



**KULLANICI / ÜRÜN ÇİFTLERİ İÇİN  
EN İYİ ÖNERİ ALGORİTMALARININ  
TESPİT EDİLMESİ**

Yüksek Lisans Tezi

İsmail TERZİ

Eskişehir, 2017

**KULLANICI / ÜRÜN ÇİFTLERİ İÇİN EN İYİ ÖNERİ ALGORİTMALARININ  
TESPİT EDİLMESİ**

**İsmail TERZİ**

**YÜKSEK LİSANS TEZİ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman Yrd. Doç. Dr. Alper BİLGE**

**Eskişehir**

**Anadolu Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Ocak, 2017**

*Bu Tez Çalışması 114E571 no.lu TUBİTAK projesi tarafından kısmen desteklenmiştir.*

## JÜRİ VE ENSTİTÜ ONAYI

**İsmail TERZİ'nin "Kullanıcı / Ürün Çiftleri İçin En İyi Öneri Algoritmalarının Tespit Edilmesi"** başlıklı tezi 17/01/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca, Bilgisayar Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

**Unvanı-Adı Soyadı**

**İmza**

**Üye (Tez Danışmanı) : Yrd. Doç. Dr. Alper BİLGE**

.....

**Üye : Doç. Dr. Cihan KALELİ**

.....

**Üye : Yrd. Doç. Dr. Uğur GÜREL**

.....

.....

Enstitü Müdürü

## ÖZET

### KULLANICI / ÜRÜN ÇİFTLERİ İÇİN EN İYİ ÖNERİ ALGORİTMALARININ TESPİT EDİLMESİ

İsmail TERZİ

Bilgisayar Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ocak, 2017

Danışman: Yrd. Doç. Dr. Alper BİLGE

İnternetin yaygınlaşması ve iletişim teknolojilerinin gelişmesiyle birlikte İnternet üzerinden sunulan hizmetler artmıştır. Bu artışın sonucunda kullanıcılar incelenmesi ve takip edilmesi gereken çok miktarda bilgi ve binlerce ürün/hizmetle karşı karşıya kalmaktadırlar. Bu durum “aşırı bilgi yükü” olarak tanımlanmaktadır. Bu alanda kullanıcılara, ürün ve hizmet seçimi konusunda yardımcı olacak öneri sistemleri geliştirilmiştir. Çevrimiçi servislerde kullanıcıların geçmişteki tercihlerine ve benzer kullanıcıların tercihlerine bakılarak kişiselleştirilmiş öneri sunmak için kullanılan en popüler öneri sistemi Ortak Filtreleme tabanlı öneri sistemleridir. Yüksek doğruluk Ortak Filtreleme tabanlı öneri sistemleri tarafından sağlanması gereken en önemli özelliklerden biridir. Bu tez kapsamında yapılan çalışmalarda hedeflenen, Ortak Filtreleme tabanlı öneri sistemleri tarafından üretilen önerilerin doğruluğunu iyileştirmektir. Günümüzde kullanılan öneri sistemleri genellikle tek bir öneri algoritması kullanmakta ve her kullanıcı için aynı algoritmayı kullanarak öneriler üretmektedirler. Bu çalışmada Ortak Filtreleme işlemleri için bir tane algoritma kullanmak yerine, alanında en iyi algoritmalar arasında gösterilen altı tane algoritma kullanılmıştır. Deneysel çalışmalarda kullanıcı veya kullanıcı/ürün çifti için her algoritmanın farklı doğrulukta öneriler ürettiği görülmüştür. Altı algoritmanın kullanıcı veya kullanıcı/ürün çiftine ürettiği tahmin değerlerinden en iyisi seçildiğinde sistem doğruluğunun yüksek anlamlılık seviyelerinde iyileşebileceği gösterilmiştir.

**Anahtar Sözcükler:** Öneri Sistemleri, Ortak Filtreleme, Doğruluk, Doğruluk İyileştirme, Ortak Filtreleme Algoritmaları

## **ABSTRACT**

### **DETERMINING THE BEST RECOMMENDATION ALGORITHMS FOR USER/ITEM PAIRS**

**İsmail TERZİ**

**Department of Computer Engineering**

**Anadolu University, Graduate School of Science, January, 2017**

**Supervisor: Asst. Prof. Alper BİLGE**

With the development and the spread of the Internet and communication technologies, the services offered via the Internet has increased. As a result of this increase, users are faced with thousands of products and information to be examined and followed. This situation is defined as "Information Overload". Recommenders systems have been developed to assist in the selection of products and services to the user. The most popular recommendation system used to provide personalized recommendations based on users' preferences for the past and similar users in online services is the Collaborative Filtering based recommendation systems. Accuracy is one of the most important features that must be satisfied by the Collaborative Filtering based recommendation systems. The aim of this thesis is to improve the accuracy of the recommendation produced by the Collaborative Filtering based recommendation systems. Today Collaborative Filtering Recommender systems typically utilizes only one recommendation algorithm and provide predictions for all users by using the same algorithm. The hypothesis is that, there is one best algorithm for each user or user/item pairs, instead of hiring the same recommendation algorithms for all users or user/item pairs, hiring the best recommendation algorithms for different users or user/item pairs can increase accuracy of recommendation system. As a result of experimental study it is proved that, there is one best algorithm for each user or user/item pairs, hiring the best algorithm can increase accuracy of Collaborative Filtering recommendation system.

**Keywords:** Recommender Systems, Collaborative Filtering, Accuracy, Accuracy Improvement, Collaborative Filtering Algorithms

17/01/2017

## **ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ**

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

.....

İsmail TERZİ

## İÇİNDEKİLER

### Sayfa

BAŞLIK SAYFASI .....	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET .....	iii
ABSTRACT .....	iv
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ .....	v
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ .....	x
SİMGELER VE KISALTMALAR DİZİNİ.....	xi
1. GİRİŞ .....	1
2. TEZİN AMACI .....	3
3. İÇERİK TABANLI FİLTRELEME ÖNERİ SİSTEMLERİ .....	5
4. ORTAK FİLTRELEME ÖNERİ SİSTEMLERİ .....	6
4.1. Hafıza Tabanlı Algoritmalar.....	9
4.2. Model Tabanlı Algoritmalar .....	11
4.3. Ortak Filtreleme Öneri Sistemlerinin Problemleri .....	12
5. KARMA TAVSİYE SİSTEMLERİ .....	14
6. VERİ KÜMELERİ .....	15
7. DEĞERLENDİRME ÖLÇÜTLERİ .....	16
8. İLGİLİ ÇALIŞMALAR.....	18
9. ALGORİTMİK VE DENEYSEL ÇERÇEVE.....	21
9.1. Kullanılan Veri Kümeleri .....	21
9.2. Kullanıcı Oylarının z-Puanı Normalizasyonu.....	21
9.3. Kullanılan Değerlendirme Ölçütleri.....	22
9.4. Kullanılan Algoritmalar .....	22

9.4.1. Hafıza tabanlı algoritmalar .....	23
9.4.1.1. Kullanıcı tabanlı K en yakın komşu .....	23
9.4.1.2. Kullanıcı tabanlı eşik değeri .....	24
9.4.2. Model tabanlı algoritmalar.....	25
9.4.2.1. Ürün tabanlı K yakın komşu .....	25
9.4.2.2. Ağırlıklandırılmış Bir Eğimli Tahmin Edici.....	25
9.4.3. Hibrit algoritmalar .....	26
9.4.3.1. K-Ortalamlar .....	26
9.4.3.2. Bulanık C- ortalamalar.....	27
9.5. Kullanılan Metodoloji.....	27
10. DENEYSEL ÇALIŞMALAR.....	30
10.1. Kullanılan Algoritmaların <i>OMH</i> ve <i>HKOK</i> Değerleri .....	30
10.2. Kullanıcı İçin En İyi Algoritmanın Tespit Edilmesi.....	34
10.2.1. MovieLens veri kümesi deney sonuçları .....	34
10.2.2. Netflix Deney Sonuçları .....	37
10.3. Kullanıcı/Ürün Çifti İçin En İyi Algoritmanın Tespit Edilmesi .....	41
10.3.1. MovieLens deney sonuçları.....	41
10.3.2. Netflix Deney Sonuçları .....	45
11. SONUÇLAR VE DEĞERLENDİRME .....	50
KAYNAKÇA .....	51
ÖZGEÇMİŞ.....	58



## TABLolar DİZİNİ

### Sayfa

<b>Tablo 4.1.</b> 5-Yıldızlı Oy Değerlikli Kullanıcı-Ürün Matrisi .....	7
<b>Tablo 4.2.</b> İkili Oy Değerlikli Kullanıcı-Ürün Matrisi.....	7
<b>Tablo 4.3.</b> Basit Bir Kullanıcı-Ürün Matrisi.....	8
<b>Tablo 4.4.</b> Örnek Kullanıcı-Ürün Matrisi.....	11
<b>Tablo 6.1.</b> Veri kümeleri.....	15
<b>Tablo 9.1.</b> Netflix Veri Kümesininininden Seçilen Altküme .....	21
<b>Tablo 9.2.</b> Deneylerde Kullanılan Algoritmalar .....	23
<b>Tablo 9.3.</b> Ürünler Arasında Benzerlik Hesabı.....	25
<b>Tablo 9.4.</b> Kullanılan TEST Kümesindeki Kullanıcı, Ürün ve Oy Sayıları .....	27
<b>Tablo 9.5.</b> Örnek Tahmin Kümeleri KS=500 .....	28
<b>Tablo 9.6.</b> Sonuç Kümesi Örneği KS=500 .....	29
<b>Tablo 10.1.</b> MovieLens Veri Kümesi 6 Algoritmanın Deney Sonuçları .....	30
<b>Tablo 10.2.</b> Netflix Veri Kümesi 6 Algoritmanın Deney Sonuçları.....	32
<b>Tablo 10.3.</b> MovieLens, Kullanıcı İçin En İyi Algoritma KS=500.....	34
<b>Tablo 10.4.</b> MovieLens, Kullanıcı İçin En İyi Algoritma KS=1.000 .....	35
<b>Tablo 10.5.</b> MovieLens, Kullanıcı İçin En İyi Algoritma KS=2.000 .....	35
<b>Tablo 10.6.</b> MovieLens, Kullanıcı İçin En İyi Algoritma KS=4.000 .....	36
<b>Tablo 10.7.</b> Kullanıcı İçin En İyi Algoritma Seçildiğinde OMH.....	36
<b>Tablo 10.8.</b> Netflix, Kullanıcı İçin En İyi Algoritma KS=500 .....	38
<b>Tablo 10.9.</b> Netflix, Kullanıcı İçin En İyi Algoritma KS=1.000.....	38
<b>Tablo 10.10.</b> Netflix, Kullanıcı İçin En İyi Algoritma KS=2.000 .....	39
<b>Tablo 10.11.</b> Netflix, Kullanıcı İçin En İyi Algoritma KS=4.000 .....	39
<b>Tablo 10.12.</b> Kullanıcı İçin En İyi Algoritma Seçildiğinde OMH .....	40
<b>Tablo 10.13.</b> Kullanıcı için En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanılan 6 farklı Algoritmanın Ortalama Doğruluğu için t-test Sonuçları.....	41
<b>Tablo 10.14.</b> MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=500 .....	42
<b>Tablo 10.15.</b> MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=1.000 .....	42
<b>Tablo 10.16.</b> MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=2.000 .....	43

<b>Tablo 10.17.</b> MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=4.000 .....	<b>43</b>
<b>Tablo 10.18.</b> Kullanıcı/Ürün Çifti İçin En İyi Algoritma Seçildiğinde OMH .....	<b>44</b>
<b>Tablo 10.19.</b> Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=500.....	<b>45</b>
<b>Tablo 10.20.</b> Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=1.000 .....	<b>46</b>
<b>Tablo 10.21.</b> Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=2.000 .....	<b>46</b>
<b>Tablo 10.22.</b> Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=4.000 .....	<b>46</b>
<b>Tablo 10.23.</b> Kullanıcı/Ürün Çifti İçin En İyi Algoritma Seçildiğinde OMH .....	<b>47</b>
<b>Tablo 10.24.</b> Kullanıcı/Ürün çifti için En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanılan 6 farklı Algoritmanın Ortalama Doğruluğu için t-test Sonuçları.....	<b>48</b>
<b>Tablo 10.25.</b> Kullanıcı/ürün Çifti İçin En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanıcı İçin En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk için t-test Sonuçları .....	<b>49</b>

## ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
<b>Şekil 10.1.</b> MovieLens, Kullanılan Algoritmaların OMH Değerleri .....	<b>31</b>
<b>Şekil 10.2.</b> MovieLens, Kullanılan Algoritmaların HKOK Değerleri .....	<b>31</b>
<b>Şekil 10.3.</b> Netflix, Kullanılan Algoritmaların OMH Değerleri .....	<b>33</b>
<b>Şekil 10.4.</b> Netflix, Kullanılan Algoritmaların HKOK Değerleri .....	<b>33</b>
<b>Şekil 10.5.</b> MovieLens, Kullanıcı için En iyi Algoritma .....	<b>37</b>
<b>Şekil 10.6.</b> Netflix, Kullanıcı için En iyi Algoritma .....	<b>40</b>
<b>Şekil 10.7.</b> MovieLens, Kullanıcı/Ürün Çifti için En iyi Algoritma .....	<b>44</b>
<b>Şekil 10.8.</b> Netflix, Kullanıcı/Ürün Çifti için En iyi Algoritma .....	<b>47</b>

## SİMGELER VE KISALTMALAR DİZİNİ

<b>ABETE</b>	: Ağırlıklandırılmış Bir Eğimli Tahmin Edici
<b>BCO</b>	: Bulanık C-Ortalamalar
<b>GAİ</b>	: Gizli Anlamsal İndeksleme
<b>HKOK</b>	: Hata Karekök Ortalama Kare
<b>KO</b>	: K-Ortlamamalar
<b>KS</b>	: Kullanıcı Sayısı
<b>KTED</b>	: Kullanıcı Tabanlı Eşik Değer
<b>KTKEYK</b>	: Kullanıcı Tabanlı K En Yakın Komşu
<b>NOMH</b>	: Normalleştirilmiş Ortalama Mutlak Hata
<b>OF</b>	: Ortak Filtreleme
<b>OMH</b>	: Ortalama Mutlak Hata
<b>TBA</b>	: Temel Bileşenler Analizi
<b>TDA</b>	: Tekil Değer Ayrışımı
<b>ÜTKEYK</b>	: Ürün Tabanlı K En Yakın Komşu

## 1. GİRİŞ

İletişim teknolojilerinin gelişmesi birçok alanda günlük hayatı kolaylaştırmaktadır. Bilgisayar, akıllı telefon, tablet vb. elektronik ve İnternet ağına bağlı cihazlar sayesinde bilgiye ulaşmak ve iletişim kurmak tarihte hiç olmadığı kadar kolaydır. Her yaştan kişiler bu iletişim araçlarından birine kolaylıkla sahip olabilmektedir. İletişim teknolojilerinin erişilebilirliğinin artmasıyla İnternet ortamında sunulan hizmetlerde de artış görülmektedir. Özellikle son on yılda e-ticaret yapan işletmelerin sayısı hızla artmış ve insanların dikkati bu yöne çevrilmiştir [1]. Mağazalardan veya marketlerden satın alınabilecek hemen her ürün İnternet üzerinden satın alınabilir durumdadır.

Günümüzde hizmetlerini çevrimiçi olarak sunan birçok e-ticaret platformu bulunmaktadır. Bunlardan bazıları Booking.com, Amazon.com, Netflix.com, Last.fm, Hepsiburada.com, alibaba.com, n11.com gibi popüler e-ticaret platformlarıdır. Çevrimiçi hizmetler sunan servisleri kullanmak kullanıcılara her anlamda kolaylıklar sağlamaktadır. Kullanıcılar daha kısa sürelerde daha çok ürünü inceleme ve ürünleri birbirleriyle karşılaştırma imkânını bulabilmektedirler. Ürünler hakkında diğer kullanıcı yorumlarına bakarak bilmedikleri ürün hakkında fikir edinebilmek, çevrimiçi servislerin sunduğu kolaylıklar arasında sayılabilir.

Binlerce ürünü etkin biçimde kullanıcılarına sunabilmek çevrimiçi hizmet sunan platformlar için avantajdır, ancak artan ürün ve veri miktarı karşısında, kullanıcıların çok fazla veriyi incelemek zorunda kalması, istediği bilgiye hızlı ulaşmasını engelleyici bir faktör olarak değerlendirilmektedir. Bu problem “Aşırı Bilgi Yükü” olarak tanımlanır. İnternet üzerinden sunulan servislerdeki aşırı bilgi yoğunluğunda kullanıcılara ilgi duyabilecekleri ürünlere ulaşmalarına yardımcı olacak olacak öneri sistemleri geliştirilmiştir.

Kullanıcılar İnternet üzerinden sunulan servislerde aradıkları ürünün ne olduğunu bildiklerinde ilgili ürüne arama motorları sayesinde ulaşabilirler. Ayrıca servislerin sunulduğu ortamlardaki arama özellikleri sayesinde de aranılan ürünler kolaylıkla bulunabilir. Öneri sistemleri, onlarca ürün arasından ilk defa karşılaştığı ve hangi ürünü seçeceğine karar veremediği durumlarda kullanıcılara ürün seçme

aşamasında yardımcı olmak için geliştirilmiş sistemlerdir [2]. Yapılan aramalarda kullanıcıya arama ile ilgili her ürünün gösterilmesi yerine kullanıcının beğenebileceği ürünleri göstermekle arama performansı ve sistemin kullanılabilirliği iyileşmiş olur. Bu sayede sistemdeki kullanım kolaylığı ve kullanıcı memnuniyeti artabilir [3].

