizileri, programları veya filmleri bulmanıza yardımcı olmak için kişiselleştirilmiş öneriler sunan abonelik hizmeti modelidir. Bunu yapmak için tescilli, karmaşık bir öneri sistemi oluşturduk." (Netflix, 2016).

3.4.3. N11.com çevrimiçi alışveriş sitesi

Açık pazar platformu olan N11.com öneri sistemleri için kişileştirilmiş öneri sistemleri alanında çalışan prudsys RDE (Real-Time Decisioning Engine) ile çalışmaktadır (N11 Kurumsal, 2014).



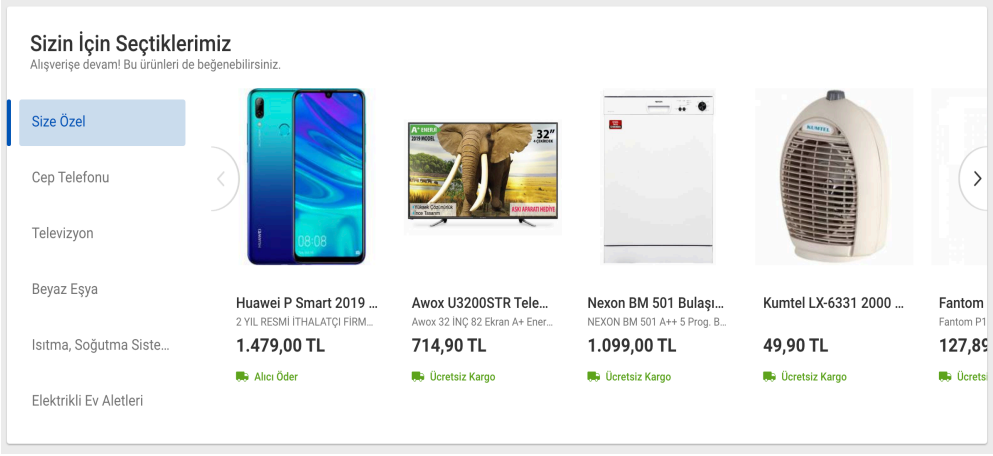
Şekil 3.5. N11.com çevrimiçi mağazası (N11.com, 2019)

Şekil 3.5.'de görüldüğü gibi N11.com sitesi "Bu ürünleri gördünüz mü" başlığı altında müşterilerine önerilerde bulunmaktadır.

Şirketin bilgi teknolojileri genel müdür yardımcısı olan Cenk Çivici, kişileştirilmiş öneri sistemleri ile gerçekleştirilen uygulamalarda olumlu geri dönüşler sağlandığını ve mağazada önerilere dayanan satışlarda belirgin bir şekilde artış olduğunu açıklamıştır. N11.com, kendi internet sitelerinde yaptıkları açıklamayla prudsys RDE'nin ihtiyaçlarını fazlasıyla karşıladığını ve bu öneri sistemleri sayesinde satışlarını bir ay gibi kısa bir sürede %10 arttırdıklarını söylemişlerdir (N11 Kurumsal, 2014).

3.4.4. Gittigidiyor.com çevrimiçi alışveriş sitesi

eBay'in Türkiye şubesi olan Gittigidiyor.com çevrimiçi alışveriş sitesi güncel trendleri göz önüne alarak, Netflix ödülünü kazanan Grativy R&D ile önerilerini kişiselleştirmek için 2016 yılında anlaşma yapmıştır. Gravity R&D'nin sistemi şu anda Gittigidiyor.com'un tüm site içeriğinde kullanılmaktadır (Aimgroup, 2016; Gravity R&D, 2016).



Şekil 3.6. Gittigidiyor.com çevrimiçi mağazası (Gittigidiyor.com, 2019)

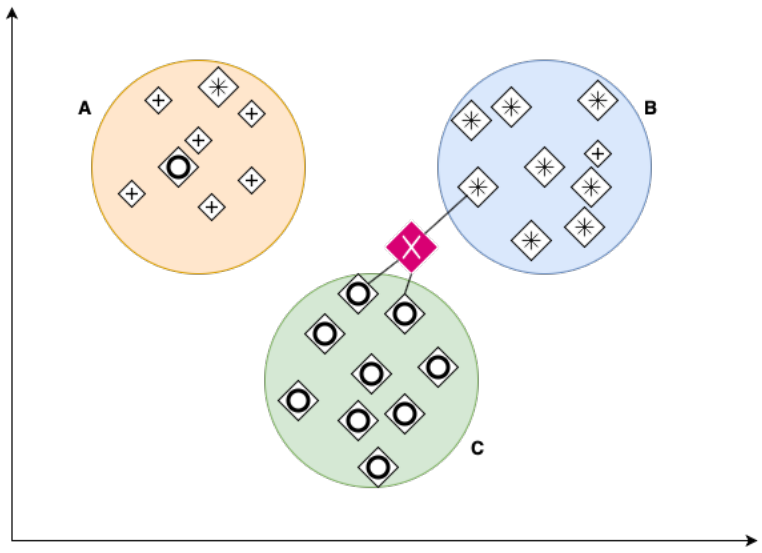
Şekil 3.6.'da görüldüğü gibi Gittigidiyor.com sitesi "Sizin İçin Seçtiklerimiz" başlığı altında müşterilerine önerilerde bulunmaktadır.

Şu an yaklaşık 150 terabayt verisi bulunduğunu belirten GittiGidiyor.com Teknoloji Direktörü Oğuzhan Poyrazoğlu, kendi sitelerinde şöyle bir açıklama yapmıştır: "Bu veriler üzerinde çalışan algoritmalar kullanıcıların alışveriş ve ziyaret verilerini analiz ederek kullanıcılara özel alanlar oluşturulmaktadır. 'Sizin İçin Seçtiklerimiz' bölümünde sunulan ürünlerin tamamı kullanıcıların daha önceki ziyaret ve alışveriş davranışları dikkate alınarak, arkada çalışan akıllı bir algoritma ile özel olarak belirleniyor. Aynı algoritma üyelere gönderilen e-postalarda da kullanılıyor." (Gittigidiyor Kurumsal, 2017)

3.5. *k*-NN Algoritması

K en yakın komşu algoritması aslında bir sınıflandırma algoritmasıdır. Sınıflandırma işlemi bittikten sonra yeni gelen veriye en yakın k tane komşu seçilir. En çok hangi sınıfta komşusu varsa yeni gelen veri o sınıfa atanır (Larose, 2014).

Komşuluk seçimi aşamasında *k* en yakın komşu algoritması öneri sistemlerinde en çok kullanılan algoritmadır (Milli, 2013). *K* en yakın komşu algoritması öğelerin birbirine olan uzaklıklarına göre belli sayıda (k tane) komşu seçildikten sonra yeni gelen öğenin hangi sınıfa ait olduğunu bulmaya yarayan bir algoritmadır.



Şekil 3.7. *k*-EYK algoritması çalışma şekli (k=4)

Şekil 3.7.'de üç sınıfa ayrılmış bir veri setine yeni eklenen X öğesinin hangi sınıfa ait olduğu *k*-EYK algoritmasına göre hesaplanmaktadır. k=3 alındığında X öğesine olan uzaklıklar hesaplanmış ve en yakın 3 komşu öğe seçilmiştir. Bu seçilen komşulardan 2 tanesi C sınıfına ait 1 tanesi B sınıfına ait olduğu için X öğesi daha çok komşuya sahip olduğu C sınıfına dahil edilmektedir.

Bu tez kapsamında *k*-EYK algoritması hem içerik tabanlı filtrelemede hem de işbirlikçi filtrelemede kullanılmıştır. *k*-EYK algoritması için en önemli faktör komşu sayısı olan *k* değerinin seçilmesidir. *K* değerinin çok yüksek seçilmesi benzerlik oranının az olduğu öğelerin hesaplamaya dahil edilmesine ve doğal olarak yapılan tahminlerin kalitesinin düşmesine sebep olmaktadır. *K* değerinin çok küçük seçilmesi ise benzerliğin yüksek olduğu bazı öğelerin hesaplamaya dahil edilmemesine neden olup yapılan tahminlerin çeşitliliğinin az olmasına ve daha düşük kalitede olmasına neden olabilmektedir. Herlocker ve ark. (2002), *k*-EYK için *k* değerinin 20 ile 50 arasında seçilmesinin en iyi sonuçları verdiğini söylemişlerdir. Bu çalışmaya dayanılarak yapılan deneylerde *k* değeri 40 olarak alınmıştır.

3.6. Kullanılan Değerlendirme Ölçütleri

Öneri sistemlerinde geliştirilen algoritmaların kalitesi doğruluk, hız ve kişiselleştirme seviyesine göre belirlenir. Öneri sistemlerinin kalitesini belirleyen bu kriterlerden en önemlisi doğruluktur (Milli, 2013). Doğruluğu ölçmek için literatürde birçok yöntem önerilmiştir. Tez kapsamında bu yöntemlerden Ortalama Mutlak Hata ve Hata Karelerinin Ortalamasının Karekökü yöntemleri kullanılmıştır.

3.6.1. Ortalama mutlak hata

Ortalama mutlak hata, hataların mutlak değerlerinin ortalamasıdır. Kullanıcının yapmış olduğu değerlendirmenin ne kadar doğru tahmin edildiğini ölçmek için kullanılmaktadır.

$$OMH = \frac{1}{N}\sum_{i=1}^{N} |R_{u,i} - \overline{R_{u,i}}| \qquad (3.15)$$

Denklem (3.15)'de verilen OMH hata oranı hesaplama denkleminde; N test verilerinin sayısı, $R_{u,i}$ test verisinde yer alan u kullanıcısının i filmi için yaptığı değerlendirme ve $\overline{R_{u,i}}$ u kullanıcısının i filmi için yapacağı değerlendirmenin algoritma tarafından yapılan tahminidir. Veri setinde bulunan gerçek değerlendirmeler ile tahmin edilen değerlendirmeler arasındaki sayısal uzaklıkların toplanıp test verisinde bulunan toplam veri sayısına (N) bölünmesiyle OMH değeri bulunmaktadır.

3.6.2. Hata karelerinin ortalamasının karekökü

Doğruluk ölçütlerinde gerçek değerlendirme ile tahmin edilen değerlendirme arasındaki farktan çok uzaklık önemlidir. Aradaki farkın negatif veya pozitif olması bir şey ifade etmemektedir. Bu sebeple OMH yönteminde mutlak değer kullanılmıştır. Hata karelerinin ortalamasının karekökü (HKOK) yönteminde ise aradaki farkların karesi alınarak negatiflikten durumu ortadan kaldırılmaktadır (Milli, 2013).

$$HKOK = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_{u,i} - \overline{R_{u,i}})^2} \qquad (3.16)$$

Denklem (3.16)'da $R_{u,i}$ test verisinde yer alan u kullanıcısının i filmi için verdiği derecelendirme, $\overline{R_{u,i}}$ algoritmanın tahmin ettiği değerdir. Denklemde hata farklarının karesi alınıp negatiflikten kurtulduktan sonra aritmetik ortalama işlemi yapılmaktadır. Daha sonra bulunan hata karelerinin ortalaması alınmaktadır.

3.7. Genetik Algoritma

Artan veriler sayesinde günümüzde veri işlemenin maliyetleri giderek artmaktadır. Maliyetleri düşürmek için yeni çözüm arayışları yapılmaktadır. Öneri sistemlerinde bir çözüm olarak öne sürülen evrimsel algoritmaların (evolutionary algorithm) kullanımı giderek artmaktadır. Bir yapay zekâ ve makina öğrenme algoritması olan Genetik Algoritma sıkça tercih edilen yöntemlerindendir (Emel ve Taşkın, 2002).

"Genetik Algoritma" terimi ilk olarak Bagley tarafından 1967 yılında kullanılmıştır (Tuğ, 2005). 1975 yılında John Holland tarafından temel ilkeleri ortaya

konulan GA, başta optimizasyon problemleri olmak üzere çok çeşit problemde başarılı bir şekilde uygulanmıştır (Mitchell, 1998).

GA canlı yaşamdaki kalıtımsal özelliklerin nesilden nesile aktarılmasında yaşanan doğal seleksiyon ve mutasyonu taklit edilerek oluşturulmuş bir algoritmadır. Aslında popülasyon temelli bir optimizasyon yöntemidir. Genetik algoritma nesil sayısı kadar iterasyona tabi tutulmaktadır. Her iterasyonda çözüm kümesini (kromozom) oluşturan katsayılar (gen) çeşitli evrimsel işlemlere (çaprazlama, mutasyon) tabi tutularak yeni değerlere dönüştürülerek uygunluk fonksiyonu (fitnes fonksiyonu) hesaplanmaktadır. Bu işlemler belli bir nesil sayısına veya istenen uygunluk değerine ulaşıncaya kadar devam etmektedir (Jang, 1997).

Genetik algoritmalar olasılık kurallarına göre çalışıp yalnızca uygunluk fonksiyonuna ihtiyaç duymaktadırlar. GA çözüm uzayının çok geniş, değişken sayısının çok fazla olduğu problemlere de kolayca uygulanabilmektedir (Goldberg, 1989). Çünkü kromozomlar sayesinde bir önce değerlerin kalıtımsal olarak en iyi olanları bir sonraki nesile aktarıldıkları için genetik algoritmalar tüm olası çözümleri denemek zorunda değillerdir. Tüm çözüm uzayı taranmadığı için çok daha kısa sürede çözüme ulaşmaktadırlar. GA en iyi sonuç veya sonuca en yakın çözümü bulmak için sıklıkla kullanılmaktadır (Tuğ, 2005).

Genetik algoritma oluşturulurken popülasyon boyutu, çaprazlama oranı, mutasyon oranı, kuşak farkı, seçim yöntemi ve fonksiyon ölçeklemesi gibi parametrelerin seçilmesi problemin çözümünde etkin tol oynamaktadır (Emel ve Taşkın, 2002).

Popülasyon boyutu: Popülasyon boyutunun çok küçük seçilmesi yerel bir en iyiye takılmaya neden olabilirken çok büyük seçilmesi işlem süresini çok arttırıp maliyetli olabilmektedir. Schaffer ve ark. (1989) yapmış oldukları araştırma sonucunda popülasyon boyutunun belli bir aralıkta (20 - 30) seçilmesinin genel olarak sonuçlar üzerinden olumlu bir etki yaptığını belirtmişlerdir (Emel ve Taşkın, 2002).

Buna dayanarak tez kapsamında yapılan çalışmada kullanılan genetik algoritmada popülasyon boyutu 20 seçilmektedir.

Çaprazlama oranı: Çaprazlama, mevcut kromozomların çaprazlanarak yeni genler oluşturulma işlemidir. Çaprazlama oranı (P(c)) bir kromozomum ne kadarının değiştirileceğidir. Çaprazlama oranı büyük seçilirse bazı iyi kromozomlar bozulabileceği gibi küçük seçilmesi durumunda da istenen çözüme ulaşmak çok uzun sürebilmektedir. Tez kapsamında yapılan çalışmada kullanılan genetik algoritmada çaprazlama oranı 0.8 seçilmektedir.