Öneri sistemleri hizmetlerin çevrimiçi sunulabildiği film satılan veya kiralanabilen, müzik dinlenebilen, video izlenebilen, kitap satın alınabilen, haber takip edilebilen, tatil paketi satın alınabilen platformlarda yaygın olarak kullanılmakta ve kullanımı günden güne artmaktadır [4]. Popüler çevrimiçi film izleme platformu olan Netflix üzerinde izlenen filmlerin %70'e yakını sunulan öneriler sonucu tercih edilmiştir [5]. Google News öneri sistemi sayesinde ziyaretçilerinin tıklama oranlarını %38 artırmıştır [5]. Amazon.com kullandığı öneri sistemi sayesinde satışlarını %35 artırmıştır [5]. Öneri sistemleri kullanıcılarına kişiselleştirilmiş ve otomatize edilmiş öneriler sunarlar. Televizyon ve radyo reklamları da öneri sistemi olarak düşünülebilir ancak kişiselleştirilmiş öneri sistemi olarak kabul edilemezler. Televizyonda veya radyoda yayınlanan reklamlar ilgili olan ya da olmayan kişiler tarafından izlenir.

Öneri sistemleri İçerik Tabanlı Filtreleme, Ortak Filtreleme ve Karma Sistemler olmak üzere üç ana grupta toplanabilir.

## 2. TEZİN AMACI

Ortak Filtreleme algoritmalarının deneysel çalışmalarında en çok dikkat edilen performans göstergesi doğruluk parametresidir. Sistem doğruluğunun ölçülmesi aşamasında Ortalama Mutlak Hata (OMH) en sık karşılaşılan doğruluk ölçütüdür. Bu ölçüte göre, algoritmanın ürettiği tahmin değerleriyle gerçek değerler arasındaki farklar hesaplanır. Bu farkların mutlak değerlerinin ortalaması alınarak doğruluk hesaplanır. Hesaplanan değer algoritmanın ortalama mutlak hatasıdır. Üretilen bazı tahmin değerleri gerçek değerlerden uzak olsa da diğer tahmin değerleri gerçek değerlere yakın olabileceğinden, gerçek değerlerden uzak olan tahminler görünmez olur. Benzer bir senaryo Tablo 2.1'de bir kullanıcı için üç farklı algoritma tarafından üretilen tahmin değerleri verilerek açıklanmıştır.

**Tablo 2.1.** Ortalama Mutlak Hata Hesabı

	ürün <sub>1</sub>	ürün <sub>2</sub>	ürün <sub>3</sub>	ürün <sub>4</sub>	ürün <sub>5</sub>	
Gerçek Değer	4	5	1	2	3	<b>OMH</b>
Algoritma-1	3,67	3,89	2,50	2,45	2,87	0,70
Algoritma-2	4,25	4,89	3,25	3,57	2,98	0,84
Algoritma-3	4,12	4,75	1,8	3,57	3,65	0,68

Algoritma-1 tarafından üretilen tahmin değerlerine bakıldığında *ürün<sub>1</sub>*, *ürün<sub>4</sub>*, *ürün<sub>5</sub>* için gerçeğe görece daha yakın sonuçların elde edildiği görülmektedir. Ancak *ürün<sub>2</sub>* ve *ürün<sub>3</sub>* için üretilen tahmin değerlerinin gerçek değerlerden uzak olduğu söylenebilir. Kullanılan algoritmaların birbirinden farklı yöntemler uygulamasından dolayı elde ettikleri hata değerleri de farklılık göstermektedir. Her algoritmanın ürettiği sonuçların ortalama mutlak hatası diğerlerinden farklı olacağından, bu sonuçlardan birinin diğerlerinden daha iyi olduğu sonucuna varılabilir. Dolayısıyla her kullanıcı için en iyi sonucu verecek bir algoritmanın var olduğu ve bir kullanıcının tercihlerine en yüksek doğrulukta tahmin üretebilecek algoritma tespit edilebilirse, sistem doğruluğunun da iyileşeceği sonucuna varılır.

Bu tezde amaçlanan Ortak Filtreleme öneri sistemleri tarafından üretilen tahminlerin doğruluğunu artırmaktır. Doğruluk bütün öneri sistemlerinde en

önemli performans göstergesi olarak sayılabilir. Dolayısıyla bu alanda yapılacak iyileştirmeler *OF* öneri sistemlerine önemli bir katkı yapabilir. Bu tezde, kullanıcı veya kullanıcı/ürün çiftleri için en iyi sonuç veren bir algoritmanın bulunduğu ve Ortak Filtreleme öneri sisteminde sabit bir algoritma kullanmak yerine, kullanıcı veya kullanıcı/ürün çiftleri için en uygun algoritmanın kullanılması durumunda sistem doğruluğunun artacağı öne sürülmektedir. Deneysel çalışmalar sonunda kullanıcı veya kullanıcı/ürün çiftleri için en iyi performans gösteren bir algoritmanın bulunduğu ve Ortak Filtreleme öneri sisteminde sabit bir algoritma kullanmak yerine kullanıcı veya kullanıcı/ürün çiftleri için en iyi sonuç veren algoritmanın kullanılması durumunda sistem doğruluğunun iyileşebileceğinin gösterilmesi amaçlanmıştır.



### 3. İÇERİK TABANLI FİLTRELEME ÖNERİ SİSTEMLERİ

İçerik Tabanlı Filtreleme (*ITF*) öneri sistemlerinde kullanıcı ve ürün profili oluşturulur ve kullanıcının hangi ürünleri veya hangi türden ürünleri sevebileceği tespit edilir. Sonra ürünler analiz edilerek her ürün hakkında ayrıntılı bilgi çıkarılır. İçerik tabanlı sistemlerinin başarısı, kullanıcıların ve ürünlerin özelliklerinin iyi bir şekilde ortaya çıkarılmasıyla mümkün olur [6]. Kullanıcıyı tanımlayabilmek için kullanıcı profili demografik veriler kullanılarak ayrıntılı biçimde tanımlanmalıdır. Kullanıcının yaşı, cinsiyeti, aylık gelir durumu, yaşadığı bölge, medeni hali, kişinin karakteristik özellikleri, nelerden hoşlandığı gibi bilgiler kullanıcıyı tanımak için kullanıcı profilinde mutlaka yer almalıdır. Kullanıcının profili kullanıcının ilgili olduğu ürünleri temsil eder [7]. Ürünlerin de aynı şekilde analiz edilmesi gereklidir, ürün eğer yenilebilir bir ürünse tadı, kokusu, rengi, şekli, ağırlığı vb. fiziksel özellikleri tam olarak ürün profilinde belirtilmelidir. Ürün giyilecek bir ürün ise rengi, bedeni, modeli, ağırlığı, ürünün kumaş cinsi vb. bilgiler ürün profilinde ayrıntılı olarak yer almalıdır. Ürün eğer bir kitap ise yazarı, dili, baskı kalitesi, sayfa sayısı, kitabın türü gibi bilgiler kitabın profilinin oluşturulması için gereklidir. Ürün eğer bir film ise konusu, yönetmeni, oyuncularını, filmin çekildiği mekân bilgileri, filmin dili, film başka dillere çevrildiyse seslendirmeleri yapan sanatçıların bilgileri ayrıntılı olarak profilinde tanımlanmalıdır. Bütün bu tanımlamalar yapıldıktan sonra artık öneri sisteminden öneri isteyen kullanıcılara kendi profilleriyle örtüşen ürünler önerilebilir [8].

Çevrimiçi hizmet sunan servislerde müşteri ve ürün sayısı milyonlarca olduğu düşünüldüğünde, ürünlerin ve kullanıcının profillerinin ortaya çıkarılması kolay değildir. Kullanıcı ve ürün profili oluşturma işi gerçek kişiler tarafından yapılacaksa ilgili kişilerin bu alanda uzman olması gerekir. Ayrıca kişilerin değerlendirmeleri çoğu zaman kendi kişisel görüşlerini yansıtacağından objektif değerlendirmeden uzak olabilir. Kullanıcı ve ürün profilleri bilgisayar tarafından oluşturulacaksa renk, koku ve hissetmeye dayalı bazı ürünlerde profil oluşturulması kolay olmayacaktır [7]. Bu sebeplerden dolayı *ITF* sistemlerinin ürün ve kullanıcı sayısının çok fazla olduğu sistemlerde kullanılması zordur.

#### 4. ORTAK FİLTRELEME ÖNERİ SİSTEMLERİ

Günlük hayatta insanlar verdikleri kararlarda çoğu zaman başkalarının önerilerini de dikkate alırlar. Özellikle kendilerine yakın olarak gördükleri kişilerin fikirlerini daha çok önemserler ve verdikleri kararlarda kendilerine yakın olan insanların görüşlerinden etkilenirler. Ortak Filtreleme (*OF*) sistemlerinin temelinde kabul edilen fikir şudur: Eğer iki kişi geçmişte benzer davranışlar göstermişse, örneğin geçmişte alışveriş yaparken, müzik dinlerken veya film izlerken aynı türden ürünleri tercih etmişlerse, gelecekte de aynı tercihlerde bulunabilirler [9]. *OF* algoritmaları, bilmediği bir ürün hakkında öneri isteyen bir kullanıcıya, ilgili kullanıcının geçmişteki tercihlerini ve kullanıcıya benzeyen diğer kullanıcıların tercihlerini birlikte değerlendirerek öneriler üretir [10]. Kullanıcılara daha önce hakkında fikir sahibi olmadığı veya oylamadığı ürünler için öneri üretilir. *OF* öneri sisteminden herhangi bir ürün için öneri isteyen kullanıcı daha önce ilgili ürüne oy vermişse, o ürün için öneri üretilmez [11].

*OF* alanında yapılan çalışmalar ilk olarak 1992 yılında başlamış ve 'Ortak Filtreleme' terimi Xerox Palo Alto araştırma merkezinde otomatik e-posta filtreleme sistemi olan Tapestry tasarımcıları tarafından ortaya atılmıştır [12]. *OF* sistemlerinde *İTF* sistemlerindeki gibi kullanıcı ve ürün profilleri oluşturulmaz, bunun yerine kullanıcıların tercihlerinin tutulduğu kullanıcı-ürün matrisleri kullanılır. Bu matris  $m$  adet kullanıcıdan  $\{k_1, k_2, k_3, \dots, k_m\}$  ve  $n$  tane üründen  $\{ü_1, ü_2, ü_3, \dots, ü_n\}$  oluşan  $[m \times n]$  kullanıcı-ürün matrisidir. Bu matristeki hücre değerleri, kullanıcıların ürünler hakkındaki beğeni seviyelerini gösterir. Kullanıcılar öneri sistemindeki herhangi bir ürünü doğrudan oylayabilirler ya da kullanıcıların sistemdeki davranışları değerlendirilerek ürün hakkındaki düşünceleri aşağıdaki yöntemlerden biriyle tespit edilebilir [13]:

- Kullanıcının ürünü satın alıp almama durumu,
- Kullanıcının ürün sayfasında geçirdiği süre,
- Kullanıcının ürün hakkındaki yorumu.

Kullanıcı-ürün matrisindeki boş olan alanlar kullanıcının ilgili ürün hakkında görüşünün olmadığı veya görüş belirtmek istemediği anlamına gelir.

*OF* sistemleri *İTF* sistemlerinde bahsedilen problemlerden daha az etkilenir [3]. *İTF* sistemlerinin başarımı iyi oluşturulmuş kullanıcı ve ürün profillerine bağlıdır. *OF* öneri sistemlerinde *İTF* öneri sistemlerindeki gibi kullanıcı veya ürün profilleri oluşturulmaz; bunun yerine Tablo 4.1'deki gibi kullanıcıların tercihlerinin tutulduğu bir kullanıcı-ürün matrisi yeterlidir.

**Tablo 4.1.** 5-Yıldızlı Oy Değerlikli Kullanıcı-Ürün Matrisi

	$\bar{u}_1$	$\bar{u}_2$	$\bar{u}_3$	$\bar{u}_4$	$\bar{u}_5$
$k_1$	4	1		2	2
$k_2$		3	4	5	1
$k_3$	3		4		5
$k_4$	1	2		3	5

**Tablo 4.2.** İkili Oy Değerlikli Kullanıcı-Ürün Matrisi

	$\bar{u}_1$	$\bar{u}_2$	$\bar{u}_3$	$\bar{u}_4$	$\bar{u}_5$
$k_1$	1	0		0	0
$k_2$		0	1	1	0
$k_3$	0		1		1
$k_4$	0	0		0	1

Kullanıcıların verdikleri oy değerleri ürünleri beğeni derecelerini gösterir, 5-yıldızlı oylama sistemini kullanan bir sistemde bir ürüne 5-yıldız veren bir kullanıcı ürünü çok beğenmiş, 1-yıldız veren kullanıcı ise ürünü beğenmemiş demektir. Ürüne verilen yıldız miktarı arttıkça ürünün beğenilme derecesi artmış olur. İkili oylama sistemini kullanan bir öneri sisteminde, bir kullanıcı ürüne oy değeri olarak 1 vermişse beğenmiş, 0 vermişse beğenmemiş demektir. İkili oylama sisteminde 0 ve 1'den başka oy yoktur. Kullanıcı-ürün matrisindeki boş alanlar, kullanıcıların ilgili ürün hakkında fikir beyan etmediğini gösterir. *OF* öneri sistemlerinde Tablo 4.1 ve Tablo 4.2'deki gibi kullanıcıların ürünler hakkındaki değerlendirmelerinin tutulduğu kullanıcı-ürün matrisi oluşturmak yeterlidir. *OF* öneri sistemlerinin çalışma yöntemi kullanıcılar arasındaki benzerlik esasına dayandığından, kullanıcılar arasındaki benzerlikler ne kadar yüksek doğrulukla hesaplanırsa üretilen tahminlerin doğruluğu da o oranda iyileşebilir. Kullanıcı-ürün matrisinin doluluk oranı kullanıcılar arasındaki benzerliklerin hesaplanmasını etkiler.

Kullanıcı-ürün matrisinin doluluk oranı yüksek ise kullanıcılar arasındaki benzerlik değeri daha doğru biçimde hesaplanabilir. Kullanıcılar arasında iyi hesaplanmış benzerlik oranı, üretilen tahmin değerlerinin doğruluklarını artırabilir. *OF* için oluşturulmuş kullanıcı-ürün matrisleri genellikle çok boşluklu yapıdadır [14].

*OF* öneri sistemlerinde kullanıcılar geçmişte benzer şekilde davranmışlarsa, aynı ürünleri beğenmiş veya beğenmemişlerse benzer kullanıcı olarak kabul edilir. İki kullanıcı arasında birbirine yakın şekilde değerlendirilen ürün sayısı ne kadar çoksa kullanıcı benzerliği o kadar yüksektir. Tablo 4.3' te verilen basit bir kullanıcı-ürün matrisinde birbirine çok benzeyen *A* ve *B* kullanıcılarının  $\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5$  ürünleri hakkındaki değerlendirmeleri verilmiştir. Kullanıcılar aynı ürünlere yakın oylar verdiklerinden birbirlerine benzedikleri kabul edilir.

**Tablo 4.3.** Basit Bir Kullanıcı-Ürün Matrisi

	$\bar{u}_1$	$\bar{u}_2$	$\bar{u}_3$	$\bar{u}_4$	$\bar{u}_5$
<i>A</i>	4	5	2	3	5
<i>B</i>	5	4	2	3	?

Tablo 4.3' te gösterildiği gibi *B* kullanıcısının  $\bar{u}_5$  ürününe vereceği oyu tahmin etmek için *A* kullanıcısı ile olan benzerliğine bakılır. *A* kullanıcısı ile *B* kullanıcısının ( $\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4$ ) ürünlerine verdikleri oy değerleri çok benzerdir. *A* kullanıcısı  $\bar{u}_5$  ürününe 5 verdiği için *B* kullanıcısının  $\bar{u}_5$  ürününe vereceği oyu 5'e yakın olması beklenir. Başka bir deyişle *A* kullanıcısı  $\bar{u}_5$  ürününü beğenmiştir ve *B* kullanıcısı da  $\bar{u}_5$  ürününü beğenebilir. Tablo 4.3'te çok benzer iki kullanıcı verilmiş ve tahmin üretme işlemi iki kullanıcı arasında gösterilmiştir, gerçek veri kümelerinde kullanıcı ve ürün sayıları çok daha fazladır. Bu durumda veri kümesindeki oy değerleri kullanılarak kullanıcılar arasındaki benzerlik değerleri hesaplanır ya da veri kümesindeki kullanıcı oy değerleri kullanılarak bir model oluşturulur. Hesaplanan benzerlik değerleriyle ya da oluşturulan modele göre tahmin değeri hesaplanır.

*OF* öneri sistemleri kullanıcılara öneri üretirken bu alanda önerilmiş algoritmalar kullanılır. Bu alanda 1992 yılından bu güne kadar önerilmiş birçok

algoritma vardır. *OF* öneri algoritmaları (i) hafıza tabanlı algoritmalar, (ii) model tabanlı algoritmalar ve (iii) karma algoritmalar olarak üç sınıfa ayrılır [15].

#### 4.1. Hafıza Tabanlı Algoritmalar

Hafıza tabanlı algoritmalar öneri üretme aşamasında kullanıcı-ürün matrisindeki bütün kullanıcı oylarını kullanırlar. Öneri tahminleme işlemi çevrimiçi yapılır. Ancak bu algoritmalar veri kümesinin tamamı üzerinde çalıştığından hafıza bağımlıdırlar. Hafıza tabanlı algoritmaların ürettiği tahmin değerlerinin doğruluğu model tabanlı algritmalara göre nispeten daha yüksektir [7]. Herhangi bir kullanıcı, hafıza tabanlı bir algoritma kullanarak öneriler üreten sistemden bir ürün için öneri istediğinde, diğer ürünlere verdiği oyları bir vektör olarak ele alınır. Kullanıcının bu vektörü, kullanıcı-ürün matrisindeki diğer tüm kullanıcıların vektörleriyle benzerlik hesaplama işlemine tabi tutularak benzer ya da komşu kullanıcılar belirlenir. Komşu kullanıcıların ilgili ürün için tercihleri üretilecek öneriyi belirler. Genel olarak aktif kullanıcı ile diğer kullanıcılar arasındaki benzerlik değerleri hesaplanır. Yakın kullanıcıların hedef ürüne verdikleri oylar, aktif kullanıcı ile olan benzerlikleriyle ağırlıklandırılarak tahminleme yapılır. Bu şekilde, aktif kullanıcının diğer kullanıcılarla olan benzerlik değerliklerini hesaplayarak öneri üreten sistemlere, komşuluk tabanlı öneri sistemleri de denir. Hafıza tabanlı algoritmalar, kullanıcılar arasındaki benzerlik esasına göre çalışıyorsa kullanıcı-tabanlı, ürünler arasındaki benzerlik esasına göre çalışıyorsa ürün-tabanlı algoritmalar olarak tanımlanır [10, 16, 17].

Kullanıcılar arasındaki benzerlikleri hesaplamak için kullanılan yöntemler genellikle Denklem 4.1.'de verilen Pearson korelasyon katsayısı [10] gibi veri noktaları arasındaki ikili ilişkiyi veya Denklem 4.2.'de verilen vektör benzerlik ölçütü gibi veri noktaları arasındaki geometrik açının büyüklüğünü esas alır [18]. Pearson korelasyon katsayısı kullanıcılar arasındaki benzerliği hesaplama ölçütü olarak kullanıldığında, kullanıcılar arasındaki benzerlik değerleri [-1,+1] aralığında reel bir sayı olur, '+1' kullanıcıların tamamen benzeştiğini, '-1' ise kullanıcıların tamamen zıt karakteristiğe sahip kişiler olduğunu gösterir. Benzerlik değerinin '0' olması kullanıcılar arasında herhangi bir bağlantı olmadığını gösterir. Vektör benzerlik ölçütü kullanıldığında, kullanıcılar arasındaki benzerlik değeri yine

[-1,+1] arasında reel değerlerden oluşur. Vektör benzerlik ölçütü, kullanıcıları vektör olarak kabul eder. Vektörler arasındaki açının kosinüs değeri hesaplanır. İki vektör arasındaki açının kosinüsü '1' olursa, bu vektörler aynı yöne bakıyor demektir. Kullanıcıların vektör olduğu düşünüldüğünde, kullanıcıların da "aynı yöne bakan" ya da "benzer şekilde kararlar veren kullanıcılar" oldukları kabul edilir. Vektör benzerlik değeri '-1' olursa, bu vektörler tamamen birbirinin tersi yönündedir. Aynı çıkarımla, kullanıcılar arasındaki vektör benzerlik değeri '-1' olduğunda, bu kullanıcıların verdikleri kararlarda birbirlerinin tersi yönünde hareket edecekleri düşünülebilir.

$$benz(a, b) = \frac{\sum_{\bar{u} \in \bar{U}} (R_{\bar{u},a} - \bar{R}_a)(R_{\bar{u},b} - \bar{R}_b)}{\sqrt{\sum_{\bar{u} \in \bar{U}} (R_{\bar{u},a} - \bar{R}_a)^2 \sum_{\bar{u} \in \bar{U}} (R_{\bar{u},b} - \bar{R}_b)^2}} \quad (4.1)$$

$$benz(a, b) = \sum_j \frac{R_{a,\bar{u}}}{\sqrt{\sum_{\bar{u} \in \bar{U}} R_{a,\bar{u}}^2}} \frac{R_{b,\bar{u}}}{\sqrt{\sum_{\bar{u} \in \bar{U}} R_{b,\bar{u}}^2}} \quad (4.2)$$

Denklem 4.1. ve Denklem 4.2.'de  $a$  ve  $b$  kullanıcıları;  $\bar{u}$ ,  $a$  ve  $b$  kullanıcıları tarafından oylanan ürünü;  $\bar{U}$ ,  $a$  ve  $b$  kullanıcısı tarafından oy verilmiş ürün kümesini;  $R_{\bar{u},a}$  ve  $R_{\bar{u},b}$ ,  $a$  ve  $b$  kullanıcılarının sırasıyla  $\bar{u}$  ürününe verdiği oy değerini;  $\bar{R}_a$  ve  $\bar{R}_b$ ,  $a$  ve  $b$  kullanıcılarının verdikleri oyların ortalamalarını gösterir.

Kullanıcıya öneri üretmek için Denklem 4.3 kullanılır. Aktif kullanıcının sistemdeki diğer kullanıcılarla olan benzerlikleri hesaplandıktan sonra her kullanıcının benzerlikleri, hedef ürüne verdikleri oyla çarpılarak ağırlıklandırılmış tahmin değeri hesaplanır ve aktif kullanıcıya sunulur.

$$\bar{O}_{a,\bar{u}} = \frac{\sum_{b \in N} benz(a, b)(R_{b,\bar{u}})}{\sum_{b \in N} |benz(a, b)|} \quad (4.3)$$

Denklem 4.3'te  $a$  aktif kullanıcıyı;  $\bar{u}$  aktif kullanıcının hakkında öneri istediği hedef ürünü;  $N$  aktif kullanıcının hakkında öneri istediği  $\bar{u}$  ürününe oy vermiş

kullanıcı kümesini;  $benz(a,b)$  Denklem 4.1 veya Denklem 4.2 yardımı ile hesaplanan kullanıcı benzerliğini;  $R_{b,\tilde{u}}$   $\tilde{u}$  ürününe oy veren  $b$  kullanıcısının oy değerini gösterir.

**Örnek:** Tablo 4.4'teki kullanıcı-ürün matrisine göre  $k_a$  kullanıcısının  $\tilde{u}_5$  ürünü için beğeni değerini hesaplayalım.

**Tablo 4.4.** Örnek Kullanıcı-Ürün Matrisi

	$\tilde{u}_1$	$\tilde{u}_2$	$\tilde{u}_3$	$\tilde{u}_4$	$\tilde{u}_5$
$k_1$	4	1	1	2	2
$k_2$	4	3	1	-	-
$k_3$	3	2	-	5	5
$k_4$	4	-	2	1	5
$k_a$	4	1	1	3	?

$k_a$  kullanıcısı ile diğer kullanıcılar arasındaki benzerlik değerleri Pearsons korelasyon katsayısı ile hesaplandığında aşağıdaki benzerlik değerleri bulunur.

$$benz(k_a,k_1)=0,94 \quad benz(k_a,k_2)=0,76 \quad benz(k_a,k_3)=0,50 \quad benz(k_a,k_4)=0,50$$

Benzerlik değerleri hesaplandıktan sonra, kullanıcılar arasından  $\tilde{u}_5$  ürününü oylayan kullanıcıların verdikleri oylar benzerlik değerleriyle çarpılarak ağırlıklandırılmış tahmin değeri hesaplanır. Burada  $k_2$  kullanıcısı  $\tilde{u}_5$  ürününe oy vermediğinden hesaba katılmaz. Denklem 4.3 kullanılarak aktif kullanıcı için  $\tilde{u}_5$  ürünü hakkındaki öneri değeri aşağıdaki şekilde hesaplanır.

$$\tilde{O}_{a,\tilde{u}} = \frac{\sum_{b \in N} benz(a,b)(R_{b,\tilde{u}})}{\sum_{b \in N} |benz(a,b)|} = \frac{0,94 * 2 + 0,50 * 5 + 0,50 * 5}{0,94 + 0,50 + 0,50} = 4,35$$

Öneri sistemi  $k_a$  kullanıcısına  $\tilde{u}_5$  ürünü hakkında 4,35 değerini üretmiştir. Bu işlem sonunda üretilen 4,35 değeri  $k_a$  kullanıcısının  $\tilde{u}_5$  ürününü beğenebileceğini gösterir. Burada gösterilen dört kullanıcıdan oluşan bir *OF* öneri sistemidir, gerçekte öneri sistemleri binlerce kullanıcı ve üründen oluşan kullanıcı-ürün matrisi kullanırlar.

## 4.2. Model Tabanlı Algoritmalar

Hafıza tabanlı algoritmaların tahmin üretme işleminde kullanıcılar arasında komşuluk hesaplanmalıdır. Kullanıcıların ortak oyladıkları ürün yoksa komşuluk hesaplanamaz ve tahmin üretilemez. Hafıza tabanlı algoritmaların bu eksikliğini

giderebilmek için model tabanlı algoritmalar geliştirilmiştir. Model tabanlı algoritmalar, veri setinin tamamını kullanarak bir model oluşturur ve tahminler bu model kullanılarak üretilir. Hafıza tabanlı algoritmalar çevrimiçi çalışıklarından öneri üretme süreleri daha uzundur. Hafıza tabanlı algoritmalar daha kolay bir şekilde kurulabilir ve doğrulukları genelde daha yüksektir. Ancak boşluklu veri kümelerinde model tabanlı algoritmalar daha iyi sonuçlar verir. Hafıza tabanlı algoritmalarda, aktif kullanıcıya öneri üretirken tüm veri kümesi kullanıldığından ve bütün süreçler aynı anda gerçekleştirildiğinden çevrimiçi performansları düşüktür. Model tabanlı algoritmaların model üretme işlemi çevrimdışı, tahmin üretme süreci çevrimiçi yapılıdır. Bu durumda model tabanlı algoritmalar çok daha hızlı çalışırlar [18]. Model tabanlı algoritmalarda model güncellemesi sürekli yapılmadığından sisteme yeni katılan kullanıcıların oy değerleri öneri sistemine hemen eklenmez ve tahmin üretme işlemine hemen dâhil olmaz. Literatürde model tabanlı birçok algoritma önerilmiştir. Tekil değer ayrışımı (TDA), gizli anlamsal indeksleme (GAI), temel bileşenler analizi (TBA) gibi boyut indirgeme teknikleri tabanlı *OF* yaklaşımları, matris çarpım tabanlı, sınıflandırma tabanlı, faktör analiz tabanlı algoritmalar geliştirilmiştir [19, 20, 21, 22, 23, 24].

### **4.3. Ortak Filtreleme Öneri Sistemlerinin Problemleri**

Doğruluk, bir öneri sisteminin gerçek değerlere ne kadar yakın tahminler ürettiğini gösterir [25]. Çevrimiçi hizmet sunan herhangi bir platformda kullanıcının beğenebileceği ürünler önerilmelidir. Kullanıcıya önerilen ürün kullanıcı tarafından beğenilmediği zaman kullanıcının platforma olan güveni azalabilir. Bu nedenle doğruluğun öneri sistemleri için önemi yüksektir.

Kullanıcı-ürün matrisinin boşluklu yapısı, *OF* öneri sistemleri için önemli bir problemdir. *OF* öneri sistemleri, kullanıcıların önceden sistemde tutulan tercihlerinin bulunduğu kullanıcı-ürün matrisini kullanarak öneri üretir. *OF* öneri sistemlerinin kullanıcılara öneri sunabilmesi için kullanıcılar arasında benzerlik hesaplanması veya kullanıcı-ürün matrisi kullanılarak model oluşturulması gereklidir. *OF* öneri sisteminin öneri isteyen kullanıcıya doğruluğu yüksek öneriler üretebilmesi için, kullanıcılar arasındaki komşulukların iyi hesaplanabilmesi gerekir. Aynı şekilde oluşturulan modellerin de öneri üretilebilecek şekilde olması



için kullanıcı-ürün matrisinin doluluk oranının yüksek olması gerekir. Kullanıcılar genelde binlerce ürün arasından çok az sayıda ürüne oy verirler. Bu durum da oluşan veri kümelerinin boşluklu yapıda olmasına neden olur [17].

Soğuk-Başlangıç problemi [26] doluluk oranı düşük olan boşluklu yapıdaki kullanıcı-ürün matrisleriyle oluşturulmuş *OF* öneri sistemlerinin problemidir. Boşluklu yapıdaki veri kümelerinde, kullanıcılar arasında komşuluk oluşturmak bazen mümkün olamamaktadır. *OF* öneri sisteminden öneri isteyen kullanıcıya öneri üretilebilmesi için kullanıcının sistem tarafından tanınması gerekir. Kullanıcının önceden sistemde hiç oyu yoksa bu kullanıcı *OF* öneri sistemi tarafından tanınmaz ve kullanıcıya hiçbir ürün için öneri üretilemez. Sisteme yeni eklenen ürünleri hiçbir kullanıcı değerlendirmemişse, bu ürünler de kimseye önerilemez. Kullanıcıya öneri üretilememe durumuna kullanıcı soğuk başlangıç problemi, ürünü hiçbir kullanıcıya önerememe durumuna da ürün soğuk başlangıç problemi denir [27, 28]. Kullanıcı soğuk başlangıç problemi, sisteme yeni kullanıcı eklendiğinde; ürün soğuk başlangıç problemi de sisteme yeni ürün eklendiğinde oluşur.

Kızırtırma atakları [29, 30] *OF* öneri sistemlerinin başka bir problemidir. Öneri sistemindeki bazı ürünler hakkında gerçek dışı değerlendirmeler yapan kötü niyetli kullanıcılar, öneri sonuçlarını değiştirmeyi amaçlarlar. Bu durumda ya önerilmesi gereken ürünler kullanıcılara önerilemez, ya da önerilmemesi gereken ürünler kullanıcılara önerilir. Sonuç olarak *OF* öneri sistemi güvenilirliğini yitirir.

Gizlilik *OF* öneri sisteminin önemli problemlerinden biridir. Kullanıcıların aldıkları ürünler, okudukları kitaplar, izlediği filmler gibi öneri sistemlerine sundukları beğeni geçmişleri, kullanıcılar için mahrem ve başka kişiler tarafından bilinmesi istenmeyen bilgiler olarak değerlendirilir. Bu bilgilerin veriyi toplayan öneri sistemi tarafından istismar edilmesi ya da kötü niyetli kişilerin eline geçmesi durumunda sakıncalı sonuçlar doğabilir. Kullanıcılar da bilgilerinin güvende olmadığı, başka kişilerin eline geçeceği endişesi taşarsa ilgili platformu kullanmaktan çekinebilirler. Bu tür çekinceleri ortadan kaldırmak ve kullanıcı bilgilerinin gizliliğini ihlal etmeyecek şekilde kayıt altında tutmak için bazı yöntemler önerilmiştir [31, 32].

## 5. KARMA TAVSİYE SİSTEMLERİ

Karma (Hibrit) öneri sistemleri *İTF* öneri sistemleriyle *OF* öneri sistemlerinin avantajlı olan yönleri kullanılarak geliştirilmiş öneri sistemleridir. *İTF* ve *OF* sistemlerinin hem avantajlı hem de dezavantajlı yönleri vardır. *OF* öneri sistemlerinde kullanıcıya öneri sunabilmek için veri kümesinde benzer kullanıcılar bulmak gerekir. Bazen benzer kullanıcı bulunamadığı durumlarda öneri üretilemez. Böyle durumda *İTF* tabanlı öneri sistemi kullanılabilir, kullanıcının *İTF* profiline göre ürünler önerilebilir. Aynı şekilde kullanıcı veya ürün profili oluşturulması zor olan durumlarda kullanıcı veya ürünler için komşuluk hesaplanarak, *OF* öneri sistemlerinin avantajları kullanılarak öneri üretilebilir. Bu alanda önerilen bazı yaklaşımlar vardır. Pennock ve ark. [33] kişisel tanılama olarak bilinen hafıza ve model tabanlı *OF* öneri sistemini birleştiren bir yaklaşım önermişlerdir. Chee ve ark [34] RecTree olarak bilinen hibrit bir algoritma önermişlerdir. Önerilen algoritma ağaç yapısında bir sınıflandırma oluşturur. Tahmin değerleri ağırlıklandırılmış ortalama hesabına göre üretilir. Rashid ve ark. [35] sınıflandırma ve en yakın  $K$  tane komşu tabanlı hibrit bir yaklaşım önermişlerdir. Acilar ve Arslan [36] yapay bağışıklık ağı tabanlı (aiNet) *OF* algoritması önermişlerdir. Bu algoritma aiNet modelini kullanarak veri miktarını azaltır. Kullanıcıları sınıflandırdıktan sonra hafıza tabanlı *OF* yaklaşımıyla öneriler üretir. Ju ve Xu [37] yapay arı kolonisi algoritmasını kullanan *OF* tabanlı en iyi  $N$  tane ürün öneren hibrit bir algoritma önermiştir. Cheng ve ark. [38] kullanıcının dinamik bilgilerini kullanan hibrit bir algoritma önermişlerdir.

## 6. VERİ KÜMELERİ

*OF* öneri sistemlerinde kullanıcıların tercihlerinin tutulduğu veri kümeleri kullanılır, bu veri kümelerinde kullanıcıların ürünler hakkındaki değerlendirmeleri tutulur. Veri kümeleri genel olarak [kullanıcı sayısı×ürün sayısı] biçiminde matris şeklindedir. *OF* öneri sistemleri için oluşturulmuş birçok veri kümesi bu alanda araştırma yapan araştırmacılar tarafından kullanılmaktadır [39]. Tablo 6.1’de *OF* öneri sistemlerinde araştırmacılar tarafından en çok kullanılan gerçek veri kümelerinden bazıları gösterilmiştir.

**Tablo 6.1.** Veri kümeleri

Veri Kümesi	Kullanıcı Sayısı	Ürün Sayısı	Oy sayısı	Doluluk Oranı
Netflix	480.189	17.770	95.947.878	%1,1248
MovieLens	6.040	3.952	1.000.000	%4,1893
EachMovie	72.916	1.682	2.811.983	%2,2928
BookCrossing	278.858	271.379	1.149.780	%0,0015

Tablo 6.1’de veri kümelerinin doluluk oranlarının çok düşük olduğu görülmektedir, bu da veri kümelerinin çok boşluklu bir yapıda olduğu anlamına gelir. Veri kümelerinin doluluk oranları arttıkça *OF* sistemlerinin ürettikleri tahminlerin doğruluğunun artması beklenir. Veri kümesinin boşluklu yapıda olması doğruluğu düşürebileceği gibi bazı kullanıcılara da *OF* sistemi kullanarak tahmin üretilemez [17]. Bu sorunların üstesinden gelebilmek için farklı kuruluşlar kendi oluşturacakları veri kümelerini bir araya getirerek daha dolu veri kümesi elde edebilir ve elde ettikleri ortak veri kümesi üzerinden kullanıcılarına daha doğru tahminler sunabilirler [40].