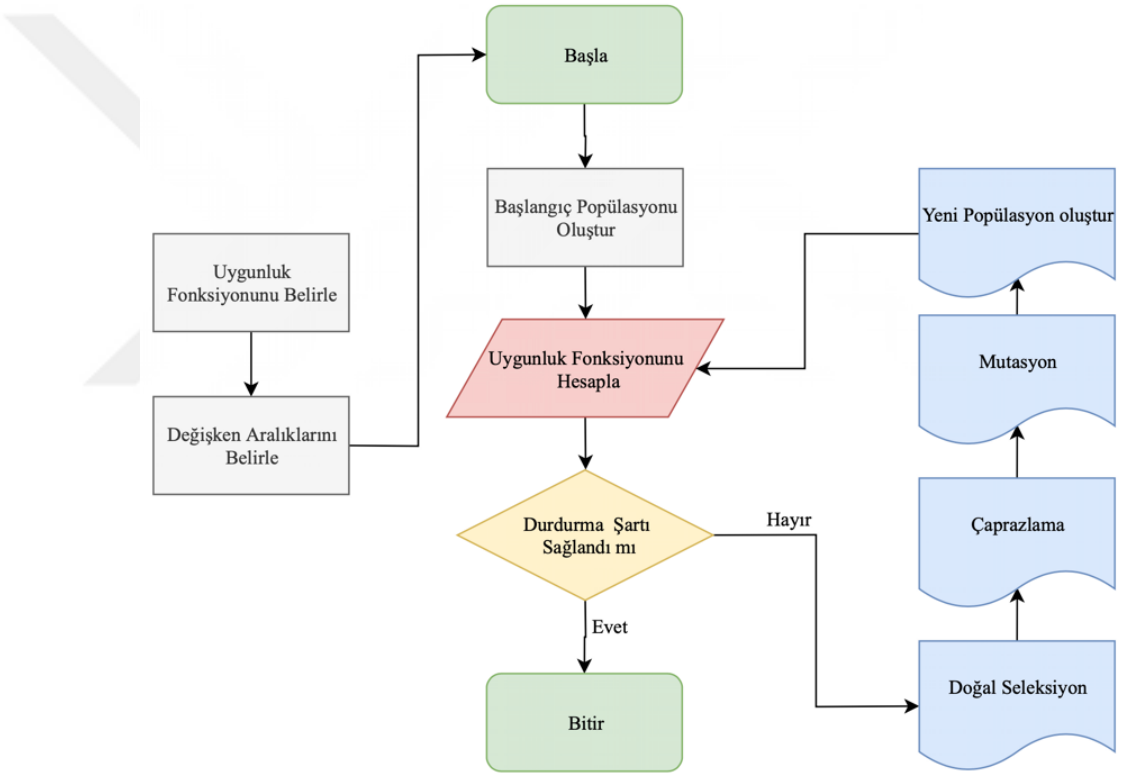
Mutasyon oranı: Mutasyon, nesilden nesile aktarılma esnasında kromozomlarda belli orandaki (mutasyon oranı) bitin değiştirilme işlemidir. Mutasyon oranı büyük seçilirse mevcut çözümü bulmak uzun sürer ve algoritma rastlantısal bir hal almaktadır. Aynı zamanda genetik çeşitlilik artacağı için çözüme daha erken varma ihtimali de vardır. Tez kapsamında yapılan çalışmada kullanılan genetik algoritmada mutasyon oranı 0.1 seçilmektedir.

Kuşak farkı: Her nesildeki yeni kromozomların sayısını belirtir. Yüksek bir değer seçilmesi bir önceki nesilden olası gelebilecek iyi çözümlerin göz ardı edilmesine neden olabilir. Tez kapsamında yapılan çalışmada kullanılan genetik algoritmada kromozomların sadece en iyisi bir sonraki nesile hiç değişmeden aktarılmaktadır.

Seçim yöntemi: Eski kuşak üç farklı şekilde yenilenmektedir. “Kuşaksal” yönteminde tüm kromozomlar evrimsel işlemler sayesinde yenilenmektedir. “En Uygun” yönteminde en iyi kromozomlar değiştirilmeden bir sonraki nesile aktarılmaktadır. “Denge Durumu” yönteminde her yeni nesil için sadece belli sayıda kromozom değiştirilmektedir. Tez kapsamında yapılan çalışmada kullanılan genetik algoritmada kuşaksal yöntemi kullanılmaktadır.

Fonksiyon ölçeklemesi: Doğrusal ve üstsel ölçekleme diye iki şekilde uygulanır. Uygun ölçeklemenin seçilmesi uygunluk fonksiyonunun daha başarılı şekilde çözülmesine etki etmektedir.

Genetik algoritma bilinen bir sürü probleme başarılı bir şekilde uygulanmıştır. Gezgin satıcı problemi, yerleşim tasarımı problemi, postacı problemi, ekip eşleştirme problemi, araç rota problemi, küme bölme modeli problemi, sırt çantası problemi vb. problemler örnek olarak gösterilebilir (Emel ve Taşkın, 2002; Çetin ve ark., 2008). Kullanılan genetik algoritmanın akış şeması Şekil 3.8. de gösterilmiştir.



Şekil 3.8. Genetik algoritma akış şeması

Genetik algoritmanın çalışma adımları:

1- Başlangıçta çözülmesi istenen fonksiyon belirlenir. Bu uygunluk fonksiyonudur.
2- Değişkenler için alt ve üst sınırlar belirlenir.
3- Başlangıç popülasyonunu rastgele değerlerden oluşturulur.
4- Popülasyon için uygunluk değeri hesaplanır.

5- Durdurma şartının (hedef nesil sayısı veya hedef başarı değeri) sağlanıp sağlanmadığı kontrol edilir.
6- Şart sağlandıysa Adım 9'a git.
7- Doğal seleksiyon-çaprazlama-mutasyon işlemlerini kromozomlar üzerinde uygulayarak yeni popülasyonu belirle.
8- Adım 4'e git.
9- Algoritmayı bitir.

Genetik algoritmalarda tüm olası çözümler kromozom (ikili bit bizisi) olarak adlandırılmaktadır. Genetik algoritmalarda her nesilde popülasyonlar tutularak işlem yapılmaktadır. Her popülasyonun içinde birden fazla çözüm noktaları olabilmektedir. Her nesilde seleksiyon, çaprazlama ve mutasyon işlemleri ile bir önceki popülasyondan bir sonraki popülasyon oluşturulmaktadır. Her nesilde en iyi çözümler bir sonraki nesile aktarılarak daha iyiye doğru gidilmektedir (Emel ve Taşkın, 2002).

Genetik algoritma hedeflenen nesil sayısına (veya hedeflenen uygunluk değerine) ulaşıncaya kadar belli bazı adımlar şeklinde çalışmaktadır (Goldberg, 1989; Jang, 1997; Çolak, 2010). Genelde durdurma şartı nesil sayısı olarak seçilip belirlenen nesil sayısına ulaşıldığında algoritma sonlandırılmaktadır. Elde edilen sonuçlar arasından en iyisi (Global Best) çözüm olarak sunulmaktadır.

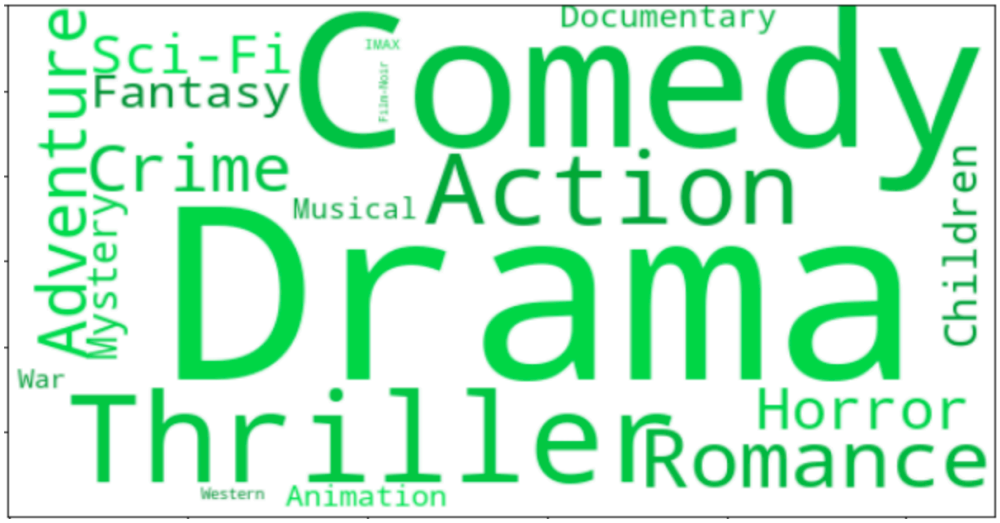
Genetik algoritma film öneri sistemleri için sıklıkla kullanılmaktadır. Örneğin Alhijawi ve Kilani (2016) ile Bobadilla ve ark. (2011), benzerlik hesaplama aşamasında genetik algoritma kullanmışlardır. Sadeghi ve Asghari (2017), kullanıcıların açık ve kapalı özelliklerinin ağırlıklarının bulunması için genetik algoritma kullanılan bir hibrit film önerisi geliştirmiştir. Tez kapsamında genetik algoritma işbirlikçi ve içerik tabanlı filtreleme algoritmalarından gelen tahminleri bir araya getirmede en uygun oranların bulunması için kullanılmıştır.

3.8. Kullanılan Veri Seti

MovieLens veri setleri eğitim, araştırma ve endüstride yaygın olarak kullanılmaktadır. Popüler basın programlama kitaplarında, geleneksel ve çevrimiçi kurslarda ve yazılımlarda aktif olarak kullanılmaktadırlar. Her yıl yüz binlerce kez indirilmektedirler. Bu veri setleri, 1997'de piyasaya sürülmesinden bu yana birçok deneyi barındıran aktif bir araştırma platformu olan MovieLens film öneri sisteminde, kullanıcıların katılımının aktif bir ürünüdür (Harper ve Konstan, 2016).

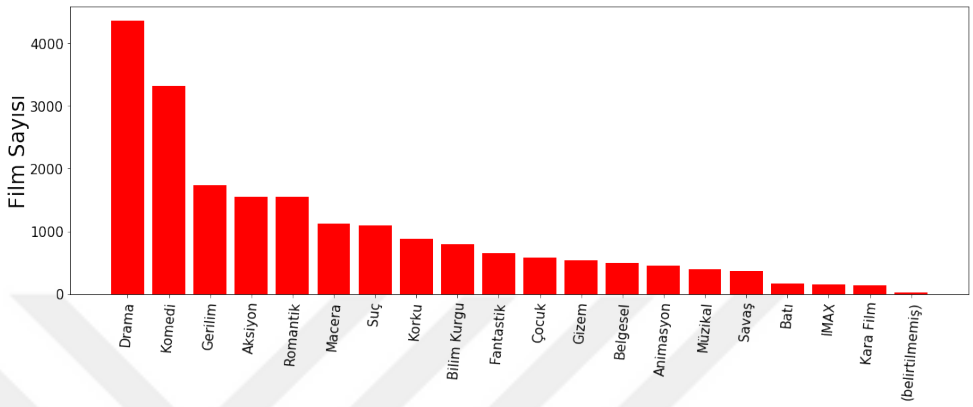
MovieLens veri setleri yoğun bir şekilde indirilmekte (2014'te 140.000'den fazla indirme) ve araştırma literatüründe (Google Akademik'teki "MovieLens" e 7.500'den fazla) atıfta bulunulmaktadır (Harper ve Konstan, 2016).

Veri setleri değerlendirme sayısına göre dört farklı büyüklükte yayınlanmıştır. Bunlar 100K, 1M, 10M ve 20M olarak bilinen MovieLens veri setleridir (Harper ve Konstan, 2016). Bu tez kapsamında önerilen yöntemler için yapılan deneylerde 100K olan veri seti kullanılmaktadır. Yapılan deneylerde MovieLens 100K veri setinde bulunan ratings.csv (Çizelge 3.5.), movies.csv (Çizelge 3.6.) ve tags.csv (Çizelge 3.7.) dosyalarındaki veriler kullanılmıştır.



Şekil 3.9. Türler için oluşturulmuş kelime bulutu

Veri setinde toplam 19 film türü bulunmaktadır. Şekil 3.9.'da türler için oluşturulmuş kelime bulutu görülmektedir. Şekil 3.9.'da tür isimlerinin karakter boyutu o türe sahip film sayısı ile doğru orantılı olacak şekilde büyük veya küçük yapılmaktadır. Hangi türde ne kadar film olduğu Şekil 3.10.'da gösterilmektedir.



Şekil 3.10. Türlerin filmler üzerindeki dağılımı

Bu tez çalışması kapsamında kullanılan MovieLens 100k veri setinde bulunan ratings.csv dosyasında 672 kullanıcı tarafından 9126 film için yapılmış 100021 değerlendirme bulunmaktadır. Bu veri setinde her kullanıcı en az 20 filme oy vermiştir.

Çizelge 3.5. ratings.csv dosyasından ilk 20 satır

kullanıcıId	filmId	Değerlendirme Puanı	Zaman
1	31	2.5	1260759144
1	1029	3.0	1260759179
1	1061	3.0	1260759182
1	1129	2.0	1260759185
1	1172	4.0	1260759205
1	1263	2.0	1260759151
1	1287	2.0	1260759187
1	1293	2.0	1260759148
1	1339	3.5	1260759125
1	1343	2.0	1260759131
1	1371	2.5	1260759135
1	1405	1.0	1260759203
1	1953	4.0	1260759191

Çizelge 3.5. (**devam**)

kullanıcıId	filmId	Değerlendirme Puanı	Zaman
1	2105	4.0	1260759139
1	2150	3.0	1260759194
1	2193	2.0	1260759198
1	2294	2.0	1260759108
1	2455	2.5	1260759113
1	2968	1.0	1260759200
1	3671	3.0	1260759117

$$R = n_u \times n_{film} \tag{3.17}$$

Denklem (3.17)'de *R* matrisi tüm değerlendirmelerin matrisi, n_u veri setindeki toplam kullanıcı sayısı ve n_{film} veri setindeki toplam film sayısıdır. Bu, matrisin 672 x 9126 = 61326672 tane değerlendirmeye sahip olması gerektiği anlamına gelmektedir. Ama MovieLens 100K veri setinde 100021 değerlendirme bulunmaktadır. Öneri sistemlerinde bir sorun olan seyreklik (sparsity) %98,36 oranındadır.

$$\frac{n_{sum}}{n_u \times n_{film}} \times 100 = 1.63 \tag{3.18}$$

Denklem (3.18)'de n_{sum} veri setindeki kullanıcıların filmlere yaptığı değerlendirmelerin toplam sayısıdır, değeri 100021'dir. Toplam değerlendirme sayısı, olması gereken değerlendirme sayısına bölünüp 100 ile çarpıldığında veri setinin doluluk oranı %1,63 olarak elde edilmektedir.

Çizelge 3.6. movies.csv dosyasından ilk 20 satır

filmId	Başlık	Tür
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action|Crime|Thriller

Çizelge 3.6. (**devam**)

filmId	Başlık	Tür
7	Sabrina (1995)	Comedy|Romance
8	Tom and Huck (1995)	Adventure|Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action|Adventure|Thriller
11	American President, The (1995)	Comedy|Drama|Romance
12	Dracula: Dead and Loving It (1995)	Comedy|Horror
13	Balto (1995)	Adventure|Animation|Children
14	Nixon (1995)	Drama
15	Cutthroat Island (1995)	Action|Adventure|Romance
16	Casino (1995)	Crime|Drama
17	Sense and Sensibility (1995)	Drama|Romance
18	Four Rooms (1995)	Comedy
19	Ace Ventura: When Nature Calls (1995)	Comedy
20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller
…	…	…

İçerik tabanlı filtreleme algoritmamızda filmler için öznitelik olarak kullanılan tür ve yıl bilgileri MovieLens 100K veri setinde bulunan, Çizelge 3.6.'da bir kesiti verilmiş olan movies.csv dosyasından, anahtar kelimeler bilgisi Çizelge 3.7.'de bir kesiti verilmiş olan tags.csv dosyasından ve oylanma sayıları Çizelge 3.5.'de bir kesiti verilmiş olan yüz bin oylama içeren ratings.csv dosyası üzerinde yapılan hesaplamalar ile elde edilmiştir.

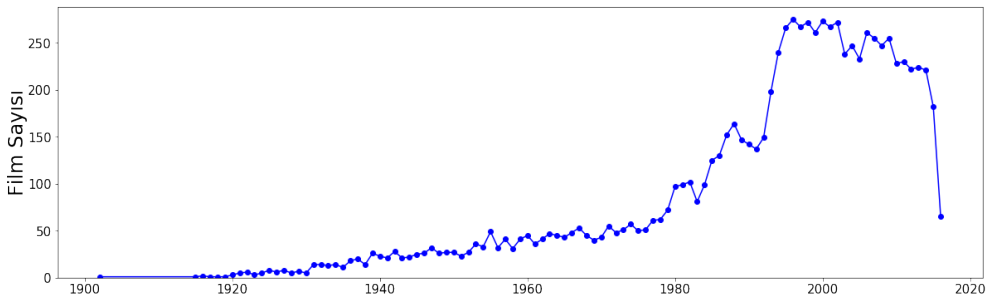
Çizelge 3.7. tags.csv dosyasından ilk 20 satır

kullanıcıId	filmId	Anahtar Kelimeler	Zaman
15	339	sandra 'boring' bullock	1138537770
15	1955	dentist	1193435061
15	7478	Cambodia	1170560997
15	32892	Russian	1170626366
15	34162	forgettable	1141391765
15	35957	short	1141391873
15	37729	dull story	1141391806
15	45950	powerpoint	1169616291

Çizelge 3.7. (**devam**)

kullanıcıId	filmId	Anahtar Kelimeler	Zaman
15	100365	activist	1425876220
15	100365	documentary	1425876220
15	100365	uganda	1425876220
23	150	Ron Howard	1148672905
68	2174	music	1249808064
68	2174	weird	1249808102
68	8623	Steve Martin	1249808497
73	107999	action	1430799184
73	107999	anime	1430799184
73	107999	kung fu	1430799184
73	111624	drama	1431584497
73	111624	indie	1431584497
…	…	…	…

Veri setinde 1902-2016 yılları arasında vizyona girmiş filmler bulunmaktadır. Filmlerin yıllara göre dağılımları Şekil 3.11.'de gösterilmiştir.



Şekil 3.11. Yıllara göre film dağılımları

Tez kapsamında yapılan deneylerde MovieLens 100K veri setinin %75'i eğitim verisi (algoritmayı eğitmek için) ve %25'i test verisi (algoritmanın doğruluğunu ölçmek için) olarak kullanılmıştır.

3.9. Önerilen Yöntemler

Tez kapsamında hibrit bir film öneri sistemi sunulmaktadır. Önerilen film öneri sistemi içerik tabanlı filtreleme ve işbirlikçi filtreleme yöntemlerinin tahminlerininin bir araya getirilmesi ile oluşturulmaktadır. İçerik tabanlı ve işbirlikçi filtreleme yöntemlerini bir araya getirmenin farklı yolları vardır. Bu birleştirme yöntemleri hakkında Bölüm 3.1.3.'de bilgi verilmiştir. Bu yollardan bir tanesi içerik tabanlı ve işbirlikçi filtreleme ile elde edilmiş tahminlerin belli bir oranda bir araya getirilmeleridir (Miranda ve ark., 1999; Adomavicius ve Tuzhilin, 2005). İki farklı algoritmanın bir araya getirilmesinde kullanılacak katsayılar (oranlar) genetik algoritma ile optimize edilmektedir.

Önerilen hibrit yöntem diğer yöntemlerle karşılaştırılmış ve daha başarılı olduğu deneysel sonuçlar ile ortaya konulmaktadır. Önerilen hibrit yöntemden önce hibrit sistemin bileşenleri olan önerilen içerik tabanlı filtreleme ve işbirlikçi filtreleme yöntemleri açıklanmaktadır.

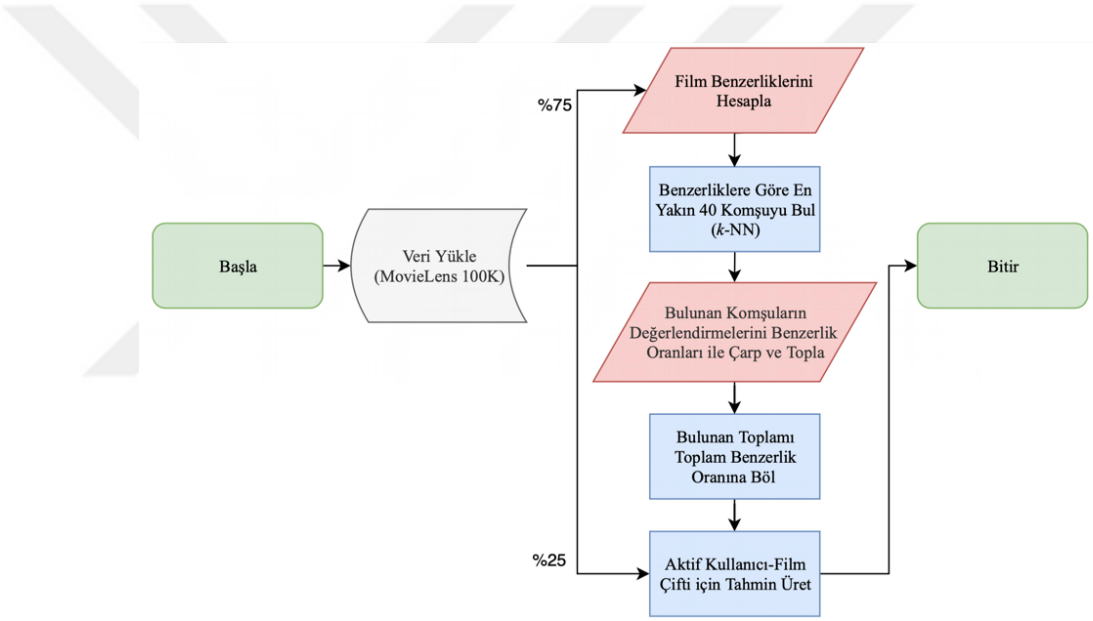
3.9.1. Önerilen içerik tabanlı filtreleme yöntemi

İçerik tabanlı filtreleme kullanan film öneri sistemlerinde filmlerin öznitelikleri kullanılarak aralarındaki benzerlikler hesaplanmaktadır. Daha sonra kullanıcıya izlediği filmlere en çok benzeyen filmler önerilir. Öznitelik olarak literatürde film tipi, yazar, yapım şirketi, anahtar kelimeler, tür, yıl vb. bilgiler sıklıkla kullanılmıştır (Debnath ve ark., 2008; Luo ve ark., 2008; Uluyağmur, 2012).

Debnath ve ark. (2008), yaptıkları çalışmada öznitelik olarak film tipi, yazar ve yapım şirketini kullanılmışlardır. Uluyağmur (2012), yaptığı çalışmada filmlerin oyuncu, tür, yönetmen, anahtar kelime, yapım yılı, yayınlandığı kanal ve yayınlandığı zaman dilimi özniteliklerini kullanmıştır.

Tez kapsamında yapılan deneylerde kullanılan MovieLens 100K veri seti içinde anahtar kelimeler, türler ve yıl bilgileri bulunmaktadır. Filmlerin toplam

oylanma sayıları (bir filmin toplamda kaç kişi tarafından değerlendirildiği sayısı) ratings.csv dosyasında bulunan yüz bin oylama içerisinden hesaplanarak bulunmuştur. Literatürde performansı arttırmak için oylanma sayıları kullanılarak birkaç film öneri sistemi geliştirilmiştir. Örneğin Mulik (2013), filmlerin oylanma sayılarını öneri yapma aşamasında filmleri sıralamak için kullanmıştır. Pera ve Ng (2013), GroupRem adında bir film öneri sistemi geliştirmiştir. GroupRem kişilerin ilgi alanlarını filmlerin öznitelikleri üzerinden bularak oylanma sayılarına göre önerilerde bulunmak üzerine tasarlanmıştır. Bu tez kapsamında önerilen içerik tabanlı filtreleme algoritmasında oylanma sayıları benzerlik aşamasında filmler arasındaki benzerliği bulmak için diğer öznitelikler ile beraber kullanılmıştır.



Şekil 3.12. İçerik tabanlı filtreleme algoritması akış şeması

İçerik tabanlı filtrelemede tüm özniteliklere göre benzerlik hesaplamaları yapılmış ve deneysel olarak karşılaştırılmıştır. Önerilen içerik tabanlı filtreleme algoritmasına ait akış şeması Şekil 3.12.'de görülmektedir. İçerik tabanlı filtrelemede algoritmasında MovieLens 100K veri setinin %75'i algoritmayı eğitmek için eğitim verisi olarak kullanılmıştır. Veri setinin geri kalan %25'lik kısmı daha sonra yapılacak doğruluk testleri (OMH ve HKOK hata oranları tespiti) için kullanılmıştır.

Çizelge 3.8. Tür listesi

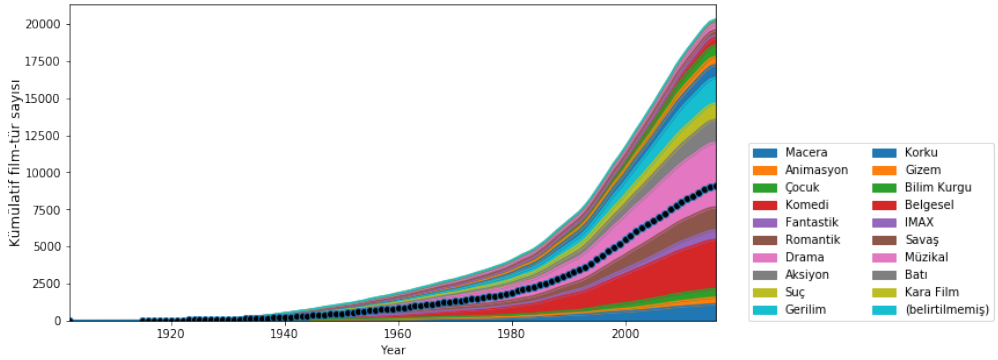
Sıra	Tür	Film Sayısı
1	Drama	4365
2	Komedi	3315
3	Gerilim	1729
4	Romantik	1545
5	aksiyon	1545
6	Macera	1117
7	Suç	1100
8	Korku	877
9	Bilim Kurgu	792
10	Fantastik	654
11	Çocuk	583
12	Gizem	543
13	Belgesel	495
14	Animasyon	447
15	Müzikal	394
16	Savaş	347
17	Batı	168
18	IMAX	153
19	Kara Film	133
20	(belirtilmemiş)	18

Kullanılan veri setinde 20 tür bulunmaktadır. Türü belirtilmeyen filmler için "belirtilmemiş" adında bir tür kullanılmıştır. Çizelge 3.5.'de türlerin listesi görünmektedir. Her bir film tek bir türde olabileceği gibi birden fazla türe de sahip olabilmektedir (Çizelge 3.6.).

Çizelge 3.9. "Toy Story (1995)" filmi için tür vektörü

1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

İki film arasında tür bazlı bir benzerlik hesaplaması yapılabilmesi için her bir film "0" ve "1"lerden oluşan yirmi bit uzunluğunda bir vektör olarak oluşturulmaktadır. Filmin sahip olduğu tür bilgisi için "1" sahip olmadığı tür bilgisi için de "0" rakamı kullanılmıştır. Çizelge 3.9.'da görülen "Toy Story (1995)" isimli film için oluşturmuş tür vektörü gösterilmektedir.



Şekil 3.13. Türlerin yıllara göre dağılımı

Kullanılan veri setinde 1902-2016 yılları arasında yayınlanmış 9125 film bulunmaktadır. Filmlerin yıllara göre dağılımı Şekil 3.13.'de gösterilmiştir. Yıl bilgilerinin sayısal değerleri büyük olduğu için aradaki fark büyük olmaktadır. Kullanılan kosinüs benzerliğinde ortaya çıkabilecek olumsuzluklar göz önüne alınarak filmlerin yayınlandıkları yıllar arasındaki farklar [0-1] aralığında normalize edilmiştir. Önerilen içerik tabanlı filtreleme yönteminde normalize edilmiş [0-1] aralığındaki benzerlik katsayıları kullanılmıştır.



Şekil 3.14. Anahtar kelimeler için oluşturulmuş kelime bulutu

Kullanılan MovieLens 100K veri setinde bulunan tags.csv (Çizelge 3.7.) dosyasında 1296 adet film için toplam 856 adet anahtar kelime bilgisi bulunmaktadır.

Şekil 3.14.'de anahtar kelimeler için oluşturulmuş kelime bulutu görülmektedir. Şekil 3.14.'de anahtar kelimelerin isimlerinin karakter boyutu o anahtar kelimeye sahip film sayısı ile doğru orantılı olacak şekilde büyük veya küçük gösterilmiştir. Filmler arasındaki anahtar kelime bazlı bir benzerlik hesaplaması için her film 855 bit uzunluğundaki 0 ve 1'lerden oluşan bir anahtar kelime vektörü ile ifade edilmiştir.

Filmler arasındaki benzerlik hesaplamaları (tür, yıl, anahtar kelime ve oylanma sayısı benzerlikleri) literatürde en çok kullanılan yöntem olan kosinüs benzerliğine göre hesaplanmıştır.

$$\text{sim(a, c)} = \cos(\vec{a}, \vec{c}) = \frac{\vec{a}.\vec{c}}{||\vec{a}||.||\vec{c}||} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2}\sqrt{\sum_i^n b_i^2}} \quad (3.19)$$

Denklem (3.19)'de *sim(a,c)*, *a* ve *i* filmleri arasındaki kosinüs benzerlik oranıdır. $\vec{a}$, *a* filmi için oluşturulmuş öznitelik vektörü, $\vec{c}$ ise *c* filmi için oluşturulmuş öznitelik vektörüdür. $|\vec{a}|$, *a* vektörünün $|\vec{c}|$, *c* vektörünün uzunluğunu göstermektedir.

MovieLens 100K veri setinde bulunan ratings.csv (Çizelge 3.5.) dosyası üzerinde yapılan hesaplamalar ile her filmin oylanma sayısı elde edilmiştir. Oylanma sayısı bir filmin tüm değerlendirmeler içinde toplamda kaç defa değerlendirildiğidir. Bulunan bu oylanma sayıları öncelikle [0-1] aralığına normalize edilmiştir. Daha sonra normalize edilmiş değerler arasında kosinüs benzerliği kullanılarak hesaplamalar yapılmıştır.

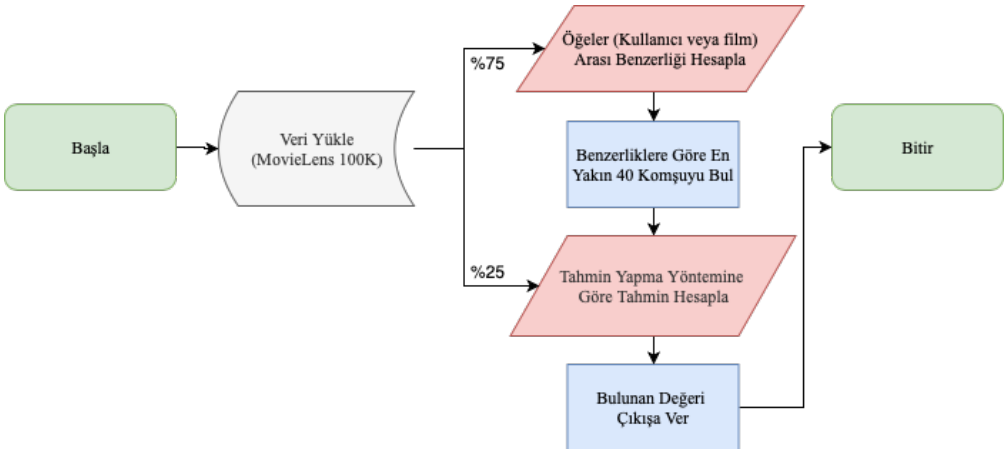
Önerilen içerik tabanlı filtreleme algoritmasında yukarıda oluşturulmuş olan öznitelik vektörleri sayesinde filmler arasındaki benzerlikler her öznitelik için ayrı ayrı hesaplanmıştır. Daha sonra bulunan bu benzerlik oranları hem tek tek hem de toplanarak içerik tabanlı filtreleme algoritmasının benzerlik aşamasında kullanılmıştır. Komşuluk seçimi için *k*-EYK algoritması kullanılmıştır. Herlocker ve ark. (2002), yaptıkları çalışmada *k*-EYK algoritmasında k değerinin 20 ile 50 arasında en iyi değerleri aldığını göstermişlerdir. Bu teze dayanarak k (en yakın komşu sayısı) değeri 40 alınmıştır. Seçilen komşuların filmler için yaptıkları değerlendirmeler benzerlik

katsayıları ile çarpılarak önerilen içerik tabanlı filtreleme algoritmasının çıktısı (tahmini) elde edilmiştir.