## 7. DEĞERLENDİRME ÖLÇÜTLERİ

*OF* öneri sistemleri doğruluk ve çevrimiçi performans gibi yönlerden değerlendirilebilir [27, 60]. Öneri sistemlerinin doğruluğunu ölçmekten kastedilen, algoritmaların tahmin ettikleri değerlerin gerçek değerlere ne kadar yakın olduğunu ölçmektir. Ortaya çıkan hata değeri üretilen tahminlerin gerçek değerlerle arasındaki farkların mutlak değerinin ortalamasıdır. *OF* öneri sisteminin ürettiği önerilerin doğruluğunu ölçmek için kullanılan bazı ölçütler; Ortalama Mutlak Hata (*OMH*), Ortalama Kare Hata (*OKH*), Ortalama Kare Karekök Hata (*HKOK*), Normalleştirilmiş Ortalama Mutlak Hata (*NOMH*) gibi değerlendirme ölçütleridir. Bu tez kapsamında kullanılan değerlendirme ölçütleri aşağıda açıklanmıştır.

**Ortalama Mutlak Hata (OMH):** En çok kullanılan değerlendirme ölçütüdür. Hesaplanan tahmin değerleriyle gerçek değerler arasındaki farkın mutlak değeridir. *OMH* Denklem 7.1'de gösterilen formülle hesaplanır.

$$OMH = \frac{1}{N} \sum_{i=1}^{i=N} |p_i - r_i| \quad (7.1)$$

**Ortalama Karekök Kare Hata (HKOK):** Gerçek değerler ile hesaplanan tahmin değerleri arasındaki farkın karelerinin toplamının ortalamasının kareköküdür. Denklem 7.2.'de gösterilen şekilde hesaplanır.

$$HKOK = \sqrt{\frac{1}{N} \sum_{i=1}^{i=N} (p_i - r_i)^2} \quad (7.2)$$

**Normalleştirilmiş Ortalama Mutlak Hata (NOMH):** *OMH* doğruluk ölçütünün normalleştirilmiş halidir. *OF* öneri sistemlerinde kullanılan veri kümesinde kullanıcılardan toplanan oy değerleri 5 yıldızlı, ikili veya daha farklı aralıklarda olabilir. Farklı değerlikteki oylardan oluşturulmuş veri kümeleri kullanılarak yapılan deney sonuçlarını karşılaştırabilmek için *NOMH* kullanılır. Denklem 7.3.'deki gibi *OMH*'nin öneri sisteminde kullanılan veri kümesindeki en büyük oy ile en küçük oy arasındaki farka bölünmesiyle hesaplanır.

$$NOMH = \frac{OMH}{r_{max} - r_{min}} \quad (7.3)$$

Denklem 7.1 ve Denklem 7.2'de;  $p_i$  üretilen tahmin değerini,  $r_i$   $k$  kullanıcısının  $i$  ürününe verdiği gerçek oy değerini,  $N$ ,  $k$  kullanıcısı için üretilen toplam tahmin sayısını gösterir. Denklem 7.3'te  $r_{max}$  veri kümesindeki en büyük oy değerini  $r_{min}$  veri kümesindeki en küçük oy değerini gösterir.  $OMH$  ve  $HKOK$  değerleri ne kadar küçükse algoritmanın ürettiği tahmin değerleri gerçek değerlere o kadar yakın olduğu anlamına gelir.



## 8. İLGİLİ ÇALIŞMALAR

Tapestry, Goldberg ve ark. [12] tarafından 1992 yılında *OF* alanında önerilmiş ilk otomatik olmayan öneri sistemidir. Bu sistemde şirket çalışanları şirkete gelen bütün e-postaları okumak yerine yaptıkları sorgulamalar neticesinde kendilerine dönen e-postaları okuyarak aşırı bilgi yükünden kurtulmuşlardır. Bu sistemin sağladığı kolaylıklar fark edilince otomatik ortak filtreleme sistemleri geliştirilmiştir. Resnick ve ark. [41] Usenet makalelerindeki birbiri ile ilgili yorumları otomatik olarak tespit edip bir araya getirerek öneriler sunabilen *OF* öneri sistemi önermiştir. Usenet makalelerinin okunma süreleri ile bu makalelere verilen oy değerleri arasında bir ilişki olduğunu gösteren Konstan ve ark. [42] bu ilişkiyi kullanarak öneriler önerilebileceğini göstermişlerdir. Bu çalışmalardan sonra *OF* öneri sistemlerinin değişik birçok alanda kullanılabilmesi ve kullanıcılara öneri üretme konusunda fayda sağlayabileceği düşünülmüştür. Kullanıcılara müzik CD'lerinin önerilmesi amacıyla Ringo müzik öneri sistemi Shardanand ve Maes [43] tarafından kullanıma sunulmuştur. Müzik CD öneri sistemine benzer şekilde Hill ve ark. [44] BellCore Video Öneri sistemi adıyla kullanıcılara film önermek için bir öneri sistemi geliştirmişlerdir. Film ve müzik öneri sisteminden başka "Jester" olarak bilinen 2001 yılında Goldberg ve ark. [20] tarafından geliştirilmiş, kullanıcılara yüz tane fıkra arasından ilgilerini çekebilecek fıkraların önerilmesi amacıyla tasarlanmış fıkra öneri sistemi geliştirilmiştir. Herlocker ve ark. [45] yaptıkları çalışmada algoritmik bir çerçeve sunmuşlar ve bu çerçevede, benzerlik, önem ve varyans ağırlıklandırarak, komşuluk seçimi ve oy normalizasyon teknikleriyle doğruluk ve performans iyileşmesi sağlamışlardır. Liu ve ark. [46] tarafından önerilen yaklaşımda kullanıcılar arasında benzerlik hesabı yapılırken Pearson ve kosinüs benzerliğinde başka bir yöntemle kullanıcıya sistemde çok az oyu bile olsa yüksek doğruluk ve performansta öneriler üretilebilmektedir. Chen ve He [47] soğuk başlangıç problem üzerinde çalışmış ve kullanıcı benzerliklerini kullanıcıların demografik özelliklerini kullanarak hesaplamış ve bu sayede üretilen önerilerin kalitesini artırmıştır. Wang ve ark. [48] benzerlik füzyonu (similarity fusion SF) isimli, hem kullanıcı tabanlı hem de ürün tabanlı *OF* öneri sistemlerine uygun bir algoritma önermişlerdir. Bu algoritma boşluklu yapıdaki veri kümelerinde de yüksek doğrulukta öneriler üretebilmektedir. Bell ve Koren [49] en

yakın  $k$  tane komşu algoritmasını enterpolasyon ağırlıklandırma ile birleştirmişlerdir. Bu yöntem enterpolasyon tabanlı *OF* olarak adlandırılır. Bu yöntemle algoritmanın çalışma zamanında kötüleşme olmadan doğruluk iyileşmektedir. Ma ve ark. [50] Pearsons korelasyon katsayısına bir parametre ekleyerek kullanmışlar ve etkin bir tahminleme algoritması önermişlerdir. Luo ve ark. [51] yerel ve global kullanıcı benzerlik yöntemini birleştirerek hafıza tabanlı *OF*'de iyileşme sağlamışlardır. Bu yöntem, LU&GU öneri metodu olarak bilinir. Polatidis ve Georgiadis [52] *OF* önerilerinin kalitesini artıran dinamik, çok düzeyli bir ortak filtreleme yöntemi önermişlerdir. Hamidreza ve Kouros [53] kullanıcı tabanlı *OF* için bulanık C ortalamalar yaklaşımı önermişlerdir. Önerilen yöntemle *OF* öneri sisteminin doğruluğunu iyileştirmişlerdir. Kaleli [54] entropi tabanlı komşuluk seçme yöntemi önermiş ve bu yöntemle *OF* önerilerin doğruluğunu artırmıştır. Breese ve ark. [19] farklı birçok *OF* algoritmalarının doğruluk değerlerini karşılaştırmıştır. Vozalis ve Margaritis [55] kullanıcı ve ürün tabanlı *OF* algoritmalarını karşılaştırmıştır. Candillier [8] farklı *OF* algoritmalarını karşılaştırarak en iyi kullanıcı tabanlı, ürün tabanlı ve model tabanlı bir algoritma ortaya çıkarmak için çalışmıştır. Huang ve ark. [56] altı tane *OF* algoritmasını e-ticaret ortamları için karşılaştırmıştır. Algoritmaların güçlü ve zayıf yönlerini ortaya koymuş ve farklı veri kümelerinde nasıl çalıştıklarını analiz etmiştir. Lee ve ark. [39] hem klasik hem de alanında iyi sayılan *OF* algoritmalarını ürün sayısı, kullanıcı sayısı, veri kümesi doluluk oranı ve performans gibi parametreler yönünden karşılaştırmışlardır. Hangi algoritmanın hangi durumda daha iyi çalıştığını tespit etmeye çalışmışlardır. Cacheda ve ark. [7] farklı *OF* algoritmalarının özelliklerini incelemiş algoritmaları birbirleriyle karşılaştırıp her birinin güçlü ve zayıf yönlerini ortaya çıkarmışlardır.

Literatürü taradığımızda, bulabildiğimiz *OF* tabanlı öneri sistemleri genellikle tek bir algoritma üzerine kurulmakta ve öneri üretme işlemi bütün kullanıcılar için aynı algoritma tarafından yapılmaktadır. *OF* öneri sistemlerinde doğruluk iyileştirmede kullanılan yaklaşımlar genellikle var olan bir algoritmanın iyileştirilmesi şeklinde, yeni bir algoritma önermek suretiyle ya da algoritmaların kullandıkları yöntemleri iyileştirerek doğruluk iyileştirme şeklinde karşımıza çıkmaktadır.

Bu tezde literatürde daha önce kullanılmayan bir yöntem araştırılmıştır. *OF* öneri sisteminde kullanıcı veya kullanıcı/ürün çifti için tek bir algoritma kullanmak yerine altı farklı algoritma arasından en iyi sonuç veren algoritmanın var olup olmadığı, eğer varsa en iyi sonuç veren algoritmanın seçilmesi durumunda üretilen önerilerin doğruluğunun hangi oranda iyileşebileceği araştırılmıştır.





## 9. ALGORİTMİK VE DENEYSEL ÇERÇEVE

### 9.1. Kullanılan Veri Kümeleri

Deneylerde iyi bilinen iki farklı gerçek veri kümesi kullanılmıştır. Bunlardan biri GroupLens (<http://movielens.umn.edu/>) tarafından oluşturulmuş MovieLens Million (MLM) veri kümesidir. Bu veri kümesinde 5 yıldızlı oy değerlerinden oluşan, 3.952 ürün için 6.040 kullanıcının oyu vardır. Kullanılan ikinci veri kümesi Netflix (<http://www.netflixprize.com/>) veri kümesidir. Netflix veri kümesinde de 5 yıldızlı oy değerlerinden oluşan, 17.770 ürün için 480.189 kullanıcının oyu vardır. Netflix veri kümesi *OF* algoritmalarının işleyemeyeceği kadar büyük bir veri kümesidir [7]. Deneylerde katmanlı örneklem (stratified sampling) yöntemiyle Netflix veri kümesinin özelliklerini taşıyan 10.000 kullanıcı ve 4.000 üründen oluşan bir altküme seçilmiştir. Seçilen bu alt küme Netflix veri kümesiyle yaklaşık olarak aynı özellikleri taşır. Netflix veri kümesinin doluluk oranı, oyların oranı ile seçilen alt kümenin doluluk oranı ve altkümedeki oyların oranı Tablo 9.1’de gösterilmiştir. Her iki veri kümesinde de yapılan deneyler aynı sonucu vermiş ve ortaya konan yaklaşımın veri kümesinden bağımsız olduğu gösterilmiştir.

**Tablo 9.1.** *Netflix Veri Kümesinininden Seçilen Altküme*

	Neflix	Seçilen Alt Küme
Doluluk Oranı	% 1,18	% 1,17
1 oranı	% 4,59	% 4,51
2 oranı	% 10,08	% 10,09
3 oranı	% 28,67	% 29,08
4 oranı	% 33,58	% 33,64
5 oranı	% 23,05	% 22,66

### 9.2. Kullanıcı Oylarının z-Puanı Normalizasyonu

MovieLens ve Netflix veri kümelerindeki kullanıcıların oyları [1,5] arasında ayırık değerlerden oluşmaktadır. Bu kısımda anlatılan deneysel çalışmalarda kullanılan test ve eğitim kümelerindeki bütün kullanıcı oyları z-puanına çevrilerek normalize edilmiştir. z-puanı normalizasyonu yapılan oy değerleri üzerinden

yapılan hesaplamalarda doğruluk iyileşmektedir [45]. Herhangi bir  $k$  kullanıcısının herhangi bir  $j$  ürünü hakkındaki oyu  $v_{kj}$ , kullanıcının tüm oylarının ortalaması  $\bar{v}_k$  ve kullanıcının oylarının standart sapması  $\sigma_k$  olursa  $z$ -puanı ( $z_{kj}$ ) Denklem 9.1'deki gibi hesaplanır.

$$z_{kj} = \frac{v_{kj} - \bar{v}_k}{\sigma_k} \quad (9.1)$$

Deneysel çalışmaların tamamında test ve eğitim kümelerindeki kullanıcı oyları Denklem 9.1. deki gibi  $z$ -puanı değerlerine dönüştürülmüş ve benzerlik hesabında ve tahmin üretme sürecinde  $z$ -puanı kullanılmıştır. Kullanıcılar arasındaki benzerlik hesabı kullanıcıların  $z$ -puanı değerleri çarpılarak hesaplanır. Denklem 4.1'de Pearson korelasyon formülüne bakıldığında kullanıcılar arasındaki benzerlik hesabının, kullanıcıların  $z$ -puanı değerlerinin çarpımı şeklinde olduğu görülebilir.

### 9.3. Kullanılan Değerlendirme Ölçütleri

Deneylerde hem kullandığımız algoritmaların doğruluğunu hem de ortaya konulan yaklaşımın geçerliliğini göstermek için *OMH* ve *HKOK* kullanıldı. Kullanılan veri kümelerindeki oy değerleri aynı değerlerden oluştuğu için normalleştirilmiş değerlendirme ölçütü kullanmaya gerek görülmedi.

### 9.4. Kullanılan Algoritmalar

Deneylerde altı farklı *OF* öneri algoritması kullanıldı. Kullanılan algoritmalar *OF* öneri sistemlerinde en çok kullanılan, doğrulukları iyi olarak kabul edilen algoritmalar. *OF* öneri sistemlerinde kullanılan algoritmalar, hafıza tabanlı, model tabanlı ve hibrit algoritmalar olmak üzere üç grupta incelenir. Seçilen altı algoritma her gruptan iki tane olacak şekilde seçilmiştir. Seçilen algoritmalar Tablo 9.2'de verilmiştir.

**Tablo 9.2.** *Deneylerde Kullanılan Algoritmalar*

Algoritma Sınıfı	Algoritma Adı
Hafıza Tabanlı	Kullanıcı Tabanlı K En Yakın Komşu (KTKEYK)
	Kullanıcı Tabanlı Eşik Değer (KTED)
Model Tabanlı	Ürün Tabanlı K Yakın Komşu (ÜTKEYK)
	Ağırlıklandırılmış Bir Eğimli Tahmin Edici (ABETE)
Hibrit	Bulanık C-Ortalamalar (BCO)
	K-Ortalamalar (KO)

### 9.4.1. Hafıza tabanlı algoritmalar

#### 9.4.1.1. *Kullanıcı tabanlı K en yakın komşu*

*KTKEYK* algoritması *OF* öneri sistemlerinde kullanılan popüler hafıza tabanlı algoritmadır. Algoritma komşuluk hesabı üzerine çalışır. Bu kısımda *OF* öneri sisteminden öneri almak için gelen kullanıcı aktif kullanıcı, aktif kullanıcının, hakkında öneri istediği ürün hedef ürün olarak isimlendirilecektir. *OF* öneri sisteminden hedef ürün için öneri isteyen aktif kullanıcının öncelikle sistemdeki diğer kullanıcılarla olan benzerlikleri hesaplanır. Pearson Korelasyon katsayısı popüler benzerlik hesaplama yöntemidir. Denklem 4.1' deki formülle aktif kullanıcı ile sistemde var olan diğer bütün kullanıcılar arasındaki benzerlik değerleri hesaplanır. Benzerlik değerleri hesaplandıktan sonra aktif kullanıcıya en çok benzeyen  $K$  tane kullanıcı belirlenir. Aktif kullanıcıya en çok benzeyen  $K$  tane kullanıcının hedef ürüne verdikleri oy değerleri aktif kullanıcıyla olan benzerlik değerleriyle ağırlıklandırılarak toplanır ve aktif kullanıcıyla olan benzerlik değerlerinin mutlak değerlerinin toplamına bölünür. Ortaya çıkan bu sonuç aktif kullanıcıya hedef ürün hakkındaki öneri değeri olarak sunulur [45].

Yapılan deneylerde *KTKEYK* algoritmasında  $K$  değeri 50 olarak belirlenmiştir. Benzerlik hesabında ve tahmin hesabında  $z$ -puanı kullanıldığından ortaya çıkan tahmin değerinin denormalize edilerek sunulması gerekmektedir. Denormalize işleminde ortaya çıkan tahmin değeri kullanıcının standart sapması ile çarpılıp

ortalaması ile toplanır. Denormalize işlemi Denklem 9.2'de gösterildiği şekilde yapılmıştır.

$$\hat{O}'_{a,\ddot{u}} = \bar{v}_a + \sigma_a \times \hat{O}_{a,\ddot{u}} \quad (9.2)$$

$\bar{v}_a$  aktif kullanıcının oylarının ortalamasını,  $\sigma_a$  aktif kullanıcının standart sapmasını,  $\hat{O}_{a,\ddot{u}}$  aktif kullanıcıya Denklem 4.3'te gösterilen şekilde normalize edilmemiş öneri değerini,  $\hat{O}'_{a,\ddot{u}}$  aktif kullanıcı için üretilen normalize edilmiş öneri değerini gösterir.