3.9.2. Önerilen işbirlikçi filtreleme yöntemi

İşbirlikçi filtreleme daha çok metin ile tanımlanamayan ürünlerin önerilmesinde kullanılan bir yöntemdir. Kitap, dergi, haber makaleleri vb. doküman öneri sistemlerinde içerik tabanlı filtreleme tercih edilirken; film, müzik, video vb. ürünlerin olduğu sistemlerinde daha çok işbirlikçi filtreleme tercih edilmektedir. İşbirlikçi filtreleme aradaki benzerlikleri hesaplanan öğenin türüne göre iki şekilde kullanılmaktadır. Bunlar kullanıcı tabanlı ve ürün tabanlı işbirlikçi filtrelemelerdir.

Tez kapsamında önerilen hibrit film öneri sisteminde kullanılan işbirlikçi filtreleme için ilk başta [kullanıcı (671) X film (9125)] matrisi oluşturulmuştur. Ortaya çıkan matris çok büyüt boyutlara sahip olmaktadır. Şekil 3.1.'de görülen kullanıcı ürün matrisinde ürün olarak filmler kullanılmıştır. Bu matriste satırlar her bir kullanıcının ilgili film için yapmış olduğu değerlendirmeyi göstermektedir. Sütunlar ise ilgili filmin kullanıcılar tarafından yapılmış tüm değerlendirmelerini göstermektedir. İşbirlikçi filtreleme algoritmasının akış şeması Şekil 3.15.'de görülmektedir.



Şekil 3.15. İşbirlikçi filtreleme algoritması akış şeması

Yapılan deneylerde işbirlikçi filtrelemenin ilk aşaması olan benzerlik hesaplama aşamasında hem kullanıcılar hem de filmler arasındaki benzerlikler

hesaplanmıştır. Film tabanlı işbirlikçi filtrelemede her film 671 tane değerlendirmeden oluşan bir sayı dizisi ile ifade edilmiştir. Tüm kullanıcıların değerlendirmelerinden oluşan bu sayı dizisi bir filmin vektörünü oluşturmaktadır. Kullanıcı tabanlı işbirlikçi filtrelemede ise her kullanıcı 9125 tane değerlendirmeden oluşan bir sayı dizisi ile ifade edilmiştir. İlgili kullanıcı tarafından yapılmış tüm değerlendirmelerden oluşan bu sayı dizisi bir kullanıcının vektörünü oluşturmaktadır. Daha önceki bölümlerde anlatılmış olan benzerlik bulma yöntemleri yapılan deneyler ile karşılaştırmalı olarak değerlendirilmiş olup en düşük hata oranlarına sahip işbirlikçi filtrelemenin pearson korelasyonu benzerlik hesaplama yöntemi ile yapılan deneyden elde edildiği tespit edilmiştir.

İkinci aşama olan komşuluk seçimi yönteminde k-EYK algoritması kullanılmıştır. Herlocker ve ark. (2002), yaptıkları çalışmada k-EYK algoritmasında k değerinin 20 ile 50 arasında en iyi değerleri aldığını gözlemlemişlerdir. Bu teze dayanarak k (en yakın komşu sayısı) 40 olarak alınmıştır.

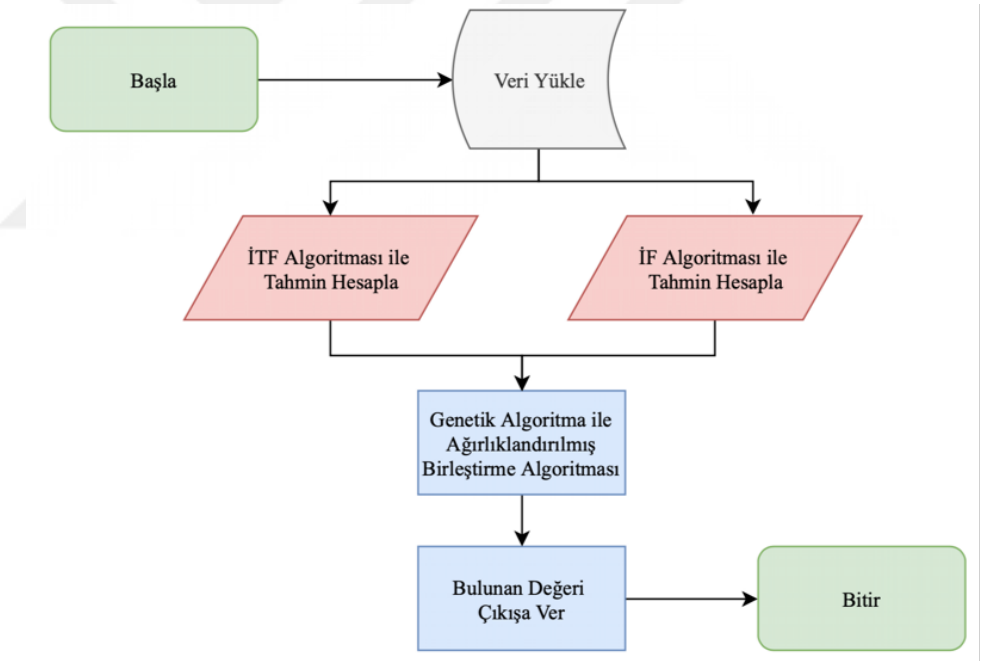
Her kullanıcının oylama skalası farklı olabilmektedir. Kimi kullanıcılar düşük puan verme eğiliminde olabilirken kimileri de yüksek puan verme eğiliminde olabilmektedir. Bu nedenledir ki kişilerin sadece benzerlik oranlarına göre yapılan BO tahmin yapma yöntemi yeterli başarıyı gösterememiştir. Yapılan deneysel çalışmalar neticesinde film tabanlı işbirlikçi filtrelemelerde AAO tahmin yapma yöntemi, kullanıcı tabanlı işbirlikçi filtrelemelerde SAAO tahmin yapma yöntemi ile yapılan tahminlerin daha doğru sonuçlar verdiği gözlemlenmiştir.

İşbirlikçi filtrelemede algoritmasında MovieLens 100K veri setinin %75'i algoritmayı eğitmek için eğitim verisi olarak kullanılmıştır. Veri setinin geri kalan %25'lik kısmı yapılacak doğruluk testleri (OMH ve HKOK hata oranları tespiti) için kullanılmıştır.

3.9.3. Önerilen hibrit filtreleme yöntemi

Öneri sistemlerinde gerek geleneksel filtrelemelerin (işbirlikçi filtreleme, içerik tabanlı filtreleme) başarı oranlarını arttırmak gerekse eksikliklerini gidermeleri için işbirlikçi filtreleme ve içerik tabanlı filtreleme yöntemleri sıklıkla bir arada kullanılmaktadır (Adomavicius ve Tuzhilin, 2005).

Geleneksel yöntemlerin bir araya getirilmelerinin farklı yolları Bölüm 3.1.3.'de anlatılmıştır. Bu tez kapsamında önerilen hibrit yöntemde, Adomavicius ve Tuzhilin'nin (2005), yaptıkları çalışmada anlatılan yöntemlerden biri olan "İşbirlikçi ve içerik tabanlı yöntemleri veri seti üzerinde ayrı ayrı uygulamak ve tahminlerini birleştirmek" yöntemi kullanılmıştır.



Şekil 3.16. Hibrit film öneri algoritması akış şeması

Önerilen hibrit yönteme ait akış şeması Şekil 3.16.'da görülmektedir. Önerilen hibrit yöntemde içerik tabanlı filtrelemeden ve işbirlikçi filtrelemeden gelen tahminler bir birleştirme algoritmasına tabi tutularak bir araya getirilmiştir. İki farklı birleştirme algoritması ile iki farklı senaryo düşünülmüştür. İki algoritmanın bir araya getirilmelerinde kullanılan birleştirme algoritmalarında kullanılan katsayılar GA ile

optimize edilmiş ve aritmetik ortalamaya göre daha iyi sonuçlar elde ettiği yapılan deneyler ile gözlemlenmiştir.

Bazı tahminlerde içerik tabanlı bazı tahminlerde ise işbirlikçi filtrelemenin daha başarılı tahmin yapacağı varsayılmıştır. Çünkü kullanıcının sistem içindeki davranışları farklı olmaktadır. Değerlendirme aşamasında her kullanıcı farklı bakış açılarına sahiptir. Bu durum her algoritmanın her kullanıcı için en iyi performansı göstermemesine neden olmaktadır. Bunun en iyi ispatı hibrit yöntemlerin geleneksel filtreleme kullanılan yöntemlerden daha başarılı olmasıdır. Algoritmaların ürettikleri tahminlerin kıyaslanarak farklı oranlarda bir araya getirilmelerinin daha doğru sonuçlar vereceği düşünülmüştür. Böylelikle kullanıcılar için üretilen tahminlerde hangi algoritmanın hangi şartlar altında daha doğru sonuçlar verdiği anlaşılabilir. İçerik tabanlı filtreleme ile işbirlikçi filtreleme algoritmalarının ürettikleri tahminleri bir araya getirmek için iki farklı birleştirme algoritması önerilmiştir.

Birleştirme-1 Algoritması adı verilen ilk birleştirme algoritması ile önerilen hibrit algoritmaya girdi olarak alınan iki farklı tahminin karşılaştırması planlanmıştır. Algoritmada girdi kabul edilen tahminlerin büyüklüklerinin doğru tahmin üzerinde etkili olacağı varsayılarak büyüklüklerine göre farklı katsayılar kullanılmıştır. Kullanılan katsayılar OMH hata oranlarını en küçük yapacak şekilde tasarlanmış genetik algoritma ile optimize edilmiştir. Tez kapsamında önerilen Birleştirme-1 Algoritmasına ait sözde kodlar Şekil 3.17.'de görülmektedir.

$İF_{tahmin} > İTF_{tahmin}$ ise

$HF_{tahmin} = (x * İF_{tahmin} + y * İTF_{tahmin}) / (x + y)$

değil ise

$HF_{tahmin} = (y * İF_{tahmin} + x * İTF_{tahmin}) / (x + y)$

Şekil 3.17. Birleştirme-1 algoritması

Birleştirme-2 Algoritmasında ek olarak iki algoritmadan gelen tahminler arasındaki uzaklığın belli bir değer üzerinde olmasının doğru tahmin üzerindeki

etkileri araştırılmıştır. Tez kapsamında önerilen Birleştirme-2 Algoritmasına ait sözde kodlar Şekil 3.18.'de görülmektedir.

$İF_{tahmin} >= İTF_{tahmin}$ ise
    $| İF_{tahmin} - İTF_{tahmin} | > k$ ise
        $HF_{tahmin} = (x * İF_{tahmin} + y * İTF_{tahmin}) / (x + y)$
    değil ise
        $HF_{tahmin} = (z * İF_{tahmin} + d * İTF_{tahmin}) / (z + d)$
değil ise
    $| İF_{tahmin} - İTF_{tahmin} | > m$ ise
        $HF_{tahmin} = (f * İF_{tahmin} + g * İTF_{tahmin}) / (f + g)$
    değil ise
        $HF_{tahmin} = (j * İF_{tahmin} + n * İTF_{tahmin}) / (j + n)$

Şekil 3.18. Birleştirme-2 algoritması

Şekil 3.18.'de görülen Birleştirme-1 ve Birleştirme-2 Algoritmalarına ait sözde kod içinde yer alan $İF_{tahmin}$ işbirlikçi filtreleme algoritması, $İTF_{tahmin}$ ise içerik tabanlı filtreleme algoritması ile hesaplanan tahmini değerlendirmelerdir. Sözde kodlarda yer alan "*k, x, y, m, f, g, j, n"* katsayıları OMH değerlerini minimum yapacak şekilde genetik algoritma ile optimize edilmiştir.

İki algoritmanın sahip oldukları düşük OMH değerlerine dayanarak yapacakları tahminlerin farklarının ikiden fazla olmayacağı düşünüldüğünden genetik algoritma değişkenlerinden k ve m değerleri için [0-2] aralığında değerler kümesi, x, y, f, g, j, n değerleri için de [0-1] aralığında değerler kümesi tanımlanmıştır. Genetik algoritma için nesil sayısı bazı deneylerde 10 bazı deneylerde 20 olarak alınmıştır. Çaprazlama olasılığı 0.8, mutasyon oranı 0.1 ve popülasyon boyutu 20 olarak alınmıştır.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

4.1. Yapılan Deneylerin Performansları

Bu bölümde önerilen yöntemlerin parametreleri arasında karşılaştırmalı değerlendirmeler yapılmıştır. Bu bölüm sonunda önerilen hibrit film öneri sistemi literatürdeki başka yöntemler ile bilinen değerlendirme metrikleri üzerinden karşılaştırtılmıştır.

Tüm deneyler Spyder Derleyicisi ile Python dilinde yazılmış kodlarla MovieLens 100K veri seti kullanılarak gerçekleştirilmiştir. Deneylerde kullanılan bilgisayar MacOS 10.15 işletim sistemi, Intel Core i7 8750H işlemci, 4,1 GHz işlemci hızı ve 16GB DDR4 Ram özelliklerine sahip bir Macbook Pro'dur.

4.1.1. Önerilen içerik tabanlı filtreleme yöntemi deneyleri

Yapılan deneylerde MovieLens 100K veri seti içinde bulunan anahtar kelimeler, türler, yıl bilgileri ile yüz bin değerlendirme içerisinden hesaplanan toplam oylanma sayıları benzerlik bulma aşamasında kullanılmıştır. Filmler arasındaki benzerlikler hesaplanırken her bir deney için sadece tek bir öznitelik kullanılarak hesaplama yapılmıştır.

Örneğin tür özniteliği kullanılarak yapılan benzerlik bulma işleminde ilk olarak her film için Çizelge 3.9.'da gösterildiği gibi bir tür vektörü oluşturulmuştur. Daha sona Denklem (3.19)'da gösterilen kosinüs benzerlik hesaplama denklemi sayesinde iki film vektörü arasındaki benzerlikler bulunmuştur. Yapılan bu hesaplamalar sonucu [film X film] benzerlik matrisi elde edilmiştir. Çizelge 4.1.'de tür özniteliği için oluşturulmuş olan benzerlik matrisinin [10 X 10]'luk bir kesiti görülmektedir. Verilen çizelge oluşturulurken bir filmin kendisi ile olan benzerlikleri daha sonra yapılacak tahmin hesaplama aşamasında ön plana çıkmamaları için hesaplanmamış, 0 (Sıfır)

değeri olarak alınmıştır. Benzerlik oranı 1 olan filmler birebir aynı vektöre sahip oldukları için birbirlerine en yakın komşuluğa sahiptirler.

Çizelge 4.1. Tür özniteliği kullanılarak oluşturulmuş benzerlik matrisinden bir kesit

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	0.3535	0.7071	0	0.5	0.5	0.577	0	0	0.75
1	0.3535	0	0	0	0.7071	0.7071	0	0	0	0
2	0.7071	0	0	0	0	0	0.4082	0	0	0.7071
3	0	0	0	0	0	0	0	0	1	0.5
4	0.5	0.7071	0	0	0	1	0	0	0	0
5	0.5	0.7071	0	0	1	0	0	0	0	0
6	0.5773	0	0.4082	0	0	0	0	0.4082	0	0.5773
7	0	0	0	0	0	0	0.4082	0	0	0
8	0	0	0	1	0	0	0	0	0	0.5
9	0.75	0	0.7071	0.5	0	0	0.5773	0	0.5	0

Diğer öznitelikler için de aynı işlemler yapılıp benzerlik matrisleri elde edilmiştir. Hesaplanan bu benzerlik oranları kullanılarak *k*-EYK algoritması sayesinde o filme en çok benzeyen 40 film (en yakın komşular) bulunmuştur. Daha sonra bu benzer filmlere kullanıcılar tarafından yapılan değerlendirmeler benzerlik oranları ile çarpılıp toplanmıştır. Elde edilen toplama işlemi sonucu toplam benzerlik oranına bölünerek bir tahmin elde edilmiştir. Her öznitelik için ayrı ayrı oluşturulmuş içerik tabanlı filtreleme algoritmaları ile elde edilen bu tahminlere ait HKOK ve OMH hata oranları Çizelge 4.2.'de görünmektedir.