#### 9.4.1.2. *Kullanıcı tabanlı eşik değer*

*KTKEYK* algoritmasında Aktif Kullanıcıya en çok benzeyen  $K$  tane kullanıcı seçilmiş ve seçilen  $K$  tane kullanıcının hedef ürüne verdikleri oylar tahmin üretme hesabında kullanılmıştır.  $K$  değeri 50 olduğu durumda sadece 50 tane kullanıcı aktif kullanıcı ile tahmin hesabına dâhil olur. *KTKEYK* algoritmasında aktif kullanıcının komşuluğunda olan kullanıcılarla olan benzerlik değerinin ne olduğuna bakılmaz, benzerlik değeri ne olursa olsun  $K$  tane kullanıcı hesaba dâhil edilir. Bazı durumlarda aktif kullanıcı ile eğitim kümesindeki kullanıcılar arasındaki benzerlik değeri yüksek olan kullanıcı sayısı  $K$  dan fazla olur.  $K$  değeri 50 olduğunda, Pearson korelasyon katsayısına göre hesaplanan benzerlik değeri '+1' olan 100 tane kullanıcı olsa, 100 kullanıcının tamamı tahmin hesaplama işlemine katılmaz sadece 50 kullanıcı hesaba katılır. Diğer taraftan aktif kullanıcıyla olan benzerlik katsayısı düşük de olsa 50 kullanıcı öneri hesaplama işlemine dâhil edileceğinden bazı kullanıcılar için hesaplanacak öneriler istenmeyen şekilde olabilir.

*KTED* algoritmasında *KTKEYK* algoritmasından farklı olarak tahmin üretme hesabında en yakın  $K$  tane komşu yerine aktif kullanıcı ile benzerlik değeri belirli bir eşik değer üzerinde bulunan bütün kullanıcılar katılır. *KTKEYK* algoritmasında aktif kullanıcının diğer kullanıcılarla olan benzerlik değerlerinin önemi yoktur,  $K$  değeri 50 olduğu durumda kullanıcılar arasındaki benzerlik değeri ne olursa olsun 50 kullanıcı tahmin hesabına katılır [45]. *KTED* algoritmasıyla ilgili literatürde  $z$ -puanı tabanlı herhangi bir çalışma bulunamamıştır. Bu tezin kapsamı en uygun  $K$  değerini tespit etmek olmadığından kullanıcı komşuluklarının oluşabileceği en uygun  $K$  değeri sezgisel olarak 0.5 seçilmiştir. Yapılan deneylerde  $K$  değerinin 0.5

seçilmesinin uygun olduğu üretilen önerilerin doğruluğunun 5 yıldızlı oylama sistemini kullanan *KTED* algoritmasından nispeten daha iyi olduğu gözlemlenmiştir.

#### 9.4.2. Model tabanlı algoritmalar

##### 9.4.2.1. Ürün tabanlı *K* yakın komşu

*ÜTKEYK* algoritması model tabanlı bir algoritmadır. Veri kümesindeki ürünler arasındaki benzerlik değerleri hesaplanarak model oluşturulur, tahminler oluşturulan bu modele göre hesaplanır. Tablo 9.3'de gösterildiği gibi ürünler arasındaki benzerlik değeri ilgili ürünleri ortak oylayan kullanıcıların verisi üzerinden yapılır.

**Tablo 9.3. Ürünler Arasında Benzerlik Hesabı**

	ü <sub>1</sub>	ü <sub>2</sub>	ü <sub>3</sub>	ü <sub>4</sub>	ü <sub>5</sub>	ü <sub>6</sub>
k <sub>1</sub>			R	R		
k <sub>2</sub>			-	R		
k <sub>3</sub>			R	R		
k <sub>4</sub>						
k <sub>5</sub>			R	R		
k <sub>6</sub>			R	-		
k <sub>7</sub>				R		

Ürünler arasındaki benzerlikler kullanıcı-ürün matrisindeki ortak oylanmış ürünler üzerinden yapılır. ü<sub>3</sub> ve ü<sub>4</sub> arasındaki benzerlik hesabına k<sub>1</sub>, k<sub>3</sub> ve k<sub>5</sub> kullanıcılarının verdikleri oylar katılır.

DeneySEL çalışmalarında ürünler arasındaki benzerlik değeri Denklem 4.1.'deki gibi Pearson korelasyon hesabına göre yapılmıştır. Veri kümesindeki bütün ürünler arasında benzerlik hesabı yapılarak model oluşturulmuştur. Aktif kullanıcının hedef ürününe tahmin üretme aşamasında hedef ürüne en benzeyen *K* tane ürün seçilmiştir. DeneySEL çalışmalarında *K* sayısı 50 olarak belirlenmiştir [17].

##### 9.4.2.2. Ağırlıklandırılmış Bir Eğimli Tahmin Edici

*ABETE* yüksek doğrulukta tahminler üretebilen model tabanlı bir *OF* öneri algoritmasıdır [57]. Algoritma ilk olarak gerçek oy değerleriyle tahminler üretmek için önerilmiştir. *ABETE* algoritmasının *z*-puan değerlerini kullanan formu da *OF*

öneri sistemlerinde kullanılmaktadır [58]. *ABETE* algoritmasında öncelikle ürünler arasındaki sapmalar Denklem 9.3'teki gibi hesaplanarak model oluşturulur.

$$sap_{ij} = \sum_{k \in S_{ij}} \frac{z_{ki} - z_{kj}}{C_{ij}} \quad (9.3)$$

Denklem 9.3'de  $sap_{ij}$   $i$  ve  $j$  ürünleri arasındaki sapma değeridir.  $z_{ki}$  ve  $z_{kj}$  herhangi bir  $k$  kullanıcısının  $i$  ve  $j$  ürününe verdiği oy değerlerinin  $z$ -puanını;  $C_{ij}$   $i$  ve  $j$  ürünlerinin her ikisini de oylayan kullanıcı sayısını göstermektedir.

Herhangi bir  $k$  kullanıcısının herhangi bir  $i$  ürününe üretilen  $\bar{O}$  öneri değeri Denklem 9.4'teki gibi hesaplanır.

$$\bar{O}_{ki} = \bar{v}_k + \sigma_k \times \frac{\sum_{j \in S} (sap_{ij} + z_{kj}) \times C_{ij}}{\sum_{j \in S} C_{ij}} \quad (9.4)$$

$\bar{v}_k$  aktif kullanıcının oylarının ortalamasını,  $\sigma_k$  aktif kullanıcının standart sapmasını gösterir.

### 9.4.3. Hibrit algoritmalar

#### 9.4.3.1. *K-Ortalamalar*

*KO* algoritması *OF* öneri sistemlerinde kullanılan hibrit bir algoritmadır. *KO* algoritması kullanıldığında eğitim kümesindeki kullanıcılar önceden belirlenen sayıda sınıfa ayrılır, bu süreç algoritmanın model tabanlı tarafını oluşturur. Aktif kullanıcıya tahmin üretme sürecinde önce aktif kullanıcının önceden oluşturulmuş sınıflardan hangisine ait olduğu tespit edilir. Aktif kullanıcının ait olduğu sınıf tespit edildikten sonra o sınıftaki kullanıcılardan en benzer  $K$  tane komşu seçilerek tahmin üretme işlemi yapılır, bu ikinci aşama *KO* algoritmasının hafıza tabanlı tarafını oluşturur. Bu sebeple *KO* algoritması hibrit bir algoritmadır. *KO* algoritmasında kullanıcılar arasındaki benzerlik hesabında birçok yöntem kullanılabilir. Bu çalışmada kullanıcılar arasındaki benzerlik hesabında kullanıcıların  $z$ -puan değerleri çarpılarak hesaplanan benzerlik değeri kullanılmıştır. Kullanıcıların sınıflandırılması işlemi eğitim kümesindeki kullanıcılar üç sınıfa ayrılmıştır. [59].

### 9.4.3.2. Bulanık C- ortalamalar

*BCO* algoritması *OF* öneri sistemlerinde kullanılan hibrit bir algoritmadır. *K-Ortalamalar* algoritmasında sınıf sayısı üç olarak seçildiği durumda herhangi bir kullanıcı mutlaka üç sınıftan birine ait olur. *BCO* algoritmasında sınıf sayısı üç olduğunda bir kullanıcı sadece bir sınıfın üyesi olmaz, üç sınıfa da farklı değerlerle üye olur. Kullanıcılar sınıflara dâhil edilirken, sınıflara olan uzaklık değerleri ile orantılı olarak sınıflara dâhil edilirler. Kullanıcı en yakın olduğu sınıfa en yüksek değerle üye olurken en uzak olduğu sınıfa en düşük üyelik değeriyle üye olur [59].

## 9.4. Kullanılan Metodoloji

MovieLens veri kümesindeki 6.040 kullanıcıdan 1.000 kullanıcı, Netflix veri kümesindeki 10.000 kullanıcıdan 1.000 kullanıcı rastgele seçilerek test kümeleri oluşturulmuştur. Seçilen test kümesi için kullanıcı sayısı, ürün sayısı ve oy sayısı bilgileri Tablo 9.4’de verilmiştir. Netflix veri kümesinin doluluk oranı MovieLens veri kümesinden düşük olduğu ve Netflix test kümesindeki oy sayısının MovieLens test kümesindeki oy sayısından az olduğu görülmektedir.

**Tablo 9.4.** *Kullanılan TEST Kümesindeki Kullanıcı, Ürün ve Oy Sayıları*

Veri kümesi	Kullanıcı Sayısı	Ürün Sayısı	Oy sayısı
MovieLens	1.000	3.952	167.123
Neflix	1.000	4.000	51.234

MovieLens veri kümesindeki 6.040 kullanıcıdan 1.000 kullanıcı test kümesi olarak ayrıldıktan sonra geri kalan 5.040 kullanıcı arasından rastgele seçilen 500, 1.000, 2.000 ve 4.000 kullanıcıdan oluşan eğitim kümeleri oluşturuldu. Eğitim kümelerindeki hiçbir kullanıcı test kümesinde olmayacak şekilde seçildi. Aynı yöntemle Netflix veri kümesindeki 10.000 kullanıcıdan 1.000 kullanıcı test kümesi, olarak ayrıldıktan sonra geri kalan 9.000 kullanıcı arasından rastgele seçilen 500, 1.000, 2.000 ve 4.000 kullanıcıdan oluşan eğitim kümeleri oluşturuldu. MovieLens veri kümesinde olduğu gibi eğitim kümelerindeki hiçbir kullanıcı test kümesinde olmayacak şekilde seçildi. Her iki veri kümesinde oluşturulan 500, 1.000, 2.000 ve

4.000 kullanıcıdan oluşan eğitim kümelerinin her birinden 10'ar tane oluşturuldu. Deneyler her gruptan 10'ar tane oluşturulmuş eğitim kümesi kullanılarak 10'ar defa yapıldı.

Her iki veri kümesi kullanılarak yapılan deneylerde test kümesi olarak ayrılan kümede bulunan 1.000 kullanıcının tüm oyları tahmin edilecek şekilde tasarlandı. Her kullanıcının verdiği gerçek oylar sırayla birer birer saklandı ve saklanan oy tahmin edildi, bir önceki tahmin işlemine saklanan oy dâhil edilerek bir sonraki oy kapatıldı. Bu yöntemle kullanıcıların sistemde var olan oyları 6 tane algoritma kullanılarak ayrı ayrı tahmin edildi. Bu şekilde MovieLens test kümesindeki 167.123 ve Netflix test kümesindeki 51.234 gerçek oyun tamamı tahmin edildi.

Kullanıcı sayısı 500 olan eğitim kümeleri, egitim500\_1, egitim500\_2, egitim500\_3, ..., egitim500\_10 şeklinde kaydedildi. Kullanıcı sayısı 1.000, 2.000 ve 4.000 olan diğer eğitim kümelerinin her birinden oluşturulan 10'ar tane küme de aynı şekilde oluşturularak kaydedildi. Kullanıcı sayısı 500 olan eğitim kümeleri kullanıldığı durumda *OMH* ve *HKOK* hesaplaması işlemi; her iki veri kümesinde test kümesi olarak ayrılan kümedeki kullanıcı oyları önce deneylerde kullanılan 6 algoritmadan birinci algoritma tarafından egitim500\_1 eğitim kümesi kullanılarak tahmin edildi. Ortaya çıkan tahmin kümesi Tablo 9.5'te gösterildiği gibi birinci tahmin kümesi olarak kaydedildi. Aynı algoritma kullanılarak test kümesindeki kullanıcı oyları egitim500\_2 eğitim kümesi kullanılarak tahmin edildi. Oluşturulan tahmin kümesi ikinci tahmin kümesi olarak kaydedildi. Bu işlem kullanıcı sayısı 500 olan 10 tane eğitim kümesi kullanılarak yapıldı. Ortaya birinci algoritma tarafından tahmin edilen 10 tane tahmin kümesi çıkmıştır.

**Tablo 9.5. Örnek Tahmin Kümeleri  $KS=500$**

Test Kümesi				1'inci tahmin Kümesi				n'inci tahmin Kümesi				10'uncu tahmin Kümesi			
	ü <sub>1</sub>	ü <sub>2</sub>	ü <sub>3</sub>		ü <sub>1</sub>	ü <sub>2</sub>	ü <sub>3</sub>		ü <sub>1</sub>	ü <sub>2</sub>	ü <sub>3</sub>		ü <sub>1</sub>	ü <sub>2</sub>	ü <sub>3</sub>
k <sub>1</sub>	3	4		k <sub>1</sub>	2,7	3,6		k <sub>1</sub>	1,8	2,9		k <sub>1</sub>	2,2	4,9	
k <sub>2</sub>		5	1	k <sub>2</sub>		2,2	1,8	k <sub>2</sub>		4,1	2,3	k <sub>2</sub>		4,9	1,4
k <sub>3</sub>	5		5	k <sub>3</sub>	4,7		4,1	k <sub>3</sub>	4,2		4,8	k <sub>3</sub>	4,6		4,3
k <sub>4</sub>		5	5	k <sub>4</sub>		4,7	4,6	k <sub>4</sub>		1,9	2,8	k <sub>4</sub>		4,8	2,8



$k_1$  kullanıcısına  $\ddot{u}_1$  ürünü için üretilen tahmindeki hata değeri hesaplanırken;  $k_1$  kullanıcısının  $\ddot{u}_1$  ürününe verdiği gerçek oy değerinden, 10 tane tahmin kümesindeki  $k_1$  kullanıcısının  $\ddot{u}_1$  ürünü için üretilen tahmin değerleri çıkarıldı. Ortaya çıkan farklarının mutlak değerlerinin ortalaması hesaplandı. Elde edilen değer  $k_1$  kullanıcısına  $\ddot{u}_1$  ürünü için üretilen öneri değerindeki hata değeridir. Test kümesindeki kullanıcıların tüm oyları için bu hesaplamalar yapıldığında Tablo 9.4 elde edilir. Tablo 9.6, birinci algoritmanın eğitim kümesi kullanıcı sayısı 500 olduğu durumda test kümesindeki kullanıcılar için ürettiği önerilerin hata değerlerini gösteren sonuç kümesidir. Tablo 9.5'te verilen tablodaki  $k_1$ ,  $k_2$ ,  $k_3$  ve  $k_4$  kullanıcılarının gerçek oyları için üretilen tahminlerdeki hata değerinin ortalamaları Tablo 9.6'da verilmiştir.

**Tablo 9.6. Sonuç Kümesi Örneği  $KS=500$**

	$\ddot{u}_1$	$\ddot{u}_2$	$\ddot{u}_3$
$k_1$	0,76	0,80	
$k_2$		1,27	0,80
$k_3$	0,50		0,60
$k_4$		1,20	1,60

Tablo 9.4'da 4 kullanıcı ve 3 ürün için gösterilen kümedeki değerlerin ortalaması bu öneri işleminde kullanılan algoritmanın ürettiği önerilerin *OMH* değerini, karelerinin ortalaması *HKOK* değerini verir.

Deneyler, bu bölümde anlatılan hesaplama yöntemiyle her iki veri kümesinde de ayrı ayrı 6 algoritma kullanılarak yapılmıştır. Kullanılan 6 algoritma tarafından eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumda üretilen tahminlerle gerçek değerler arasındaki farkların ortalamaları hesaplanarak Tablo 9.6'da gösterildiği gibi kaydedilmiştir.