Çizelge 4.2. Öznitelik hata oranları

Öznitelik	**HKOK**	**OMH**
Tür	*0.9519**	*0.7399**
Yıl	0.9590	0.7450
Oylanma Sayısı	0.9729	0.7530
Anahtar Kelime	1.0611	0.8465

Çizelge 4.2.'de görüldüğü üzere hem OMH hem de HKOK hata oranları için en düşük değerler "Tür" özniteliğine aittir. En yüksek hata oranları "Anahtar Kelimeler" üzerinden hesaplanan benzerliklere dayalı içerik tabanlı filtreleme ile yapılan tahminler sonucunda elde edilmiştir. Benzerlik aşamasında veri setinde bulunan değerlendirmelerden elde edilen oylanma sayıları özniteliği kullanılarak oluşturulmuş içerik tabanlı filtreleme algoritması ile gerçekleştirilen deneyin hata oranlarına bakıldığında anahtar kelimeler kullanılan içerik tabanlı filtreleme algoritmasından daha başarılı olduğu görülmektedir.

Bu özniteliklerin ikili, üçlü ve dörtlü kombinasyonları benzerlik matrisi hesaplama aşamasında bir araya getirilmiştir. Örneğin iki film arasındaki benzerliği bulmak için ikisinin arasındaki "Tür" ve "Yıl" benzerlikleri ayrı ayrı kosinüs benzerlik bulma yöntemine göre hesaplanmış ve bulunan oranlar toplanmıştır. Elde edilen toplam, bu iki film arasındaki yeni benzerlik oranıdır. Yapılan toplama işleminde eşit katsayılar kullanılmıştır. Bulunan bu yeni benzerlik oranlarına göre benzerlik matrisi oluşturulmuştur. Hesaplanan bu benzerlik oranları kullanılarak *k*-EYK algoritması ile o filme en çok benzeyen 40 film bulunmuştur. Daha sonra bu benzer filmlere kullanıcılar tarafından yapılan değerlendirmeler benzerlik oranları ile çarpılıp toplanmıştır. Elde edilen toplama işlemi sonucu benzerlik oranının toplamına bölünerek bir tahmin elde edilmiştir. Ayrı ayrı yapılan deneyler ile elde edilmiş OMH ve HKOK hata oranları Çizelge 4.3.'de görünmektedir. Deneyler isimlendirilirken kullanılan İTF, içerik tabanlı filtreleme algoritmasını ifade etmektedir. Kısaltmaların yanında bulunan sayılar deney sırasına göre belirlenmiştir.

Çizelge 4.3. Öznitelik kombinasyonları hata oranları

Algoritma	**Öznitelik Kombinasyonu**	**HKOK**	**OMH**
İTF-1	Tür + Yıl	0.9387	0.7297
İTF-2	Tür + Oylanma Sayısı	0.9407	0.7327
İTF-3	Tür + Anahtar Kelime	0.9574	0.7434
İTF-4	Yıl + Oylanma Sayısı	0.9456	0.7380
İTF-5	Yıl + Anahtar Kelime	0.9574	0.7434
İTF-6	Oylanma Sayısı + Anahtar Kelime	0.9697	0.7499
İTF-7	Tür + Yıl + Oylanma Sayısı	0.9302	0.7251

Çizelge 4.3. (**devam**)

Algoritma	Öznitelik Kombinasyonu	HKOK	OMH
İTF-8	Tür + Yıl + Anahtar Kelime	0.9376	0.7286
İTF-9	Tür + Oylanma Sayısı + Anahtar Kelime	0.9393	0.7313
İTF-10	Yıl + Oylanma Sayısı + Anahtar Kelime	0.9442	0.7367
İTF-11	Tür + Yıl + Anahtar Kelime + Oylanma Sayısı	*0.9292**	*0.7242**

Çizelge 4.3.'de görülen deney sonuçlarına göre, filmler arasındaki benzerlikler hesaplanırken oylanma sayısının dikkate alınmadığı "Tür + Yıl + Anahtar Kelime" öznitelikleri için bulunan benzerlik oranları toplanarak hesaplanmış olan yeni benzerlik oranının benzerlik bulma aşamasında kullanıldığı İTF-8 algoritması ile elde edilmiş olan HKOK ve OMH hata oranları oylanma sayısı eklenerek hesaplama işlemleri yapılan İTF-11 algoritmasından daha yüksek çıkmıştır. Bu sonuçlardan bir filmin toplamda kaç kişi tarafından değerlendirildiği bilgisi olan oylanma sayısının içerik tabanlı öneri sistemlerinde dikkate alınması gerektiği anlaşılmaktadır.

Deney sonuçları dikkate alınarak yapılacak hibrit filtreleme deneylerinde kullanılacak içerik tabanlı filtreleme algoritması, benzerlik matrisi bu dört özniteliğin bir araya getirilmesi ile oluşturulmuş olan İTF-11 algoritması olacaktır.

4.1.2. Önerilen işbirlikçi filtreleme yöntemi deneyleri

Önerilen hibrit öneri sisteminin diğer bir bileşeni olan işbirlikçi filtreleme algoritması hem benzerlik hesaplama aşamasında hem de tahmin yapma aşamasında Bölüm 3.1.1.1. ve Bölüm 3.1.1.3.'de anlatılmış olan yöntemlerin değişik kombinasyonları ile yapılan deneysel karşılaştırmalar sonucunda oluşturulmuştur. Bu deneylerin hem kullanıcı tabanlı hem de film (ürün) tabanlı işbirlikçi filtreleme algoritmaları için ayrı ayrı performans değerlendirmesi yapılmıştır.

Komşuluk seçiminde *k*-EYK algoritması kullanılarak oluşturulan film tabanlı ve kullanıcı tabanlı işbirlikçi filtreleme algoritmaları ile farklı tahmin ve benzerlik hesaplama yöntemleri kombinasyonları kullanılarak yapılmış olan deneylerin OMH ve HKOK hata oranları Çizelge 4.4. ve 4.5.'de gösterilmektedir. Deneyler

isimlendirilirken kullanılan KİF kısaltması kullanıcı tabanlı işbirlikçi filtreleme algoritmasını, FİF kısaltması ise film tabanlı işbirlikçi filtreleme algoritmasını ifade etmektedir. Kısaltmaların yanında bulunan sayılar deney sırasına göre belirlenmiştir.

Çizelge 4.4. Film tabanlı işbirlikçi filtreleme deney sonuçları

Algoritma	**Tahmin Yapma Yöntemleri**	**Benzerlik Yöntemleri**	**HKOK**	**OMH**
FİF-1	BO	KB	0.9969	0.7778
FİF-2	BO	PK	0.9867	0.7670
FİF-3	BO	OKF	0.9414	0.7279
FİF-4	AO	KB	0.9969	0.7778
FİF-5	AO	PK	0.9900	0.7703
FİF-6	AO	OKF	0.9393	0.7262
FİF-7	AAO	KB	0.9284	0.7143
FİF-8	AAO	PK	0.9293	0.7108
FİF-9	AAO	OKF	*0.9169**	*0.7047**
FİF-10	SAAO	KB	0.9331	0.7160
FİF-11	SAAO	PK	0.9321	0.7125
FİF-12	SAAO	OKF	0.9226	0.7073
		Ortalama	**0.9511**	**0.7343**

Çizelge 4.5. Kullanıcı tabanlı işbirlikçi filtreleme deney sonuçları

Algoritma	**Tahmin Yapma Yöntemleri**	**Benzerlik Yöntemleri**	**HKOK**	**OMH**
KİF-1	BA	KB	0.9953	0.7712
KİF-2	BA	PK	0.9987	0.7761
KİF-3	BA	OKF	0.9826	0.7600
KİF-4	AO	KB	0.9947	0.7705
KİF-5	AO	PK	1.0006	0.7761
KİF-6	AO	OKF	0.9706	0.7483
KİF-7	AAO	KB	0.9226	0.7088
KİF-8	AAO	PK	0.9241	0.7055
KİF-9	AAO	OKF	0.9193	0.7048
KİF-10	SAAO	KB	0.9206	0.7034
KİF-11	SAAO	PK	0.9207	*0.6986**
KİF-12	SAAO	OKF	*0.9187**	0.7003
		Ortalama	**0.9557**	**0.7353**

Benzerlik ve tahmin hesaplama yöntemleri aynı olacak şekilde 12 KİF ve 12 FİF deneylerinin sonuçları kullanılarak yapılan 12 karşılaştırmanın 7 tanesinde kullanıcı tabanlı işbirlikçi filtreleme daha başarılı sonuçlar verdiği Çizelge 4.4. ve 4.5.'de görülmektedir. Film tabanlı ve kullanıcı tabanlı işbirlikçi filtrelemelerin hata oranlarının ortalamalarına bakıldığında HKOK hata oranlarına göre kullanıcı tabanlı işbirlikçi filtreleme algoritmasının, OMH hata oranlarına göre ise film tabanlı işbirlikçi filtreleme algoritmasının daha başarılı olduğu görülmektedir. Standart sapmalı ayarlanabilir ağırlıklandırılmış ortalama tahmin yapma yöntemiyle yapılan deneylerin hepsinde kullanıcı tabanlı işbirlikçi filtreleme daha başarılı sonuçlar vermiştir. Genel değerlendirmede OMH hata oranlarına göre en düşük değere sahip deney kullanıcı tabanlı bir işbirlikçi filtreleme algoritması olan *KİF-11* algoritması iken HKOK hata oranlarına göre en düşük değere sahip deney film tabanlı bir işbirlikçi filtreleme algoritması olan *FİF-9* algoritması ile yapılan deney sonucunda elde edilmiştir.

Çizelge 4.4. ve 4.5.'de görülen deney sonuçları diğer parametreler aynı kalacak şekilde sadece benzerlik hesaplama yöntemlerine göre değerlendirildiğinde, 4 farklı tahmin yapma yöntemine göre de OKF benzerlik hesaplama yönteminin diğer benzerlik hesaplama yöntemlerinden daha başarılı olduğu görülmektedir. En kötü deney sonuçları KB benzerlik hesaplama yöntemine aittir. OKF benzerlik hesaplama yöntemi kullanıcı ve film tabanlı işbirlikçi filtrelemelere göre değerlendirildiğinde genel olarak film tabanlı sistemlerde daha başarılı sonuçlar elde edilmiştir. Sadece SAAO tahmin yapma yöntemi ve OKF benzerlik hesaplama yönteminin bir arada kullanıldığı *KİF-12* kullanıcı tabanlı işbirlikçi filtreleme algoritmasının, *FİF-12* film tabanlı işbirlikçi filtreleme algoritmasından daha başarılı olduğu görülmektedir.

Çizelge 4.4. ve 4.5.'de görülen deney sonuçları diğer parametreler aynı kalacak şekilde sadece tahmin yapma yöntemlerine göre değerlendirildiğinde, hata oranlarının aritmetik ortalamasına göre en başarılı yöntemin AAO yöntemi en başarısız yöntemin de BO yöntemi olduğu görülmektedir. AAO yönteminde en düşük hata oranları OKF benzerlik bulma yöntemi ile yapılan film tabanlı işbirlikçi filtreleme algoritması olan *FİF-9* ile elde edilmiştir. Bu sonuç bize bulunan komşuluklarda öğelerin kendi ortalamalarının doğruluk üzerinde önemli bir etkisi olduğunu kanıtlamaktadır. Çünkü

her kullanıcının değerlendirme skalası farklıdır. Kimi kullanıcılar çok zor beğenirken kimi kullanıcılarda sürekli yüksek değerlendirme yapma eğilimindedirler. Yapılan 24 deney içinden en kötü sonuç AO tahmin yapma yöntemi ve PK benzerlik bulma yöntemi kullanılan kullanıcı tabanlı bir işbirlikçi filtreleme algoritması olan KİF-5 deneyi ile elde edilmiştir.

Çizelge 4.4. ve 4.5.'de görülen deney sonuçlarına göre en düşük hata oranlarına sahip olan işbirlikçi filtreleme algoritmasını baz alarak hibrit bir film öneri algoritması oluşturmadan önce diğer deneylerden de bazılarının kullanıp karşılaştırma yapılmasına karar verilmiştir. Hibrit film öneri algoritması oluşturma aşamasındaki deneylerde kullanılan işbirlikçi filtreleme algoritmaları şunlardır:

- KB benzerlik bulma yöntemi kullanılan en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-10 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-7 isimli algoritma seçilmiştir.
- PK benzerlik bulma yöntemi kullanılan en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-11 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-8 isimli algoritma seçilmiştir.
- OKF benzerlik bulma yöntemi kullanılan en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-12 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-9 isimli algoritma seçilmiştir.
- BO tahmin yapma yöntemi için en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-3 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-3 isimli algoritma seçilmiştir.
- AO tahmin yapma yöntemi için en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-6 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-6 isimli algoritma seçilmiştir.
- AAO tahmin yapma yöntemi için en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-9 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-9 isimli algoritma seçilmiştir.

- SAAO tahmin yapma yöntemi için en iyi kullanıcı tabanlı işbirlikçi filtreleme algoritması olan KİF-12 isimli algoritma ve en iyi film tabanlı işbirlikçi filtreleme algoritması olan FİF-12 isimli algoritma seçilmiştir.

Bir önceki bölümde bulunan en iyi içerik tabanlı filtreleme algoritması (İTF-11) ile yukarıdaki kriterlere göre seçilen 12 farklı işbirlikçi filtreleme algoritmalarının yaptıkları tahminler belli oranlarda bir araya getirilerek en iyi sonuçlar bulunmaya çalışılmıştır.

4.1.3. Önerilen hibrit filtreleme yöntemi için yapılan deneyler

Tez sonunda önerilen hibrit film öneri algoritmasında kullanılan işbirlikçi filtreleme ve içerik tabanlı filtrelemeye karar vermek için çeşitli deneyler yapılmıştır. İlk Bölüm 4.1.1.'de anlatılmış olan içerik tabanlı filtreleme yöntemi ve Bölüm 4.1.2.'de anlatılmış işbirlikçi filtreleme yöntemleri eşit oranlarda ile bir araya getirilmiştir. Ortaya çıkan bu eşit katsayılı hibrit yöntemler için yapılan deneylerin sonuçları Çizelge 4.6.'da görülmektedir.

Çizelge 4.6. Eşit katsayı kullanılan hibrit filtreleme yöntemleri

Algoritma	**İçerik Tabanlı Filtreleme**	**İşbirlikçi Filtreleme**	**HKOK**	**OMH**
HF-1	İTF-11	KİF-3 (OKF)	0.9005	0.7026
HF-2	İTF-11	FİF-3 (OKF)	0.9239	0.7173
HF-3	İTF-11	KİF-6 (OKF)	0.8967	0.6980
HF-4	İTF-11	FİF-6 (OKF)	0.9226	0.7163
HF-5	İTF-11	FİF-7 (KB)	0.8919	0.6916
HF-6	İTF-11	FİF-8 (PK)	0.8890	0.6891
HF-7	İTF-11	KİF-9 (OKF)	0.8957	0.6923
HF-8	İTF-11	FİF-9 (KB)	*0.8886**	0.6891
HF-9	İTF-11	KİF-10 (KB)	0.8958	0.6920
HF-10	İTF-11	KİF-11 (PK)	0.8905	*0.6856**
HF-11	İTF-11	KİF-12 (OKF)	0.8938	0.6898
HF-12	İTF-11	FİF-12 (OKF)	0.8900	0.6897
		Ortalama	**0.8982**	**0.6961**

Hibrit deneyler HF (hibrit filtreleme) ve deney numarası kullanılarak isimlendirilmiştir. Çizelge 4.6.'da işbirlikçi filtreleme sütununda parantez içerisinde işbirlikçi filtreleme algoritmasında kullanılan benzerlik bulma yöntemlerini (KB-PK-OKF) gösterilmektedir.