## 10. DENEYSEL ÇALIŞMALAR

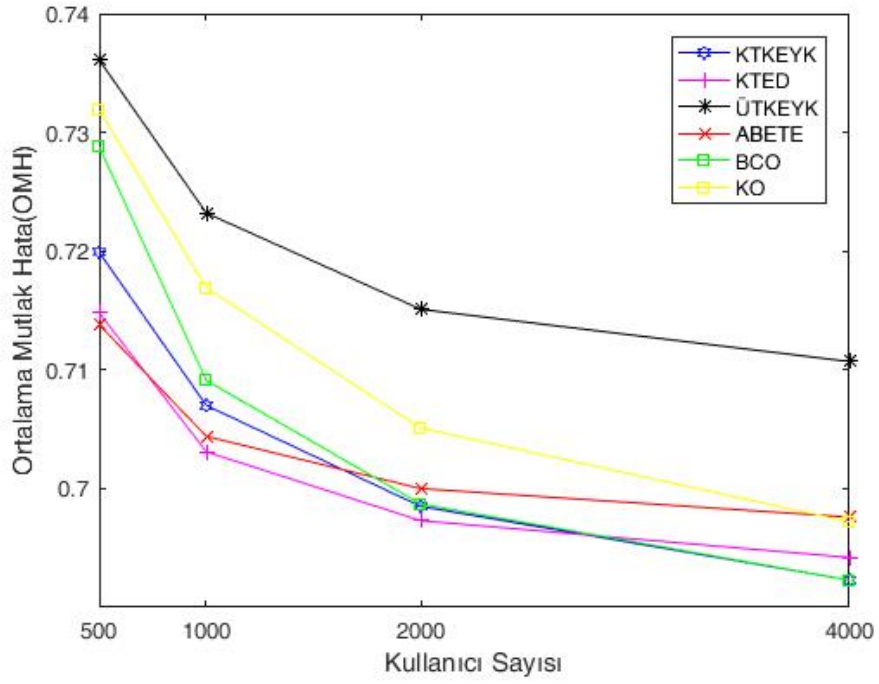
### 10.1. Kullanılan Algoritmaların *OMH* ve *HKOK* Değerleri

MovieLens veri kümesi kullanılarak yapılan deneylerde 6 tane algoritmanın sonuçları Tablo 10.1' de gösterilmiştir. Kullanıcı sayısı 500, 1.000, 2.000, 4.000 olarak değişen eğitim kümeleri kullanılarak yapılan deneylerde kullanılan algoritmaların *OMH* ve *HKOK* doğruluk ölçütlerine göre hata değerlerinin ortalamaları verilmiştir.

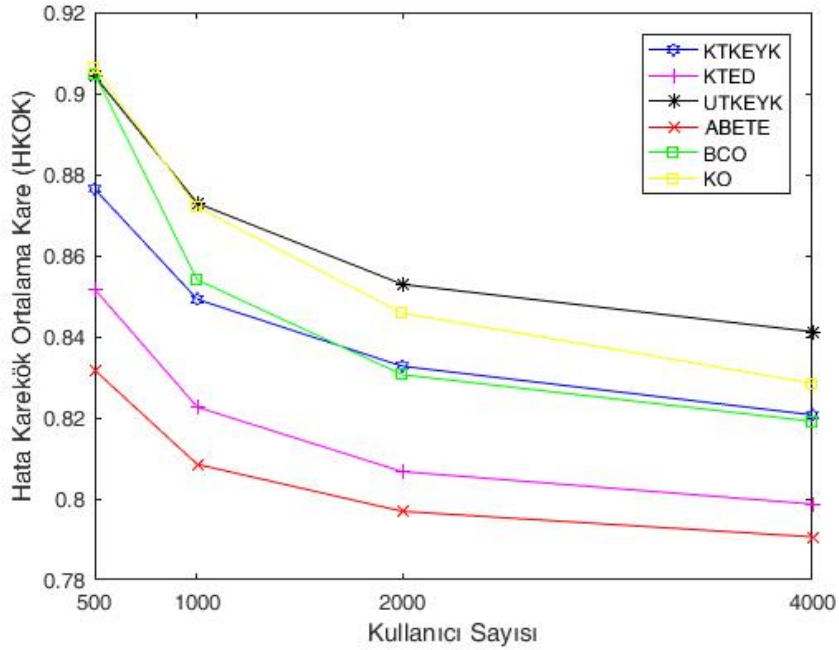
**Tablo 10.1.** *MovieLens* Veri Kümesi 6 Algoritmanın Deney Sonuçları

Algoritma	Doğruluk Ölçütü	Kullanıcı Sayısı			
		500	1.000	2.000	4.000
KTKEYK	OMH	0,7199	0,7070	0,6985	0,6923
	HKOK	0,8764	0,8492	0,8327	0,8207
KTED	OMH	0,7148	0,7031	0,6973	0,6942
	HKOK	0,8517	0,8226	0,8067	0,7987
ÜTKEYK	OMH	0,7361	0,7232	0,7151	0,7107
	HKOK	0,9043	0,8729	0,8529	0,8412
ABETE	OMH	0,7138	0,7044	0,7000	0,6976
	HKOK	0,8317	0,8085	0,7969	0,7906
BCO	OMH	0,7288	0,7091	0,6987	0,6923
	HKOK	0,9048	0,8540	0,8306	0,8191
KO	OMH	0,7319	0,7169	0,7051	0,6972
	HKOK	0,9064	0,8719	0,8458	0,8285

Her bir eğitim kümesinden 10 farklı eğitim kümesi olduğundan her algoritma için 10 farklı deney yapılmış ve bu 10 farklı sonucun ortalaması kullanılan metodoloji bölümünde anlatıldığı gibi hesaplanarak Tablo 10.1' de gösterilmiştir. Sonuçlara bakıldığında eğitim setindeki kullanıcı sayısı arttığında doğruluğun arttığı gözlemlenmektedir. Şekil 10.1 ve Şekil 10.2'de *OMH* ve *HKOK* değerleri grafikte sunulmuştur.



**Şekil 10.1.** *MovieLens, Kullanılan Algoritmaların OMH Değerleri*



**Şekil 10.2.** *MovieLens, Kullanılan Algoritmaların HKOK Değerleri*

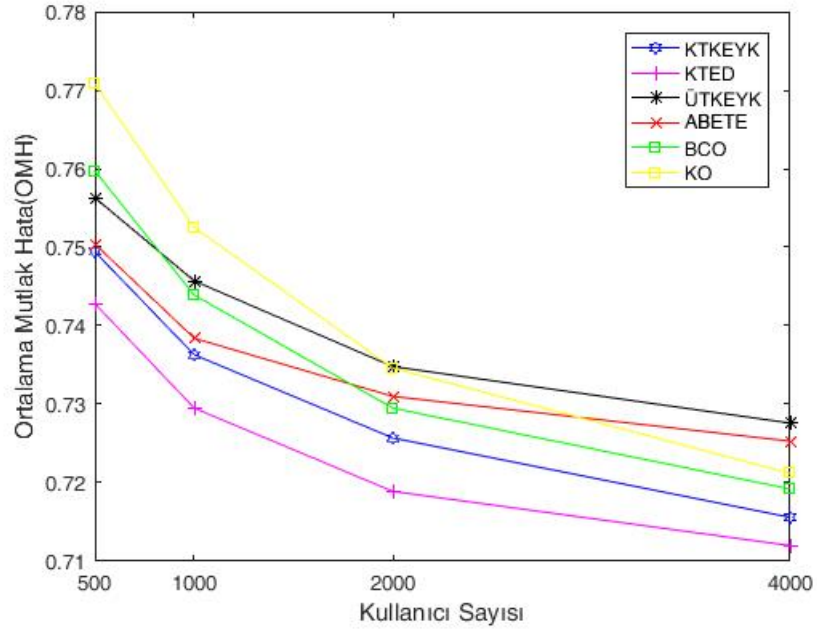
Netflix veri kümesinde yapılan deney sonuçları MovieLens veri kümesindeki deney sonuçlarıyla paralellik göstermektedir. Tablo 10.12’de Netflix veri kümesi kullanılarak yapılan deneylerde MovieLens veri kümesinde olduğu gibi 6 tane

algoritmanın sonuçları gösterilmiştir. Kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olarak değişen eğitim kümeleri kullanılarak üretilen tahminlerin *OMH* ve *HKOK* doğruluk ölçütlerine göre hata değerlerinin ortalamaları verilmiştir. Sonuçlara bakıldığında eğitim setindeki kullanıcı sayısı arttığında doğruluğun arttığı gözlemlenmektedir. Netflix veri kümesindeki deney sonuçlarına bakıldığında doğruluk değerleri MovieLens deney setindeki doğruluk değerlerinden küçüktür. Bölüm 4.3'te *OF* öneri sistemlerinin problemlerinde anlatıldığı gibi veri kümelerinin doluluk oranının doğruluk üzerindeki etkisi Tablo 10.2'de görülmektedir.

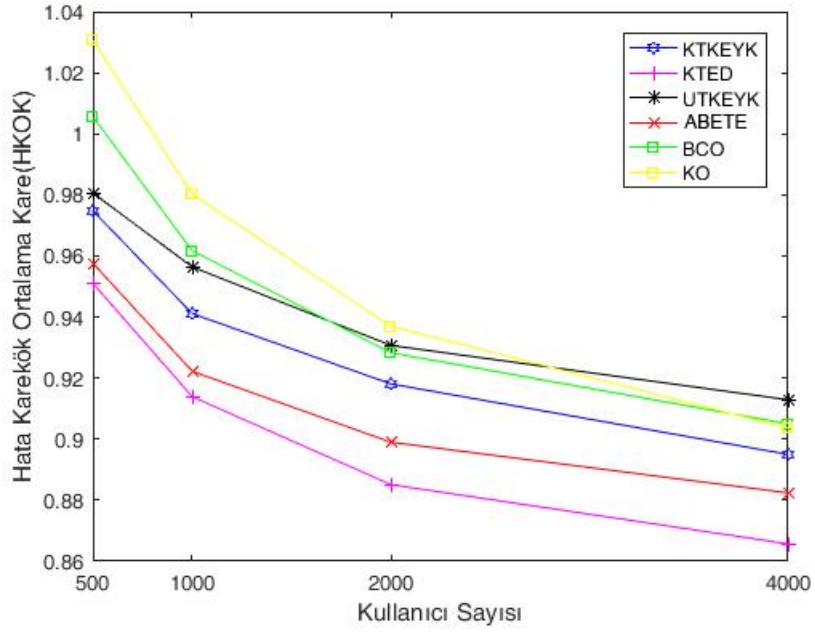
**Tablo 10.2.** *Netflix Veri Kümesi 6 Algoritmanın Deney Sonuçları*

Algoritma	Doğruluk Ölçütü	Kullanıcı Sayısı			
		500	1.000	2.000	4.000
KTKEYK	OMH	0,7494	0,7363	0,7257	0,7156
	HKOK	0,9748	0,9411	0,9181	0,8950
KTED	OMH	0,7428	0,7295	0,7189	0,7120
	HKOK	0,9510	0,9139	0,8851	0,8657
ÜTKEYK	OMH	0,7597	0,7439	0,7295	0,7192
	HKOK	1,0056	0,9617	0,9283	0,9050
ABETE	OMH	0,7708	0,7525	0,7346	0,7212
	HKOK	1,0306	0,9802	0,9370	0,9036
BC O	OMH	0,7563	0,7457	0,7348	0,7276
	HKOK	0,9806	0,9564	0,9307	0,9129
KO	OMH	0,7504	0,7384	0,7310	0,7253
	HKOK	0,9576	0,9222	0,8990	0,8824

Tablo 10.2'de Netflix veri kümesinde yapılan deneylerde kullanılan 6 farklı algoritmanın ürettiği değerlerin *OMH* ve *HKOK* değerleri verilmiştir. Şekil 10.3 ve Şekil 10.4'de *OMH* ve *HKOK* değerlerinin grafikleri gösterilmiştir.



Şekil 10.3. Netflix, Kullanılan Algoritmaların OMH Değerleri



Şekil 10.4. Netflix, Kullanılan Algoritmaların HKOK Değerleri

## 10.2. Kullanıcı İçin En İyi Algoritmanın Tespit Edilmesi

### 10.2.1. MovieLens veri kümesi deney sonuçları

MovieLens veri kümesinde yapılan deneylerde test kümesindeki 5 farklı kullanıcıya ait önerilerin *OMH* değerleri aşağıdaki tablolarda sunulmuştur.  $(k_1, k_2, k_3, k_4, k_5)$  kullanıcılarının sistemdeki gerçek oyları 6 farklı algoritma kullanılarak tahmin edilmiştir. Her bir algoritmanın, farklı sayıda kullanıcı bulunan eğitim kümeleri kullanarak ürettiği tahmin değerlerinin *OMH*'sı hesaplanarak Tablo 10.3, Tablo 10.4, Tablo 10.5 ve Tablo 10.6'da sunulmuştur. Bu kısımda sunulan tablolarda *OMH* doğruluk değerleri verilmiştir. *HKOK* doğruluk değerleri de hesaplanmış ve sonuçlar *OMH* ile benzer olduğu gözlemlenmiştir.

**Tablo 10.3.** *MovieLens, Kullanıcı İçin En İyi Algoritma KS=500*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k_1$	0,5928	0,5569	0,6688	0,5464	0,5856	0,5908
$k_2$	0,6928	0,6915	0,7243	0,6622	0,7032	0,7012
$k_3$	0,4608	0,4665	0,5268	0,5278	0,4578	0,4723
$k_4$	0,9067	0,9009	0,9502	0,9377	0,9559	1,0092
$k_5$	0,5766	0,5675	0,6495	0,5680	0,5773	0,5804

Tablo 10.3'te 5 farklı kullanıcıya ait 6 farklı algoritma tarafından üretilmiş tahminlerin *OMH* değerleri görülmektedir. Eğitim kümesinde kullanıcı sayısı 500 olduğunda  $k_1$  kullanıcısı için *ABETE* algoritmasının ürettiği tahminlerin *OMH*'sı 0,5464 ile en iyi değerdedir. Diğer bir yorumla  $k_1$  kullanıcısı için eğitim setindeki kullanıcı sayısı 500 olduğunda en iyi sonuç veren algoritma *ABETE* algoritmasıdır. Aynı şekilde  $k_3$  kullanıcısına bakacak olursak en iyi *OMH* değeri 0,4578 olarak karşımıza çıkmaktadır.  $k_3$  kullanıcısı için en iyi sonucu *BCO* algoritması vermiştir. Bu şekilde bütün kullanıcılar için en iyi sonuç veren algoritmalar seçildiğinde doğruluk değerleri ortalama olarak 0,7242 iken 0,6996 olmuştur. Bu sonuçlar Şekil 10.5'te gösterilmiştir. Tablo 10.4, Tablo 10.5, Tablo 10.6'da eğitim kümesindeki kullanıcı sayısı değiştiğinde 5 farklı kullanıcı için en iyi sonucu veren algoritmaların

nasıl deęiřtięi gsterilmiřtir. Bu tablolardaki sonulara bakıldıęında *OF* neri sistemlerinde tek bir algoritmanın btn kullanıcılar iin en iyi sonu vermedięi grlmektedir. En iyi sonucu veren algoritma, hem kullanıcıya hem de eęitim setindeki kullanıcı sayısına gre deęiřmektedir. Tablo 10.4'te eęitim kmesindeki kullanıcı sayısı 1.000 dir,  $k_1$  kullanıcısı iin en iyi sonu veren algoritma *ABETE* olmuřtur. Eęitim kmesindeki kullanıcı sayısı aynı olmasına raęmen farklı kullanıcılar geldike her kullanıcı iin en iyi sonucu farklı farklı algoritmaların verdięi grlmektedir. Bazı durumlarda aynı algoritmanın birden fazla kullanıcı iin en iyi olduęu grlmektedir.  $k_4$  ve  $k_5$  kullanıcıları iin *KTED* algoritması en iyi doęrulukta neriler retmektedir.

**Tablo 10.4.** *MovieLens, Kullanıcı İin En İyi Algoritma KS=1.000*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	TKEYK	ABETE	BCO	KO
$k_1$	0,5865	0,5519	0,6628	0,5400	0,5867	0,5992
$k_2$	0,6686	0,6767	0,7312	0,6488	0,6645	0,6846
$k_3$	0,4536	0,4565	0,5008	0,5163	0,4467	0,4525
$k_4$	0,8868	0,8855	0,9288	0,9380	0,9152	0,9773
$k_5$	0,5780	0,5608	0,6505	0,5647	0,5731	0,5832

**Tablo 10.5.** *MovieLens, Kullanıcı İin En İyi Algoritma KS=2.000*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	TKEYK	ABETE	BCO	KO
$k_1$	0,6042	0,5485	0,6664	0,5420	0,5906	0,6063
$k_2$	0,6431	0,6640	0,7060	0,6369	0,6467	0,6431
$k_3$	0,4506	0,4536	0,4842	0,5157	0,4525	0,4575
$k_4$	0,8803	0,8838	0,9274	0,9315	0,9083	0,9253
$k_5$	0,5856	0,5578	0,6401	0,5614	0,5856	0,5860

**Tablo 10.6.** *MovieLens, Kullanıcı İçin En İyi Algoritma KS=4.000*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k <sub>1</sub>	0,5805	0,5446	0,6492	0,5399	0,5860	0,5895
k <sub>2</sub>	0,6177	0,6618	0,6934	0,6386	0,6211	0,6175
k <sub>3</sub>	0,4491	0,4549	0,4678	0,5184	0,4523	0,4593
k <sub>4</sub>	0,8880	0,8829	0,9127	0,9330	0,9149	0,8899
k <sub>5</sub>	0,5915	0,5564	0,6519	0,5611	0,5883	0,5928

Tablo 10.3, Tablo 10.4, Tablo 10.5 ve Tablo 10.6'da gösterilen kullanıcılar ( $k_1, k_2, k_3, k_4, k_5$ ) her tabloda aynı kullanıcılarıdır. Kullanıcıların farklı oldukları durumlar da incelenmiştir. Test kümesinden 5 farklı kullanıcı ( $k_1, k_2, k_3, k_4, k_5$ ) alınmış ve bu kullanıcılara ait 6 algoritma tarafından üretilen tahminlere bakılmıştır. Eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000, 4.000 olduğu her durumda en iyi sonucu veren algoritmanın farklı bir algoritma olduğu görülmüştür.

Eğitim kümesi kullanıcı sayısı 500 olduğu durumda  $k_3$  kullanıcılarına en iyi doğrulukta öneri üreten algoritma *BCO* algoritmasıdır. Aynı  $k_3$  kullanıcılarına eğitim kümesi kullanıcı sayısı 4.000 olduğu durumda en iyi sonucu veren algoritma *KTEYK* olduğu görülmektedir. Eğitim kümesi kullanıcı sayısı değiştikçe bazı kullanıcılar için en iyi sonucu veren algoritma aynı kalmaktadır.  $k_1$  kullanıcılarına en iyi sonucu veren algoritma her zaman *ABETE* algoritması olmuştur.