Yapılan deneylerde tüm özniteliklerin (anahtar kelimeler, türler, yıl ve oylanma sayısı) birlikte kullanıldığı en başarılı içerik tabanlı filtreleme algoritması olan İTF-11 algoritması ve Bölüm 4.1.2.'de seçilen 12 farklı işbirlikçi filtreleme algoritması kullanılmıştır. Tüm deneylerde aynı içerik tabanlı filtreleme algoritması (İTF-11) kullanıldığı için yapılan değerlendirmeler sadece işbirlikçi filtreleme faktörü üzerinden yapılmıştır.

Hibrit filtreleme için yapılmış deneylerde kullanılan 12 farklı işbirlikçi filtreleme algoritmasından 6 tanesi kullanıcı tabanlı ve 6 tanesi film tabanlı işbirlikçi filtrelemedir. Çizelge 4.4. ve 4.5.'de görüldüğü gibi yapılan deneylerde kullanılan işbirlikçi filtreleme algoritmalarının büyük çoğunluğu benzerlik hesaplama aşamasında OKF benzerlik bulma yöntemini kullanmıştır. Bu hibrit filtreleme deneylerinde kullanılan işbirlikçi filtreleme algoritmalarında genel olarak film tabanlı yaklaşımlar kullanıcı tabanlı yaklaşımlardan daha başarılı olmuştur. Aynı doğrultuda HF-11 ve HF-12 algoritmalarında kullanılan işbirlikçi filtreleme yöntemlerinde kullanıcı tabanlı filtreleme bireysel olarak daha başarılı olurken içerik tabanlı filtreleme ile bir araya getirildiklerinde film tabanlı işbirlikçi filtrelemenin kullanıldığı HF-12 isimli algoritmanın daha başarılı olduğu görülmüştür.

Tez kapsamında önerilen hibrit film öneri sistemi için genetik algoritma ile yapılan yoğun matematiksel işlemlerin gerektirdiği yüksek maliyetler (işlem süresi) nedeni ile Çizelge 4.6.'da yapılmış deneylerden en başarılı dört tanesi genetik algoritma ile yapılan optimizasyon işlemleri için seçilmiştir. Hem OMH hem de HKOK hata oranlarında en düşük değerlere sahip hibrit yöntemler HF-6, HF-8, HF-12 ve HF-10 isimli algoritmalardır. En düşük hata oranından en yüksek olanına göre sıralama yapılırsa HKOK hata oranlarına göre HF-6, HF-8, HF-12, HF-10 sırası oluşurken, OMH hata oranlarına göre HF-10, HF-6, HF-8, HF-12 sıralaması ortaya

çıkmaktadır. Seçilen en başarılı hibrit filtreleme yöntemlerden sadece HF-10 algoritmasında kullanılan işbirlikçi filtreleme algoritması kullanıcı tabanlıdır. Yapılan deney sonuçlarına göre çoğunlukla film tabanlı işbirlikçi filtreleme algoritması kullanılan hibrit yöntemlerin daha başarılı olduğu söylenebilir.

4.2. Genetik Algoritma ile Yapılan Optimizasyon Deneyleri

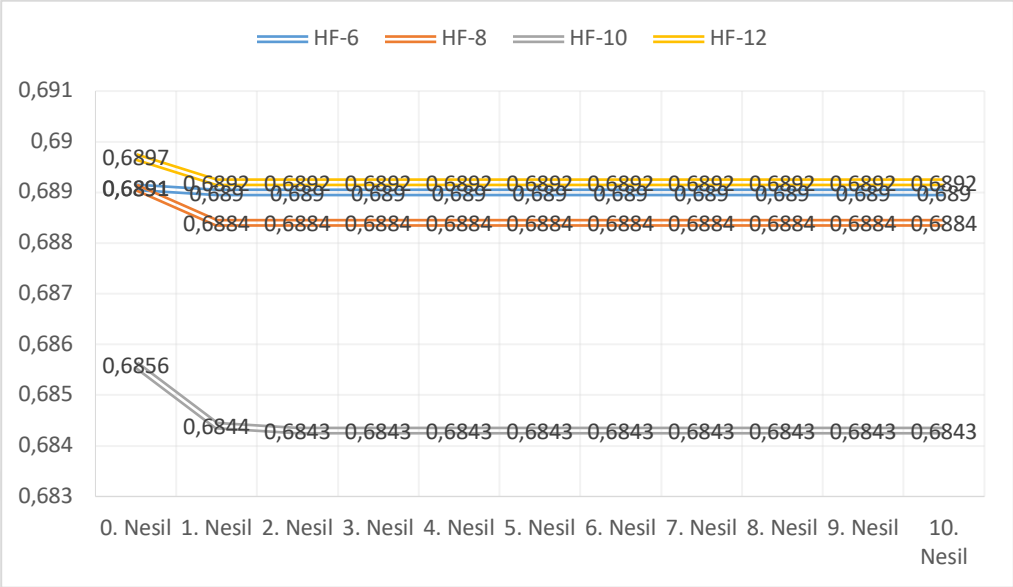
Bölüm 4.1.3.'de seçilen en başarılı HF-6, HF-8, HF-10, HF-12 isimli hibrit film öneri algoritmaları oluşturulurken içerik tabanlı filtreleme algoritması ve işbirlikçi filtreleme algoritmalarının ürettikleri tahminler aritmetik (eşit oranda) olarak bir araya getirilmiştir. İki tahmin toplanıp ikiye bölünerek hibrit yöntemin tahmini elde edilmiştir. Bu bölümde önerilen hibrit film öneri yönteminde girdi olarak kabul edilen bu iki tahminin hangi oranlarda bir araya geldiğinde daha iyi performans göstereceği yapılan deneyler ile gösterilmiştir.

Bölüm 4.1.'de yapılan tüm deneyler göz önüne alındığında genel olarak OMH ve HKOK hata oranlarının doğru orantılı olduğu görülmektedir. Bu nedenle Genetik algoritma için kullanılacak uygunluk fonksiyonu olarak bu iki hata oranından biri olan OMH hata oranı kullanılmaya karar verilmiştir. Hibrit yöntemler için OMH hata oranları minimize edilmeye çalışılmıştır. Genetik algoritmada kullanılan parametreler ve nasıl uygulandığı Bölüm 3.7.'de ve Bölüm 3.9.3.'de detaylı bir şekilde anlatılmıştır.

$$HF_{tahmin} = \frac{x*\text{İ}F_{tahmin} + y*\text{İ}TF_{tahmin}}{x+y} \quad (4.1)$$

Önerilen hibrit film öneri sisteminin aktif kullanıcının henüz değerlendirmediği filmler için yapacağı tahmini (HF_{tahmin}) hesaplamak için kullanılan Denklem (4.1)'de yer alan $İF_{tahmin}$ işbirlikçi filtreleme algoritması ile elde edilen tahmini ve $\text{İTF}_{\text{tahmin}}$ içerik tabanlı filtreleme algoritması ile elde edilen tahmini ifade etmektedir. Daha önce önerin eşit oranlı hibrit algoritmalarda (Çizelge 4.6) x ve y değişkenleri eşit alınmıştır. Hibrit film öneri algoritmasının OMH hata oranını minimize eden en uygun x ve y değişkenleri Genetik algoritma ile optimize edilmiştir.

Şekil 4.1.'de seçilmiş olan HF-6, HF-8, HF-10, HF-12 hibrit algoritmalarının OMH hata oranlarının genetik algoritma ile yapılan optimizasyon işlemlerinden sonraki yeni değerleri görülmektedir. Şekilden anlaşılacağı ikinci nesil itibari ile en uygun değerler bulunmuştur. Bu yüzden nesil sayısı için 10 değeri yeterli görülmüştür. Yapılan deneyler sonucunda her dört algoritma için az da olsa başarı sağlanmıştır. Optimize edilmiş x ve y değişkenleri sayesinde daha düşük OMH hata oranları elde edilmiştir.



Şekil 4.1. Genetik algoritma optimizasyonu ile elde edilen OMH değişimi

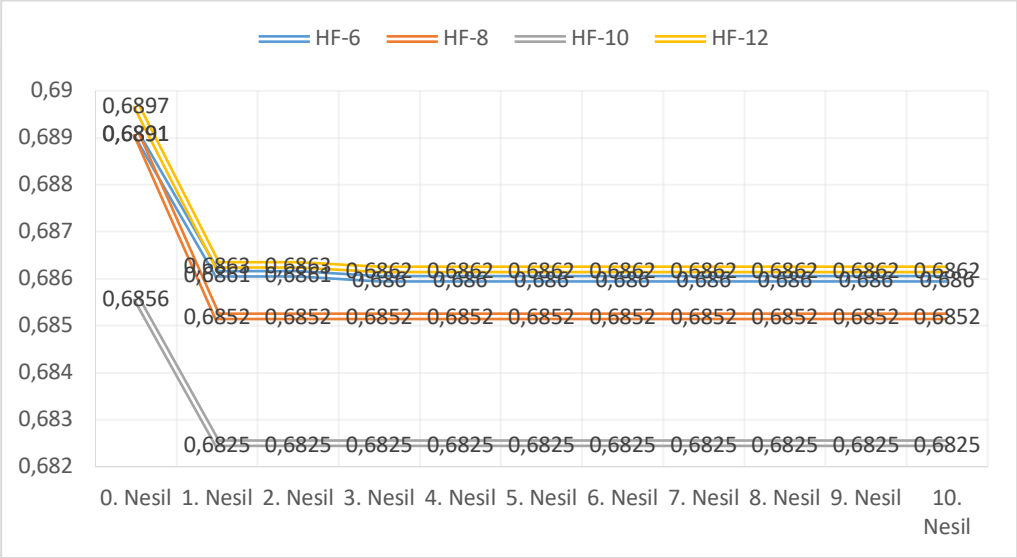
Çizelge 4.7.'de daha önceden seçilmiş olan HF-6, HF-8, HF-10, HF-12 algoritmaları için birleştirme aşamasında eşit katsayı kullanılarak bulunan OMH değerleri ve genetik algoritma ile yapılan katsayı optimizasyon işlemlerinden sonra bulunan OMH değerleri gösterilmektedir. En yüksek başarı oranının yaklaşık %0.18 ile HF-10 algoritmasında, en düşük başarı oranının ise yaklaşık %0.03 ile HF-6 algoritmasında olduğu görülmektedir.

Çizelge 4.7. Genetik algoritma ile optimize edilmiş hibrit yaklaşımlar için OMH hata oranları

Hibrit Algoritma	OMH	GA-OMH	Başarı Oranı %
HF-6	0.6891	0.6890	0.03
HF-8	0.6891	0.6884	0.08
HF-10	0.6856	*0.6843**	*0.18**
HF-12	0.6897	0.6892	0.07

Tez kapsamında işbirlikçi filtreleme ve içerik tabanlı filtreleme algoritmalarının Denklem (4.1) kullanılarak bir araya getirilmelerinde tatmin edici başarı oranları elde edilmediği için farklı birleştirme yöntemleri geliştirilmiştir.

Birleştirme-1 Algoritmasına ait sözde kod Şekil 3.17.'de verilmiştir. Bu birleştirme algoritması sayesinde oluşturulmuş hibrit film öneri algoritmalarına ait OMH hata oranları genetik algoritma ile yapılan optimizasyon deneylerine göre Şekil 4.2.'de görülmektedir.



Şekil 4.2. Genetik algoritma & birleştirme-1 algoritması için OMH değişimi

Daha önce seçilmiş olan dört hibrit algoritmanın (HF-6, HF-8, HF-10, HF-12) bileşenleri ($İTF_{tahmin}$, $İF_{tahmin}$) Birleştirme-1 Algoritması kullanılarak bir araya getirilmişlerdir. Birleştirme-1 Algoritmasında bulunan katsayılar genetik algoritma ile OMH hata oranını en küçük yapacak şekilde optimize edilmiştir. Yapılan

optimizasyon işlemine ait OMH hata oranları değişimi Şekil 4.2.'de görülmektedir. Genetik algoritma 10 nesil boyunca çalıştırılmıştır. İkinci nesil itibari ile OMH değerlerini en düşük yapacak en uygun x ve y değerleri bulunmuştur. Şekil 4.2.'de Birleştirme-1 Algoritması ve GA'nın bir arada kullanılmasının OMH hata oranları üzerindeki etkisi görülmektedir.

Çizelge 4.8. Genetik algoritma & birleştirme-1 algoritması OMH hata oranları

Hibrit Algoritma	**OMH**	**GA-OMH Birleştirme-1**	**Başarı Oranı %**
HF-6	0.6891	0.6860	0.45
HF-8	0.6891	0.6854	*0.54**
HF-10	0.6856	*0.6826**	0.44
HF-12	0.6897	0.6863	0.49

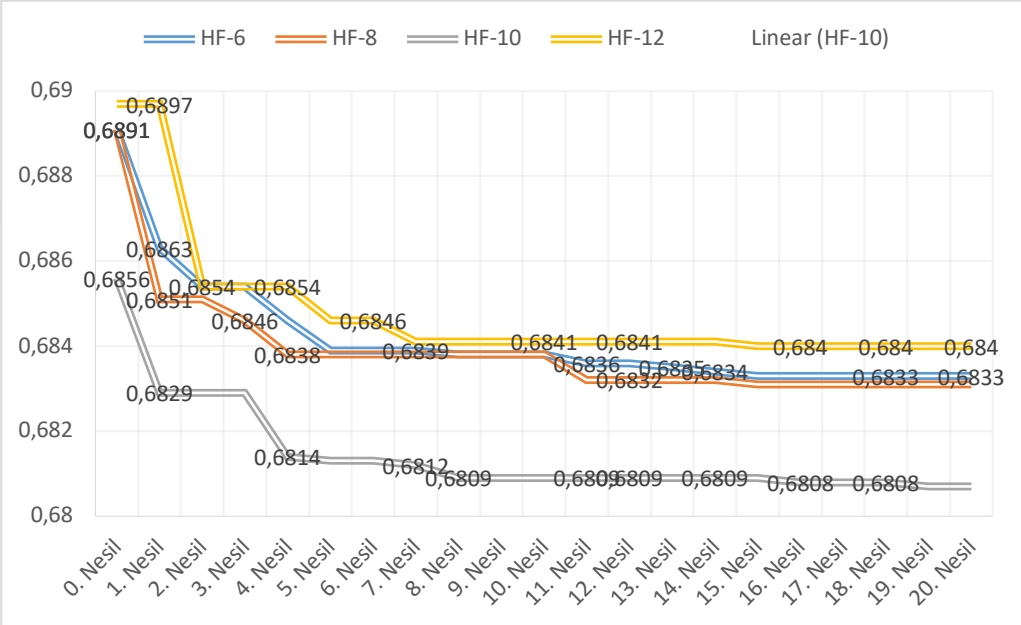
Çizelge 4.8.'de, Çizelge 4.6.'da gösterilen eşit katsayı kullanılan birleştirmelere ait hata oranları "OMH" sütununda görülmektedir. "GA-OMH Birleştirme-1" sütununda ise genetik algoritma ile en uygun katsayılar bulunan Birleştirme-1 Algoritması ile oluşturulmuş hibrit yöntemlerin minimum OMH hata oranları görülmektedir. En yüksek başarı oranının %0.54 ile HF-8 algoritmasına ve en düşük başarı oranının ise %0.44 ile HF-6 algoritmasına ait olduğu görülmektedir. HF-10 algoritması en düşük OMH hata oranlarına sahip olmasına rağmen eşit katsayı kullanılan birleştirmeye göre daha düşük bir başarı oranına sahip olmuştur.

Denklem (4.1) ile ifade edilen eşit katsayılı birleştirme yöntemine ait hata oranları (OMH sütunu) ile Birleştirme-1 Algoritmasına ait hata oranları (GA-OMH Birleştirme-1 sütunu) arasında çıkan farklar üzerinden hesaplanan başarı oranlarına dayanarak bir başarı elde edildiği söylenebilir. Yapılan bu deneyler Birleştirme-1 Algoritmasını oluştururken yapılan varsayımların doğruluğunu göstermektedir. Yani işbirlikçi veya içerik tabanlı filtreleme algoritmalarının yapmış oldukları tahminler arasındaki büyüklük faktörünün doğruluk üzerinde etkili olduğu söylenebilir.

Girdi olarak alınan bu iki tahmin ($İTF_{tahmin}$, $İF_{tahmin}$) arasındaki büyüklük faktörü bu kadar etkili olurken bu iki tahmin arasındaki farkın büyüklüğünün de bir etkisi olabileceği düşünülmüştür. Bu farkın büyüklüğünün de hesaba katıldığı Birleştirme-2

Algoritması (Bölüm 3.9.3.) adında yeni bir algoritma ile daha fazla kıyaslama yapmanın OMH hata oranları üzerinde etkili olabileceği düşünülmüştür. Bu algoritmada girdi olarak alınan tahmin değerlerinin arasındaki büyüklük faktörüne ek olarak farkın büyüklüğünün belli bir değerin üstünde olması veya olmaması durumuna göre iki tahminin bir araya gelmesinde kullanılacak katsayılar değiştirilmiştir. Bu algoritmada kullanılan büyüklük farkları olan *k* ve *m* değişkenleri ile tahmin değerlerinin katsayıları olan *x, y, z, d, f, g, j, n* değişkenleri genetik algoritma ile OMH hata oranını en küçük yapacak şekilde optimize edilmiştir.

Genetik algoritma parametreleri ayarlanırken değişken sayısının artmasından kaynaklı nesil sayısının daha yüksek bir değer (20 nesil) seçilip daha geniş bir yelpazede çözümler aranmıştır.



Şekil 4.3. Genetik algoritma & birleştirme-2 algoritması için OMH değişimi

Çizelge 4.6.'da bulunan en başarılı dört hibrit algoritmanın (HF-6, HF-8, HF-10, HF-12) bileşenleri ($İTF_{tahmin}$, $İF_{tahmin}$) Birleştirme-2 Algoritması sayesinde birleştirilmişlerdir. Birleştirme-2 Algoritmasında bulunan katsayılar genetik algoritma ile OMH hata oranını en küçük yapacak şekilde optimize edilmiştir. Genetik algoritma OMH değerlerini en düşük yapacak en uygun *x, y, z, d, f, g, j, n, k, m* değerleri bulmak için 20 nesil boyunca çalıştırılmıştır. Şekil 4.3.'de Birleştirme-2 Algoritması ve

GA'nın bir arada kullanılmasının OMH hata oranları üzerindeki etkisi görülmektedir. Birleştirme-2 Algoritması ile oluşturulmuş hibrit öneri sisteminin hem eşit katsayı kullanılan yöntem hem de Birleştirme-1 Algoritması kullanılan yöntem ile oluşturulmuş hibrit öneri sistemlerinden daha düşük OMH hata oranlarına olduğu görülmüştür.

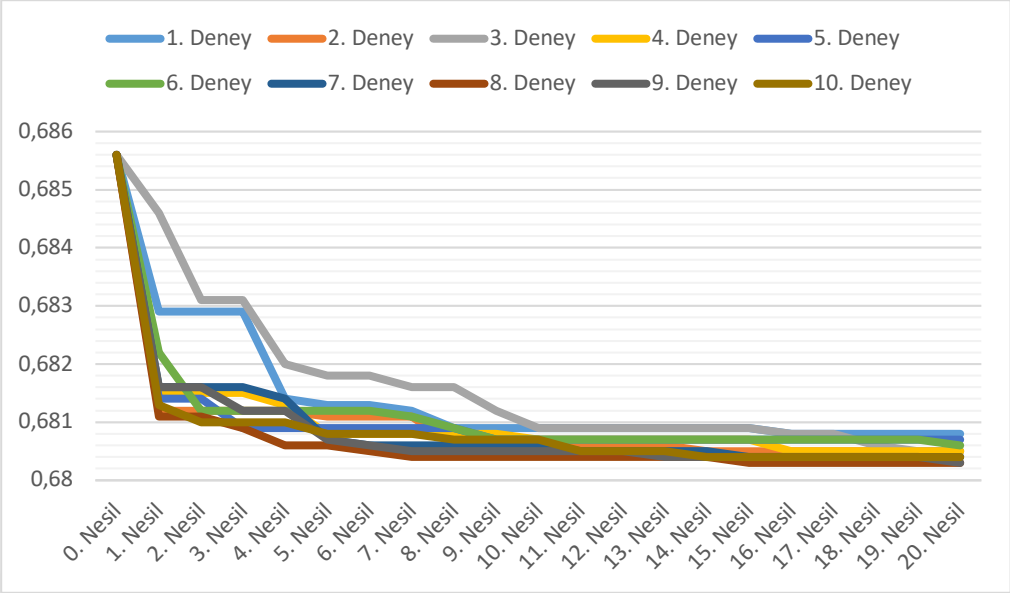
Çizelge 4.9. Genetik algoritma & birleştirme-2 algoritması OMH hata oranları

Hibrit Algoritma	**OMH**	**GA-OMH Birleştirme-2**	**Başarı Oranı %**
HF-6	0.6891	0.6833	0.84
HF-8	0.6891	0.6831	*0.87**
HF-10	0.6856	*0.6807**	0.71
HF-12	0.6897	0.6840	0.83

Çizelge 4.9.'da eşit katsayı kullanılarak oluşturulmuş hibrit yönteme ait OMH hata oranları (OMH sütunu) ve genetik algoritma ile optimize edilmiş Birleştirme-2 Algoritması kullanılarak oluşturulmuş hibrit yönteme ait OMH hata oranları (GA-OMH Birleştirme-2 sütunu) ve başarı oranları görülmektedir. En yüksek başarı oranı %0,87 ile HF-8 algoritmasına ait olurken, en düşük başarı oranı %0,68 ile HF-10 algoritmasına ait olmuştur. En düşük OMH hata oranları, başarı oranı düşük olmasına rağmen HF-10 algoritması ile elde edilmiştir.

Yapılan deneyler göz önüne alındığında işbirlikçi ve içerik tabanlı filtreleme algoritmalarının bir araya getirilmelerinde kullanılan Birleştirme-2 Algoritmasının Birleştirme-1 algoritmasından, Birleştirme-1 Algoritmasının da eşit katsayı kullanılan birleştirme yönteminden daha başarılı olduğu görülmüştür.

En başarılı sonuçların OMH hata oranlarını en düşük yapacak şekilde genetik algoritma ile katsayıları optimize edilmiş olan Birleştirme-2 Algoritması kullanılarak oluşturulmuş hibrit bir film öneri algoritması olan HF-10 algoritmasına ait olduğu yapılan deneyler ile ampirik olarak kanıtlanmıştır. Genetik algoritma sezgisel bir algoritma olduğu için her çalıştırıldığında aynı sonuçları vermeyebilir. Bu nedenle Birleştirme-2 algoritmasında kullanılan değişken katsayılarının optimizasyonu için genetik algoritma on defa çalıştırılmıştır.



Şekil 4.4. Genetik algoritma & birleştirme-2 algoritması için OMH değişimi (10 nesil)

Hibrit yöntem oluşturulurken kullanılan Birleştme-2 Algoritmasındaki değişken katsayıları OMH hata oranlarını en küçük yapacak şekilde genetik algoritma ile optimize edilmiştir. GA optimizasyon işlemleri 20 nesil için 10 defa çalıştırılmıştır. Her çalıştırmaya ait OMH hata oranları değişimleri olarak Şekil 4.4.'de görülmektedir.

Çizelge 4.10. HF-10 algoritması için OMH hata oranları

Genetik Algoritma	**OMH**	**GA-OMH Birleştirme-2**	**Başarı Oranı %**
1. Deney	0.6856	0.6808	0.70
2. Deney	0.6856	0.6805	0.74
3. Deney	0.6856	0.6805	0.74
4. Deney	0.6856	0.6805	0.74
5. Deney	0.6856	0.6807	0.71
6. Deney	0.6856	0.6806	0.73
7. Deney	0.6856	0.6804	0.76
8. Deney	0.6856	*0.6803**	*0.77**
9. Deney	0.6856	*0.6803**	*0.77**
10. Deney	0.6856	0.6804	0.76
Ortalama	**0.6856**	**0.6805**	**0.742**

Çizelge 4.10.'da görüldüğü üzere genetik algoritmanın on defa çalıştırılması sonucunda elde edilen sonuçlarda ciddi bir değişim görülmemiştir. Denklem (3.11) ile ifade edilen standart sapma denklemi kullanılarak yapılan hesaplamalar sonucunda standart sapmanın 0.0000024 ve aritmetik ortalamanın 0.6805 çıktığı sonuçlar Birleştirme-2 Algoritması değişkenleri için en uygun değerlerin yaklaşık olarak bulunduğunu göstermektedir.

4.3. Önerilen Hibrit Film Öneri Sistemi Değerlendirmesi

Christakou ve Vrettos (2007) ile Milli (2013) farklı tarihlerde yayınlanmış MovieLens 100K veri seti üzerinde çalışma yapmışlardır. Fakat toplanan veriler (kullanıcılar, filmler, değerlendirmeler) farklı tarihlerde toplandığı için birebir aynı veri seti değillerdir. Çünkü MovieLens sistemi hala güncel olarak kullanıcı görüşleri toplamakta ve veri setlerini sürekli güncellemektedir. Bu yüzden daha önce yapılmış çalışmalar ile karşılaştırma yapmak sağlıklı sonuçlar vermeyecektir.

Sarwar ve ark. (2000) tarafından yapılan çalışma ile öneri sistemlerinin sahip olduğu Seyreklik ve Boyutsallık (kullanılan kullanıcı-ürün matrisinin boyutunun büyük olması) sorunlarına bir çözüm olarak Tekil Değer Ayrıştırma metodu ortaya atılmıştır. TDA metodu [kullanıcı X ürün] matrisleri üzerinde uygulanan bir boyut azaltma tekniğidir. TDA sayesinde etkisi az olan özellikler atılarak matrisin boyutu azaltılmış bu sayede daha az matematiksel işlem ile daha az maliyetli bir öneri sistemi metodu ortaya çıkmıştır (Sarwar ve ark., 2000; Adomavicius ve Tuzhilin, 2005). Netflix Prize yarışmasında kazanan algoritma TDA algoritmasının ana hatlarını oluşturduğu yeni bir algoritma olan TDA++ algoritmasıdır (Surprise, 2017).

Tez kapsamında önerilen HF-10 algoritması, hibrit yöntem oluşturulurken kullanılan geleneksel yöntemler olan işbirlikçi filtreleme algoritması (KİF-11) ve içerik tabanlı filtreleme algoritması (İTF-11) ile Hug (2017), tarafından kodları yayınlanmış olan TDA ve TDA++ algoritmaları aynı veri seti üzerinde çalıştırılmış ve elde edilen sonuçları Çizelge 4.11.'de verilmiştir.

Çizelge 4.11. HF-10 algoritması değerlendirilmesi

Algoritma	**OMH**	**HKOK**
TDA	0.6970	0.9018
TDA ++	0.6841	*0.8877**
İTF-11	0.7242	0.9292
KİF-11	0.6986	0.9207
HF-10	*0.6805**	0.8922

Çizelge 4.11.'de görüldüğü üzere önerilen HF-10 isimli hibrit film öneri algoritması HKOK hata oranları üzerinden yapılan karşılaştırmaya TDA++ algoritması hariç diğer 3 algoritmaya göre daha iyi sonuçlar vermiştir. Genetik algoritma için uygunluk fonksiyonu olarak OMH hata oranı denkleminin kullanılmasının TDA++ algoritmasının HKOK hata oranında daha başarılı sonuçlar vermesinde etkili olduğu düşünülmektedir.

Genetik algoritmada en düşük değerini bulmaya çalıştığımız OMH hata oranları üzerinden yapılan karşılaştırmaya göre önerilen yöntem olan HF-10 algoritması diğer dört algoritmadan daha başarılı sonuçlar vermiştir. Bu sonuçlar bize Birleştirme-2 Algoritması oluşturulurken yapılan varsayımların hibrit bir film öneri algoritmasının yaptığı tahminlerin doğruluğu üzerinde etkili olduğunu göstermektedir. Yapılan tahminler arasındaki büyüklük faktörü ve aradaki farkın mutlak değeri doğruluk üzerinde etkili olmaktadır.

5. SONUÇ ve ÖNERİLER

Bu çalışmada, son yıllarda çevrimiçi ortamlarda yoğun bir şekilde kullanılan öneri sistemleri hakkında bilgi verilmektedir. Öneri sistemleri film, kitap, müzik, tatil, eşya vb. bir sürü alanda başarılı bir şekilde uygulanmaktadır. Öneri sistemleri genel olarak benzer kullanıcıların benzer ürünleri tercih edeceğini veya kullanıcının geçmişteki tercihlerine benzer tercihleri gelecekte de yapacağı varsayılarak çalışmaktadır.

Öneri sistemleri, kullanılan yaklaşıma göre işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit filtreleme olarak üç farklı şekilde uygulanmaktadır. Doğru filtreleme türünü seçmek öneri sistemleri için en önemli etkendir. Örneğin güncel haber öneri sistemleri için içerik tabanlı bir yaklaşım seçildiği düşünülürse başarı oranı düşecektir. Çünkü kullanıcı benzer haberleri defalarca okumak istemeyecektir. Ama spor haberleri için işbirlikçi filtreleme doğru bir yaklaşım olabilir. Çünkü taraftarlar tuttukları takımın elde ettiği başarıları farklı kaynaklardan farklı bakış açılarıyla tekrar tekrar okumaktan veya dinlemekten keyif alabilirler (Taşçı, 2015).

Bu tez çalışmasında farklı yöntemler kullanılarak hata oranları düşürülen hibrit bir algoritma önerilmiştir. HF-10 adı verilen bu hibrit yöntem içerik tabanlı filtreleme ve işbirlikçi filtreleme yöntemlerinin yaptıkları tahminlerin bir araya getirilmeleriyle oluşturulmuştur. İçerik tabanlı filtrelemenin doğruluk oranını arttırmak için yeni bir öznitelik eklenmiş ve diğer tüm özniteliklerin OMH hata oranlarıyla karşılaştırılmıştır. Yeni eklenen öznitelik ile beraber oluşturulan içerik tabanlı filtreleme algoritması daha başarılı olmuştur. Kullanılan işbirlikçi filtreleme, benzerlik hesaplama ve tahmin yapma aşamalarında kullanılan yöntemlerin kombinasyonları ile yapılan deneysel karşılaştırmalar sonucunda tespit edilen en başarılı yöntemlerin bir araya getirilmesiyle oluşturulmuştur. Bu her iki algoritmanın tahminleri tez kapsamında önerilen Birleştirme-2 Algoritması sayesinde bir araya getirilmiştir. Birleştirme-2 Algoritmasında kullanılan katsayılar OMH hata oranlarını en düşük yapacak şekilde tasarlanmış GA ile optimize edilmiştir.

Birleştirme-2 Algoritmasında işbirlikçi ve içerik tabanlı filtreleme algoritmalarından gelen tahminler hem büyüklüklerine göre hem de aradaki farkın büyüklüğüne göre farklı oranlarda birleştirilmişlerdir. OMH hata oranları üzerinden yapılan deneysel karşılaştırmalar neticesinde önerilen hibrit yöntem olan HF-10 algoritmasının kendi bileşenleri olan işbirlikçi filtreleme algoritması ve içerik tabanlı filtreleme algoritması ile TDA ve TDA++ algoritmalarından daha başarılı sonuçlar verdiği kanıtlanmıştır.