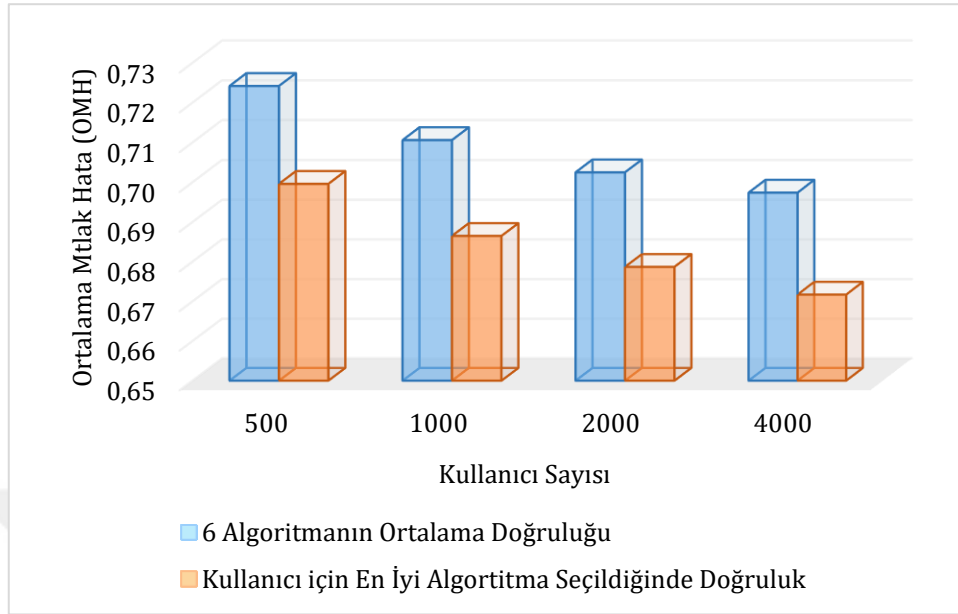
Tablo 10.7'de kullanıcı için en iyi algoritma seçildiğinde doğruluk değerleri verilmiştir. Eğitim kümesi kullanıcı sayısı 2.000 olduğu durumda yapılan deneylerde 6 algoritmanın ortalama doğruluğu 0,7025 iken kullanıcı için en iyi sonuç veren algoritma seçildiğinde doğruluğun 0,6787 olduğu görülmektedir.

**Tablo 10.7.** *Kullanıcı İçin En İyi Algoritma Seçildiğinde OMH*

	Kullanıcı Sayısı			
	500	1.000	2.000	4.000
6 Algoritmanın Ortalama Doğruluğu	0,7242	0,7106	0,7025	0,6974
Kullanıcı İçin En İyi Algoritma Seçildiğinde Doğruluk	0,6996	0,6865	0,6787	0,6717



Şekil 10.5'te Tablo 10.7'de verilen sayısal değerlerin grafiği verilmiştir.



**Şekil 10.5.** *MovieLens, Kullanıcı için En iyi Algoritma*

Şekil 10.5'te MovieLens veri kümesinde yapılan deneylerde kullanıcı için en iyi algoritma seçildiğinde doğruluğun nasıl değiştiği gösterilmiştir. Sütun grafiğindeki algoritmaların ortalama doğrulukları her bir gruptaki 6 tane algoritma kullanılarak yapılan deneylerde ortaya çıkan sonuçların ortalama değerleridir. En iyi sonuç veren algoritma seçildiğinde ortaya çıkan doğruluk değeri ise kullanıcının istediği tüm önerileri üretmek için 6 algoritma arasından en iyi sonuç veren seçildiğinde ortaya çıkan sonuçtur. Şekil 10.5'te de görüldüğü gibi kullanıcıya göre en iyi sonucu veren algoritma seçilirse doğruluk çok daha iyileşecektir.

## 10.2.2. Netflix Deney Sonuçları

Netflix veri kümesinde yapılan deneylerde test kümesindeki 5 farklı kullanıcıya ait önerilerin *OMH* değerleri Tablo 10.8, Tablo 10.9, Tablo 10.10 ve Tablo 10.11'de sunulmuştur. Bu kısımda sunulan tablolardaki hesaplamalar bölüm 9.5'te anlatıldığı gibi yapılmıştır. Kullanıcıların sistemdeki oyları 6 farklı algoritma kullanılarak tahmin edilmiş ve her algoritmanın ürettiği tahmin değerlerinin *OMH*'sı hesaplanarak Tablo 10.8, Tablo 10.9, Tablo 10.10 ve Tablo 10.11'da gösterilmiştir. MovieLens veri kümesinde de her kullanıcı için en iyi sonuç veren tek bir

algoritmanın olmadığı görülmektedir. Kullanıcıya göre en iyi sonucu veren algoritma, eğitim setindeki kullanıcı sayısına ve kullanıcıya göre değişmektedir. Her kullanıcı için en iyi sonucu veren algoritma seçilmesi durumunda doğruluk iyileşmektedir. Bununla ilgili sonuçlar Şekil 10.7’de gösterilmiştir. Bu kısımda sunulan tablolarda *OMH* doğruluk değerleri verilmiştir. *HKOK* doğruluk değerleri hesaplanmış ve sonuçların *OMH* ile aynı doğrultuda olduğu gözlemlenmiştir.

**Tablo 10.8.** *Netflix, Kullanıcı İçin En İyi Algoritma KS=500*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k <sub>1</sub>	0,5744	0,6069	0,6716	0,5808	0,5775	0,6189
k <sub>2</sub>	0,8440	0,8742	0,9462	0,9804	0,8123	0,9120
k <sub>3</sub>	0,5914	0,5793	0,6238	0,5898	0,6029	0,6173
k <sub>4</sub>	1,1123	1,1244	1,1042	1,1145	1,1319	1,1324
k <sub>5</sub>	0,9061	0,8736	0,9059	0,8767	0,8988	0,9195

Tablo 10.8, Tablo 10.9, Tablo 10.10 ve Tablo 10.11’de gösterilen kullanıcılar ( $k_1, k_2, k_3, k_4, k_5$ ) her tabloda aynı kullanıcılarıdır. Kullanıcıların farklı oldukları durumlar da incelenmiştir. Eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000, 4.000 olduğu her durumda en iyi sonuç veren algoritma farklı bir algoritma olduğu görülmektedir.

**Tablo 10.9.** *Netflix, Kullanıcı İçin En İyi Algoritma KS=1.000*

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k <sub>1</sub>	0,5287	0,5075	0,6164	0,4673	0,5060	0,5088
k <sub>2</sub>	0,8552	0,8681	0,9885	0,9540	0,8044	0,8868
k <sub>3</sub>	0,5733	0,5587	0,5963	0,5713	0,5831	0,5907
k <sub>4</sub>	1,1016	1,0969	1,0837	1,0977	1,1269	1,1094
k <sub>5</sub>	0,8814	0,8534	0,8906	0,8576	0,8516	0,8879

**Tablo 10.10.** Netflix, Kullanıcı İçin En İyi Algoritma  $KS=2.000$

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k <sub>1</sub>	0,5363	0,5034	0,5994	0,4827	0,5252	0,5545
k <sub>2</sub>	0,8347	0,8552	0,8583	0,9009	0,8211	0,8065
k <sub>3</sub>	0,5665	0,5553	0,5843	0,5735	0,5804	0,5682
k <sub>4</sub>	1,0883	1,0951	1,0891	1,0916	1,0845	1,0731
k <sub>5</sub>	0,9120	0,8608	0,9025	0,8564	0,8854	0,8732

**Tablo 10.11.** Netflix, Kullanıcı İçin En İyi Algoritma  $KS=4.000$

Kullanıcı	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
	KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k <sub>1</sub>	0,5234	0,4873	0,5947	0,4755	0,5322	0,5095
k <sub>2</sub>	0,7480	0,8374	0,9973	0,9188	0,7554	0,7605
k <sub>3</sub>	0,5835	0,5513	0,5891	0,5689	0,5817	0,5596
k <sub>4</sub>	1,0684	1,0854	1,0789	1,0915	1,0630	1,0563
k <sub>5</sub>	0,8829	0,8451	0,8594	0,8412	0,8704	0,8433

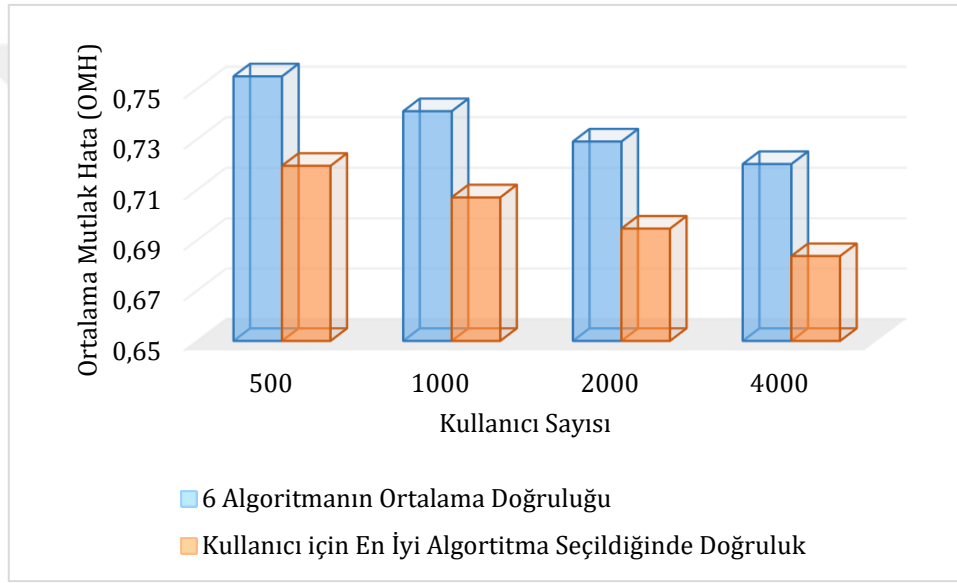
Eğitim kümesi kullanıcı sayısı sayısı 500 olduğu durumda  $k_1$  kullanıcılarına en iyi doğrulukta öneri üreten algoritma *KTEYK* algoritmasıdır. Aynı  $k_1$  kullanıcılarına eğitim kümesi kullanıcı sayısı 2.000 olduğu durumda en iyi sonucu veren algoritmanın *ABETE* olduğu görülmektedir. Eğitim kümesi kullanıcı sayısı 4.000 olduğunda da  $k_1$  kullanıcılarına en iyi doğrulukta öneri üreten algoritma *ABETE* olmuştur.

Tablo 10.12’de kullanıcı için en iyi sonuç veren algoritma seçildiğinde doğruluğun nasıl iyileştiği gösterilmiştir. Eğitim kümesi kullanıcı sayısı 4.000 olduğu durumda yapılan deneylerde 6 algoritmanın ortalama doğruluğu 0,7202 iken kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde doğruluğun 0,6837 olduğu görülmektedir.

**Tablo 10.12.** *Kullanıcı İçin En İyi Algoritma Seçildiğinde OMH*

	Kullanıcı Sayısı			
	500	1000	2000	4000
6 Algoritmanın Ortalama Doğruluğu	0,7549	0,7411	0,7291	0,7202
Kullanıcı İçin En İyi Algoritma Seçildiğinde Doğruluk	0,7195	0,7070	0,6946	0,6837

Şekil 10.6'da Tablo 10.12'de verilen sayısal değerlerin grafiği verilmiştir.



**Şekil 10.6.** *Netflix, Kullanıcı için En iyi Algoritma*

Şekil 10.6'da Netflix veri kümesinde yapılan deneylerde kullanıcı için en iyi algoritma seçildiğinde doğruluğun nasıl değiştiği gösterilmiştir. MovieLens veri kümesindeki deneylerde de olduğu gibi kullanıcıya göre en iyi sonucu veren algoritma seçilirse doğruluk çok daha iyileşecektir. Bu şekilden çıkarılabilecek diğer bir sonuç ise MovieLens deneylerinde de olduğu gibi eğitim setindeki kullanıcı sayısı arttıkça doğruluk iyileşmektedir.

Kullanıcı için en iyi sonuç veren algoritma seçilmesi durumundaki doğruluğun iyileşmesinin rastlantısal bir durum olmadığını göstermek için iyileşmenin istatistiksel açıdan anlamlı olduğunu göstermek gerekir. Bu anlamlılığı ortaya

koymak için bağımlı örneklem *t*-testi uygulanmıştır. Bağımlı örneklem *t*-testinde önerilen yöntem uygulanmadan önceki önerilerin doğruluğu ile önerilen yöntemle elde edilen önerilerin doğruluğu karşılaştırılmıştır. Eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumlar ayrı ayrı incelenmiş ve sonuçlar Tablo 10.13'te sunulmuştur.

**Tablo 10.13.** *Kullanıcı için En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanılan 6 farklı Algoritmanın Ortalama Doğruluğu için t-test Sonuçları*

Veri Kümesi	Eğitim Kümesi Kullanıcı Sayısı							
	500		1.000		2.000		4.000	
	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri
MovieLens	40,384	0,000	42,921	0,000	37,554	0,000	40,235	0,000
Netflix	36,721	0,000	67,482	0,000	48,084	0,000	38,367	0,000

MovieLens veri kümesi için üretilen 167.123 öneri ve Netflix veri kümesi için üretilen 51.234 öneri 10 farklı örnekleme bölünerek *t*-testi uygulanmıştır. Bu durumda serbestlik derecesi dokuzdur. Sonuç olarak *t*-testinde her iki veri kümesinde, eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumlarda  $p < 0,001$  olduğu görülmektedir. Bu sonuç %99,9 güvenle, kullanıcı için en iyi sonuç veren algoritma seçildiği durumdaki doğruluk ile kullanılan 6 farklı algoritmanın ortalama doğruluğu arasında çok yüksek düzeyde istatistiksel olarak anlamlı bir farklılık vardır şeklinde yorumlanabilir.

### 10.3. Kullanıcı/Ürün Çifti İçin En İyi Algoritmanın Tespit Edilmesi

#### 10.3.1. MovieLens deney sonuçları

Kullanıcı için en iyi algoritmanın tespit edilmesi işleminde, bir kullanıcının bütün gerçek oylarını tahmin eden en iyi algoritma tespit edilmiştir. Diğer bir ifadeyle öneri sistemindeki bir kullanıcının bütün oylarını tek başına en iyi doğrulukta tahmin eden bir tane algoritma seçilerek doğruluk değeri hesaplanmıştır. Kullanıcı/Ürün çifti için en iyi algoritmanın tespit edilmesi işleminde ise bir kullanıcı için sadece bir algoritma kullanmak yerine kullanıcının sistemdeki her bir oyunu en iyi doğrulukta tahmin eden algoritma seçilmiştir.

Tablo 10.14'te test kümesinden rastgele seçilen bir  $k$  kullanıcısına ait rastgele seçilen 5 tane ürünü ( $ü_1, ü_2, ü_3, ü_4, ü_5$ ) için 6 farklı algoritma tarafından üretilen tahmin değerleri ile  $k$  kullanıcısının ürünlere verdiği gerçek oy değerleri arasındaki farklar gösterilmiştir. Sonuçlara bakıldığında  $k$  kullanıcısının  $ü_1$  ürünü için gerçek oy değerine en yakın tahmini üreten algoritma *KTEYK* olmuştur,  $ü_3$  ürünü hakkındaki gerçek oy değerini 0,2028 yaklaşık değer ile *ABETE* algoritması tahmin etmiştir. Bazı ürünler için aynı algoritma en iyi sonucu veren algoritma olmuştur,  $ü_3$  ve  $ü_5$  ürünleri için *ABETE* algoritmasının en iyi sonucu verdiği görülmektedir. Kullanıcı için en iyi algoritmanın tespit edilmesi bölümünde de anlatıldığı gibi Kullanıcı/ürün çifti için de en iyi algoritma eğitim kümesindeki kullanıcı sayısı değiştiğinde değişmektedir. Tablo 10.15, Tablo 10.16 ve Tablo 10.17' de eğitim kümesindeki kullanıcı sayısının değişmesiyle farklı kullanıcılara üretilen tahminlerin nasıl değiştiği gösterilmiştir.

**Tablo 10.14.** *MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=500*

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k$	$ü_1$	0,3146	0,3183	1,9762	0,5938	1,3220	0,4913
	$ü_2$	0,2546	0,2651	0,5779	0,6576	0,6258	0,4419
	$ü_3$	0,4410	0,3682	0,9198	0,2028	0,7862	1,1608
	$ü_4$	0,8904	0,8592	0,7658	1,1929	0,7821	0,5960
	$ü_5$	0,4730	0,3624	1,1233	0,2610	1,1079	0,7480

**Tablo 10.15.** *MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=1.000*

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k$	$ü_1$	0,4116	0,3905	1,6671	0,6090	0,5842	0,5667
	$ü_2$	0,1657	0,2134	0,2803	0,6730	0,3640	0,4681
	$ü_3$	0,3758	0,3924	1,0946	0,1540	0,6043	0,5792
	$ü_4$	0,5815	0,6882	0,3686	1,1383	0,9281	0,4389
	$ü_5$	0,6007	0,3491	0,4667	0,3468	0,6854	0,7776

Tablo 10.14, Tablo 10.15, Tablo 10.16 ve Tablo 10.17’ de gösterilen  $k$  kullanıcısı her tabloda aynı kullanıcı, ürünler( $\ddot{u}_1, \ddot{u}_2, \ddot{u}_3, \ddot{u}_4, \ddot{u}_5$ ) her tabloda aynı  $k$  kullanıcısına ait ürünlerdir. Aynı  $k$  kullanıcısının farklı ürünleri de incelenmiştir. Aynı  $k$  kullanıcısına ait farklı ürünler için de 6 algoritma tarafından üretilen tahminlere bakılmıştır. Eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000, 4.000 olduğu her durumda en iyi sonuç veren algoritma her defasında farklı bir algoritma olarak karşımıza çıkmaktadır.