İşbirlikçi filtreleme ve içerik tabanlı filtrelemenin yaptıkları tahmini değerlendirmelerin arasında bir ilişki olduğu yapılan deneyler ile ortaya çıkmaktadır. MovieLens 100K veri seti üzerinde yapılan değerlendirmeler her iki geleneksel filtreleme algoritmasının yaptıkları tahminler arasındaki büyüklüğün ve büyüklük farkının oluşturulacak hibrit yöntemin doğruluğu üzerinde önemli faktörler olduğunu göstermiştir. Ayrıca yapılan deney sonuçlarına göre film öneri sistemlerinde ürün tabanlı işbirlikçi filtrelemenin genel olarak kullanıcı tabanlı işbirlikçi filtrelemelerden daha başarılı olduğu ortaya çıkmıştır.

Bundan sonraki çalışmalarda Birleştirme-2 Algoritması yeni varsayımlarla geliştirileceği gibi farklı veri setleri üzerinde uygulanarak performans değerlendirmeleri sayesinde iyileştirmeler yapılabilir.

KAYNAKLAR

ADOMAVİCİUS, G., and TUZHİLİN, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6): 734-749.

AHUJA, R., SOLANKİ, A., and NAYYAR, A., 2019. Movie recommender system using K-means clustering and K-nearest neighbor. In 2019 IEEE 9th International conference on cloud computing, data science & engineering (confluence), 8-10 January, India, p.263-268.

AIMGROUP, 2016. Uluslararası bir alman danışmanlık, iş zekâsı ve haber analiz firması. Url: https://aimgroup.com/2016/10/05/gittigidiyor-buys-recommendation-machine-of-gravity-rd-2/, (Erişim: 21/07/2019).

ALHİJAWİ, B., and KİLANİ, Y., 2016. Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 26-29 June, USA, p.1-6.

ALSPECTOR, J., KOİCZ A., and KARUNANİTHİ, N., 1997. Feature-based and clique-based user models for movie selection: A comparative study. User Modeling and User-Adapted Interaction, 7(4): 279-304.

AMAZON.COM. Seattle merkezli bir amerikan çokuluslu teknoloji şirketi. Url: https://www.amazon.com/, (Erişim: 08/07/2019).

ANAND, S., and MOBASHER, B., 2003. Intelligent techniques for web personalization. In IJCAI Workshop on Intelligent Techniques for Web Personalization, Springer, Berlin, 340p.

ANIL, U., ve AKCAYOL, M. A., 2017. Öğrenebilen ve adaptif tavsiye sistemleri için karşılaştırmalı ve kapsamlı bir inceleme. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi, 33(3): 13-34.

AYGÜN, C., 2016. Genetik Algoritma Kullanılarak İçerik Tabanlı Kitap Tavsiye Sistemi Geliştirilmesi Development Of Content Based Book Recommendation System Using Genetic Algorithm. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 43s.

BAKIR, Ç., ve ALBAYRAK, S., 2014. User based and item based collaborative filtering with temporal dynamics. In 2014 22nd Signal Processing and Communications Applications Conference (Siu), 23-25 April, Trabzon, p.252-255.

BALABANOVİC, M., and SHOHAM, Y., 1997. Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3): 66-72.

BATTAL, A., 2009. Semantik web ile geliştirilen bir televizyon program öneri sistemi. TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 78s.

BASU, C., HİRSH, H., and COHEN, W., 1998. Recommendation as classification: using social and content-based information in recommendation. In Proceedings of the 15th national conference on artificial intelligence, 26-30 Jully, Madison, p.714–720.

BILGE, A., BATMAZ, Z., ve POLAT, H., 2016. Maskelenmiş veriler için kümeleme-tabanlı şilin atak tespit yöntemi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 28(2): 207-216.

BOBADİLLA, J., ORTEGA, F., HERNANDO, A., and ALCALÁ, J., 2011. Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-based systems, 24(8): 1310-1316.

BURKE, R., 2002. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4): 331-370.

CHRISTAKOU, C., VRETTOS, S., and STAFYLOPATIS, A., 2007. A hybrid movie recommender system based on neural networks. International Journal on Artificial Intelligence Tools, 16(05): 771-792.

ÇETİN, E. İ., KURUÜZÜM, A., ve IRMAK, S., 2008. Ekip çizelgeleme probleminin küme bölme modeli ile çözümü. Journal of Aeronautics and Space Technologies, 3(4): 47-54.

ÇOLAK, S., 2010. Genetik algoritmalar yardımı ile gezgin satıcı probleminin çözümü üzerine bir uygulama. Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 19(3): 423-438.

DOOMS, S., 2013. Dynamic Generation Of Personalized Hybrid Recommender Systems. In: Proceedings Of The 7th ACM Conference On Recommender Systems, 12-16 October, New York, p.443-446.

EMEL, G. G., ve TAŞKIN, Ç., 2002. Genetik algoritmalar ve uygulama alanlari. Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21(1): 129-152.

GİTTİGİDİYOR.COM. Amerikalı eBay şirketinin Türkiye'deki sanal pazarı. Url: https://www.gittigidiyor.com/, (Erişim: 20/07/2019).

GİTTİGİDİYOR KURUMSAL, 2017. Amerikalı eBay şirketinin Türkiye'deki sanal pazarı. Url: https://kurumsal.gittigidiyor.com/hakkinda/gittigidiyor-buyuk-veriyi-kullanarak-alisveris-deneyimini-kisisellestiriyor/, (Erişim: 20/07/2019).

GOLDBERG, D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston, 432p.

GOLDBERG, D. E., NICHOLS, D., OKI, B. M., and TERRY, D., 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12): 61-70.

GOMEZ-URIBE, C. A., and HUNT, N., 2016. The netflix recommender system: Algorithms, business value, and innovation. ACM transactions on management ınformation systems (TMIS), 6(4): 1-19.

GONG, S., 2010. A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7): 745-752.

GÖKSEDEF, M., and GÜNDÜZ-ÖĞÜDÜCÜ, Ş., 2010. Combination of Web page recommender systems. Expert Systems with Applications, 37(4): 2911-2922.

GRAVİTY R&D, 2016. Öneri sistemleri teknolojileri hizmeti veren bir firma. Url: https://www.slideshare.net/gravityr-d/gravity-sunum-66206902, (Erişim: 30/06/2019).

HARPER, F. M., and KONSTAN, J. A., 2016. The movielens datasets: History and context. ACM transactions on interactive intelligent systems (tiis), 5(4): 1-19.

HUG, N., 2017. Surprise, a Python scikit for building and analyzing recommender systems Columbia University, Url: http://surpriselib.com, (Erişim: 18/09/2019).

JANG, J. S. R., 1997. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. IEEE Transactions on automatic control, 42(10): 1482-1484.

JANNACH, D., ZANKER, M., FELFERNİG, A., and FRİEDRİCH, G., 2011. Recommender System An Introduction, Cambridge University Press, New York, 331p.

KUMAR, M., YADAV, D.K., SINGH, A., and GUPTA, V. K., 2015. A movie recommender system: Movrec. International Journal of Computer Applications, 124(3): 7-11.

LAM, S. K., and RIEDL, J., 2004. Shilling recommender systems for fun and profit. In: Proceedings of the 13th international conference on World Wide Web, ACM,17-20 May, New York, p.393-402.

LAROSE, D. T., and LAROSE, C. D., 2014. Discovering knowledge in data: an introduction to data mining. John Wiley & Sons, New Jersey, 317p.

LAVETI, R. N., CH, J., PAL, S. N., and BABU, N. S. C., 2016. A hybrid recommender system using weighted ensemble similarity metrics and digital filters. In 2016 IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) IEEE, 19-22 December, Hyderabad, p.32-38.

LEKAKOS, G., and GİAGLİS, G. M., 2004. A lifestyle-based approach for delivering personalized advertisements in digital interactive television. J Comput Mediat Commun, 9(2): JCMC92.

LEKAKOS, G., and CARAVELAS, P., 2008. A hybrid approach for movie recommendation. Multimedia tools and applications, 36(1-2): 55-70.

LU, Z., DOU, Z., LIAN, J., XIE, X., and YANG, Q., 2015. Content-based collaborative filtering for news topic recommendation. In Twenty-ninth AAAI conference on artificial intelligence, 25-30 January, USA, p.217-223.

MACKENZİE, I., MEYER, C., and NOBLE, S., 2012. Amerikalı uluslararası yönetim danışmanlık firması. Url: https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers, (Erişim: 12/08/2019).

MELVİLLE, P., MOONEY, R. J., and NAGARAJAN, R., 2002. Content-boosted collaborative filtering for improved recommendations. Aaai/iaai, 23(2002): 187-192.

MILLER, B. N., ALBERT, I., LAM, S. K., KONSTAN, J. A., and RIEDL, J., 2003. MovieLens unplugged: experiences with an occasionally connected recommender system. In Proceedings of the 8th international conference on Intelligent user interfaces, 8(3): p.263-266.

MIRANDA, T., CLAYPOOL, M., GOKHALE, A., MIR, T., MURNIKOV, P., NETES, D., and SARTIN, M., 1999. Combining content-based and collaborative filters in an online newspaper. In In Proceedings of ACM SIGIR Workshop on Recommender Systems, California, p.6-22.

MITCHELL, M., 1998. An Introduction to Genetic Algorithms. MIT Press, London, 211p.

MİLLİ, M., 2013. Film öneri sistemleri için hibrit bir yöntem geliştirilmesi a new hybrid approach for movie recommender systems. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İzmir, 75s.

MUKHERJEE, R., DUTTA, P. S., and SEN, S., 2001. MOVIES2GO-A new approach to online movie recommendation. In: Proceedings of the IJCAI Workshop on intelligent Techniques for Web Personalization, Seatle, p.26-34.

MULIK, A. T., 2013. Content based and collaborative filtering for online movie recommendation. International Journal of Scientific and Engineering Research, 4(10): 1113-1121.

N11 KURUMSAL, 2014. Doğuş Grubu ile Güney Koreli SK Group'un ortak iştirakı bir sanal pazar. Url: https://kurumsal.n11.com/basin-bultenleri/n11-com-dan-kisisellestirilmis-oneri-sistemi, (Erişim: 12/07/2019).

NETFLİX PRİZE, 2009. The Netflix Prize. Url: http://www.netflixprize.com/, (Erişim: 06/07/2019).

NETFLİX, 2016. The Netflix Help. Url: https://help.netflix.com/tr/node/100639/, (Erişim: 19/07/2019).

N11.COM. Doğuş Grubu ile Güney Koreli SK Group'un ortak iştirakı bir sanal pazar. Url: https://www.n11.com, (Erişim: 12/072019).

ÖZGÖBEK, Ö., ve ERDUR, R. C., 2015. Öneri sistemleri ve bir uygulama alanı olarak haber öneri sistemleri. Akademik Bilişim Konferansları, 4-6 Şubat, Eskişehir, 31s.

TUĞ, E., 2005. Genetik algoritmalar ile tıbbi veri madenciliği. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya, 74s.

PATRA, S., and GANGULY, B., 2019. Improvising singular value decomposition by KNN for use in movie recommender systems. Journal of Operations and Strategic Planning, 2(1): 22-34.

PAZZANİ, M., 1999. A framework for collaborative, content-based and demographic filtering. Artificial intelligence review, 13(5-6): 393-408.

PERA, M. S., and NG, Y. K., 2013. A group recommender for movies based on content similarity and popularity. Information Processing & Management, 49(3): 673-687.

PEREİRA, A. L. V., and HRUSCHKA, E. R., 2015. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowledge-Based Systems, 82(2015): 11-19.

PORTER, J., 2007. What to wear daily report writer. Url: http://bokardo.com/archives/jeff-bezos-on-amazons-personalization-strategy/, (Erişim: 08/07/2019).

SADEGHİ, M., and ASGHARİ, S. A., 2017. Recommender systems based on evolutionary computing: a survey. Journal of Software Engineering and Applications, 10(05): 407.

SAĞIROĞLU Ş., and KOÇ O., 2017. Büyük veri ve açık veri analitiği: Yöntemler ve uygulamalar. Gazi Üniversitesi Big Data Center, Ankara, 331p.

SARWAR, B., KARYPIS, G., KONSTAN, J., and RIEDL, J., 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, 1-5 May, Hong Kong, p.285-295.

SCHAFFER, J. B., KONSTAN, J., and RIEDL, J., 1999. Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference on Electronic commerce, 3-5 November, Colorado, p.158-166.

SEVLİ, O., ve KÜÇÜKSİLLE E. U., 2016. Türkçe paylaşım yapan kullanıcılar için sosyal ağ tabanlı analiz ve tavsiye sistemi. Akademik Platform Mühendislik ve Fen Bilimleri Dergisi, 4(3): 15-28.

SHANİ, G., and GUNAWARDANA, A., 2011. Evaluating recommendation systems. In: Recommender systems handbook, Springer, Boston p. 257-297.

SIMSEKLI, U., BIRDAL, T., KOÇ, E., ve CEMGIL, A. T., 2013. Çevrimiçi servisler için ayrısım tabanlı tavsiye sistemi, 21st Signal Processing and Communications Applications Conference (SIU), 24-26 April, Haspolat, Turkey, p.0-0.

SOBOROFF, I., and NICHOLAS, C., 1999. Combining content and collaboration in text filtering. In: Proceedings of the IJCAI, 99(1999): 86-91.

STERN D., HERBRİCH R. ve GRAEPEL T., 2009. Thore. Matchbox: Large scale bayesian recommendations. In: 18th International World Wide Web Conference, 20-24 April, New York, p.111-120.

SU, X., and KHOSHGOFTAAR, T. M., 2009. A survey of collaborative filtering techniques. Advances in artificial intelligence, 18 December, Australia, 19p.

SURPRİSE, 2015. Mean squared difference, the python framework. Url: https://surprise.readthedocs.io/en/stable/similarities.html, (Erişim: 20/08/2019).

TAN, R. K., ve BORA, Ş., 2017. Modelleme ve benzetim ortamında parametre optimizasyonu ve kullanılan teknikler. Mühendislik Bilimleri ve Tasarım Dergisi, 5(3): 685-697.

TAŞÇI, S., 2015. İçerik bazlı medya takip ve haber tavsiye sistemi. Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 70s.

ULUYAĞMUR, M., 2012. Hibrit film öneri sistemi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 51s.

WANG, Z., YU, X., FENG, N., and WANG, Z., 2014. An improved collaborative movie recommendation system using computational intelligence. Journal of Visual Languages & Computing, 25(6): 667-675.

WIKIPEDIA, 2018. Content-control software, the free encyclopedia, Wikimedia Foundation Inc. Url: http://en.wikipedia.org/wiki/Content_filtering, (Erişim: 14/06/2019).

ZHANG, Y., and KOREN, J., 2007. Efficient bayesian hierarchical user modeling for recommendation system. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 23-27 July, Amsterdam, p.47-54.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Hamit SEYREK
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Şırnak – 01/02/1987
e-mail : hamitseyrek@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Şırnak Anadolu Lisesi, Merkez/ Şırnak	2005
Üniversite	: Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, Elazığ	2011
Yüksek Lisans	: Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği A.B.D., Şanlıurfa	2020

İŞ DENEYİMLERİ

Yıl	Kurum	Unvan
2013-Devam ediyor	GAP BKİ Başkanlığı	Mühendis

UZMANLIK ALANI

Veri Madenciliği, Sunucu Yönetim Sistemleri, Öneri Sistemleri, Web Tasarım, Mobil Uygulama.

YABANCI DİLLER

İngilizce

YAYINLAR

SEYREK, H., AYDILEK I. B., 2018. Veri Madenciliğinde kullanılan sınıflandırma algoritmalarının farklı veri kümeleri ile karşılaştırılması. Uluslararası GAP Mühendislik Kongresi, 8-10 Kasım, Harran Üniversitesi Mühendislik Fakültesi, Şanlıurfa, s.701-707.