**Tablo 10.16.** *MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=2.000*

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k$	$\ddot{u}_1$	0,3516	0,3733	2,8557	0,5967	0,7041	0,6712
	$\ddot{u}_2$	0,1500	0,1914	0,4101	0,6642	0,3504	0,4349
	$\ddot{u}_3$	0,6459	0,3644	1,0832	0,1598	0,6142	0,6941
	$\ddot{u}_4$	0,6009	0,8220	0,2087	1,1322	1,4839	0,8633
	$\ddot{u}_5$	0,2168	0,1740	0,7088	0,2961	0,6151	0,6906

Tablo 10.17’da eğitim kümesindeki kullanıcı sayısı 4.000 dir. Bu tabloda da görüldüğü gibi,  $k$  kullanıcısının her bir ürünü için en iyi sonuç veren algoritma değişmektedir. Kullanıcı/ürün çifti için en iyi sonuç veren algoritma her ürün için değişmektedir ve eğer her üründe en iyi sonucu veren algoritma tespit edilirse doğruluk çok daha iyileşecektir.

**Tablo 10.17.** *MovieLens, Kullanıcı/Ürün Çifti İçin En İyi Algoritma KS=4.000*

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k$	$\ddot{u}_1$	0,6360	0,4021	2,5923	0,6228	0,5948	0,4613
	$\ddot{u}_2$	0,2300	0,2069	0,1912	0,6756	0,2432	0,2601
	$\ddot{u}_3$	0,5798	0,3519	1,0894	0,1696	0,7291	0,7723
	$\ddot{u}_4$	0,2215	0,8379	0,2267	1,1190	1,0736	0,7072
	$\ddot{u}_5$	0,3234	0,0966	0,7576	0,3123	0,5246	0,5896

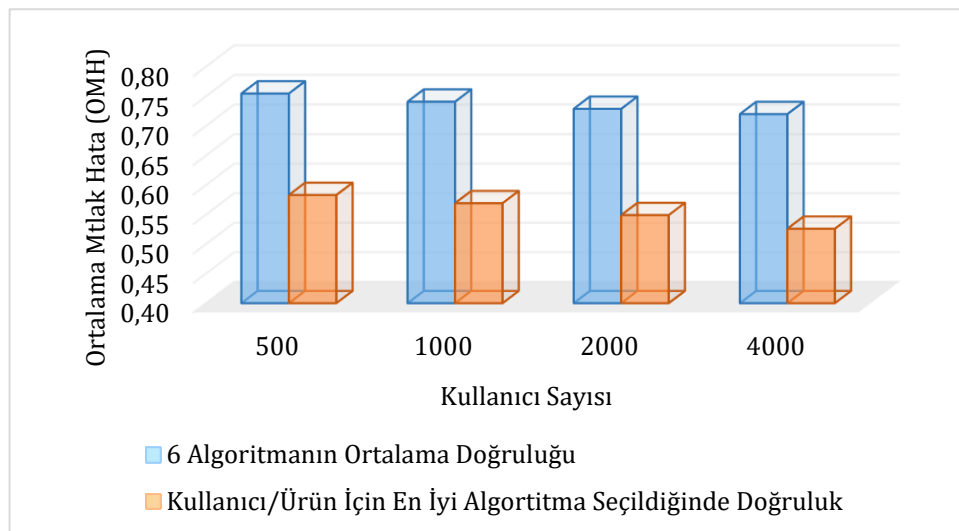
Eđitim kümesi kullanıcı sayısı 500 olduđu durumda  $k$  kullanıcısının  $ü_4$  ürününü ele aldığımızda, en yakın doğrulukta  $KO$  tahmin etmiştir. Aynı  $k$  kullanıcısının aynı ürününü eğitim kümesi kullanıcı sayısı 1.000 olduđu durumda  $ÜTKEYK$  algoritması tahmin etmiştir. Eğitim kümesi kullanıcı sayısı 4.000 olduđu durumda  $k$  kullanıcısının  $ü_4$  ürününe baktığımızda en iyi doğrulukta öneri üreten algoritmanın  $KTKEYK$  olduđu görölmektedir. Bazı ürünleri eğitim kümesi kullanıcı sayısı deđişmesine rağmen aynı algoritma tahmin etmiştir.  $ü_3$  ürününe bakıldığında her zaman  $ABETE$  algoritmasının en iyi tahmin ettiđi görülebilir.

Tablo 10.18’de Kullanıcı/Ürün çifti için en iyi sonuç veren algoritma seçildiğinde doğruluğun nasıl iyileştiđi gösterilmiştir. Eğitim kümesi kullanıcı sayısı 2.000 olduđu durumda yapılan deneylerde 6 algoritmanın ortalama doğruluđu 0,7025 iken kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde doğruluğun 0,5133 olduđu görölmektedir.

**Tablo 10.18.** *Kullanıcı/Ürün Çifti İçin En İyi Algoritma Seçildiğinde OMH*

	Kullanıcı Sayısı			
	500	1000	2000	4000
6 Algoritmanın Ortalama Doğruluđu	0,7242	0,7106	0,7025	0,6974
Kullanıcı/Ürün Çifti İçin En İyi Sonuç Veren Algoritma Seçildiğinde	0,5596	0,5352	0,5133	0,4875

Şekil 10.7’te Tablo 10.18’de verilen sayısal deđerlerin grafiđi verilmiştir.



**Şekil 10.7.** *MovieLens, Kullanıcı/Ürün Çifti için En İyi Algoritma*



Şekil 10.7’de MovieLens veri kümesi için 6 algoritmanın ortalama doğruluğu ve kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde ortaya çıkan doğruluk grafiği gösterilmiştir. Bu grafikten, *OF* öneri sisteminde kullanıcı/ürün çifti için en iyi sonuç veren algoritmanın kullanılması durumunda doğruluğun iyileşeceği çıkarılabilir.

### 10.3.2. Netflix Deney Sonuçları

MovieLens veri kümesinden yapılan deneylerde izlenen yöntem gibi Kullanıcı/Ürün çifti için en iyi algoritmanın tespit edilmesi işleminde bir kullanıcı için sadece bir algoritma kullanmak yerine kullanıcının sistemdeki her bir oyunu en yakın değerde tahmin eden en iyi algoritma araştırılmıştır.

Tablo 10.19’da bir  $k$  kullanıcısına ait rastgele seçilen 5 tane ürünü ( $\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4, \bar{u}_5$ ) için 6 farklı algoritma tarafından üretilen tahmin değerleri ile  $k$  kullanıcısının ürünlere verdiği gerçek oy değerleri arasındaki farklar gösterilmiştir. Sonuçlara bakıldığında  $k$  kullanıcısının  $\bar{u}_1$  ürünü için gerçek oy değerine en yakın tahmini üreten algoritma *BCO* olmuştur,  $\bar{u}_2$  ürünü hakkındaki gerçek oy değerini 0,9584 yaklaşık değer ile *KTED* algoritması tahmin etmiştir. Kullanıcı/ürün çifti için en iyi sonuç veren algoritmanın seçilmesi durumunda sonuçlar Şekil 10.8’de gösterilmiştir. Tablo 10.19, Tablo 10.20, Tablo 10.21 ve Tablo 10.22’de eğitim kümesindeki kullanıcı sayısının değişmesiyle kullanıcı/ürün çifti için en iyi sonuç veren algoritmaların nasıl değiştiği görülmektedir.

**Tablo 10.19.** Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma  $KS=500$

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
$k$	$\bar{u}_1$	0,3815	0,4298	0,4992	0,5668	0,3038	0,4235
	$\bar{u}_2$	1,0732	0,9584	1,0068	1,0727	1,0268	1,0827
	$\bar{u}_3$	0,9438	0,9906	0,6700	1,0165	0,8693	0,9322
	$\bar{u}_4$	0,2171	0,2882	0,2337	0,3257	0,2201	0,2631
	$\bar{u}_5$	0,7544	0,7544	0,6778	0,5045	0,7544	0,3809

**Tablo 10.20.** Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma  $KS=1.000$ 

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k	$\ddot{u}_1$	0,2945	0,3791	0,4914	0,5124	0,3130	0,3677
	$\ddot{u}_2$	0,9364	0,8945	1,0534	1,0163	0,9255	0,9377
	$\ddot{u}_3$	0,7341	0,9146	0,5733	0,9748	0,7318	0,8914
	$\ddot{u}_4$	0,2141	0,3462	0,2492	0,4381	0,2195	0,3253
	$\ddot{u}_5$	0,6675	1,0362	0,7690	0,7337	0,6675	0,6169

**Tablo 10.21.** Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma  $KS=2.000$ 

Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k	$\ddot{u}_1$	0,2995	0,3876	0,4181	0,5272	0,2695	0,2508
	$\ddot{u}_2$	1,0067	0,8716	0,9330	1,0395	1,0165	0,9299
	$\ddot{u}_3$	0,6877	0,9045	0,5807	0,9716	0,7066	0,7316
	$\ddot{u}_4$	0,1008	0,2869	0,3756	0,4001	0,1092	0,1667
	$\ddot{u}_5$	0,9793	0,8818	0,9065	0,7442	0,8654	0,9346

**Tablo 10.22.** Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma  $KS=4.000$ 

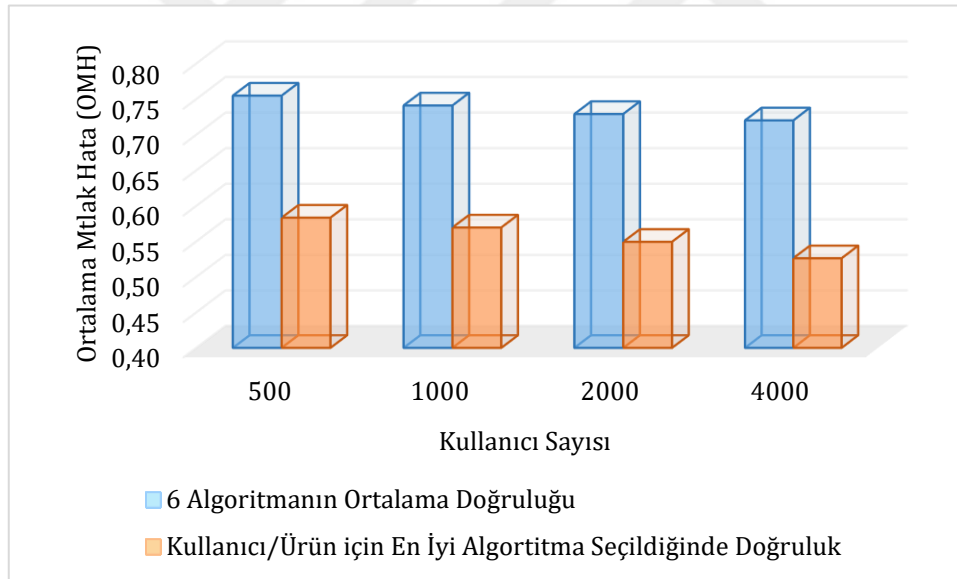
Kullanıcı	Ürün	Hafıza Tabanlı Algoritmalar		Model Tabanlı Algoritmalar		Hibrit Algoritmalar	
		KTKEYK	KTED	ÜTKEYK	ABETE	BCO	KO
k	$\ddot{u}_1$	0,1762	0,3740	0,4576	0,5254	0,1679	0,1236
	$\ddot{u}_2$	0,5850	0,7609	1,0021	0,9771	0,6701	0,8837
	$\ddot{u}_3$	0,6986	0,8999	0,5586	0,9717	0,6686	0,6921
	$\ddot{u}_4$	0,0962	0,2933	0,3939	0,4103	0,0909	0,1407
	$\ddot{u}_5$	1,0057	0,7834	0,8188	0,7262	0,9446	0,7206

Tablo 10.23' de kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde doğruluğun nasıl iyileştiği gösterilmiştir. Eğitim kümesi kullanıcı sayısı 4.000 olduğu durumda yapılan deneylerde 6 algoritmanın ortalama doğruluğu 0,7202 iken kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde doğruluğun 0,5263 olduğu görülmektedir.

**Tablo 10.23.** *Kullanıcı/Ürün Çifti İçin En İyi Algoritma Seçildiğinde OMH*

	Kullanıcı Sayısı			
	500	1000	2000	4000
6 Algoritmanın Ortalama Doğruluğu	0,7549	0,7411	0,7291	0,7202
Kullanıcı/Ürün Çifti İçin En İyi Algoritma Seçildiğinde Doğruluk	0,5835	0,5696	0,5494	0,5263

Şekil 10.8'de Tablo 10.23'de verilen sayısal değerlerin grafiği verilmiştir.



**Şekil 10.8.** *Netflix, Kullanıcı/Ürün Çifti İçin En İyi Algoritma*

Şekil 10.8'de Netflix veri kümesi için 6 algoritmanın ortalama doğruluğu ve kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiğinde ortaya çıkan doğruluk grafiği gösterilmiştir.

Kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçilmesi durumundaki doğruluğun rastlantısal bir durum olmadığını göstermek için bağımlı örneklem *t*-testi uygulanmıştır. Bağımlı örneklem *t*-testinde önerilen yöntem uygulanmadan

önceki önerilerin doğruluğu ile önerilen yöntemle elde edilen önerilerin doğruluğu karşılaştırılmıştır. Eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumlar ayrı ayrı incelenmiş ve sonuçlar Tablo 10.24'te sunulmuştur.

**Tablo 10.24.** *Kullanıcı/Ürün çifti için En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanılan 6 farklı Algoritmanın Ortalama Doğruluğu için t-test Sonuçları*

Veri Kümesi	Eğitim Kümesi Kullanıcı Sayısı							
	500		1.000		2.000		4.000	
	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri
MovieLens	93,146	0,000	149,717	0,000	155,006	0,000	135,069	0,000
Netflix	71,013	0,000	78,727	0,000	73,734	0,000	68,796	0,000

MovieLens ve Netflix kümesi için üretilen öneriler 10 farklı örnekleme bölünerek *t*-testi uygulanmıştır. Sonuç olarak *t*-testinde her iki veri kümesinde, eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumlarda  $p < 0,001$  olduğu görülmektedir. Bu sonuç %99,9 güvenle, kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiği durumdaki doğruluk ile kullanılan 6 farklı algoritmanın ortalama doğruluğu arasında çok yüksek düzeyde istatistiksel olarak anlamlı bir farklılık vardır şeklinde yorumlanabilir.

Kullanıcı/ürün çifti için en iyi sonuç veren algoritmanın seçilmesi durumunda önerilerin doğruluğundaki iyileşme, kullanıcı için en iyi sonucu veren algoritmanın seçilmesi durumundaki iyileşmeden daha fazla olmaktadır. Bu durumun rastlantısal olmadığı bağımlı örneklem *t*-testiyle gösterilmiş ve sonuç Tablo 10.25'de sunulmuştur.

**Tablo 10.25.** *Kullanıcı/ürün Çifti İçin En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk / Kullanıcı İçin En iyi Sonuç Veren Algoritma Seçildiğinde Doğruluk için t-test Sonuçları*

Veri Kümesi	Eğitim Kümesi Kullanıcı Sayısı							
	500		1.000		2.000		4.000	
	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri	<i>t</i> değeri	<i>p</i> değeri
MovieLens	100,120	0,000	157,936	0,000	137,469	0,000	114,917	0,000
Netflix	63,566	0,000	69,465	0,000	71,041	0,000	65,546	0,000

Her iki veri kümesinde üretilen öneriler 10 farklı örnekleme bölünerek *t*-testi uygulanmıştır. Sonuç olarak *t*-testinde eğitim kümesi kullanıcı sayısı 500, 1.000, 2.000 ve 4.000 olduğu durumda,  $p < 0,001$  olarak bulunmuştur. Bu sonuç %99,9 güvenle, her iki veri kümesinde kullanıcı/ürün çifti için en iyi sonuç veren algoritma seçildiği durumdaki doğruluk ile kullanıcı için en iyi sonuç veren algoritma seçildiği durumdaki doğruluk arasında çok yüksek düzeyde istatistiksel olarak anlamlı bir farklılık vardır, şeklinde yorumlanabilir.

## 11. SONUÇLAR VE DEĞERLENDİRME

Günümüzde İnternet ve iletişim teknolojilerinin gelişmesi sonucunda İnternet üzerinden çevrimiçi sunulan hizmetler popüler hale gelmiştir ve günden güne daha da önem kazanmaktadır. Bu ortamlarda sunulan ürün ve hizmetlerin artması yüksek doğrulukta çalışan öneri sistemlerine olan ihtiyacı kaçınılmaz hale getirmiştir.

Bu tez kapsamında yapılan çalışmada Ortak Filtreleme öneri sisteminin doğruluğunu artırmak amaçlanmıştır. Ortaya atılan hipotez deneysel çalışmalarla test edilmiş ve kullanıcı veya kullanıcı/ürün çiftleri için en iyi sonucu veren bir algoritmanın olduğu gösterilmiştir. Öneri üretme aşamasında kullanıcı veya kullanıcı/ürün çiftleri için en iyi sonucu veren algoritma tespit edilebilirse üretilen tahmin değerlerinin doğruluğu ile bütün öneri sisteminde bir tane algoritma kullanılması durumunda elde edilen doğruluk arasında istatistiksel olarak anlamlı bir farklılık vardır. Kullanıcı/Ürün çiftleri için en iyi sonuç veren algoritmanın tespit edilmesi durumunda doğruluk, kullanıcı için en iyi sonuç veren algoritmanın tespit edilmesine göre istatistiksel olarak daha anlamlı şekilde iyileşmektedir. Bu durum deneylerle ispatlanmış ve sayısal değerler ilgili tablolarda gösterilmiştir.

Deneysel çalışmalarda alanında en iyiler arasında gösterilen 6 farklı algoritma kullanılmıştır. Algoritmalar seçilirken hafıza, model ve hibrit olmak üzere her sınıftan ikişer tane seçilmiştir. Daha fazla algoritma seçilmesi durumunda doğruluktaki iyileşmenin daha yüksek değerlerde olup olmayacağını tespit etmek için kullanılan algoritmaların sayısı artırılmalı daha fazla algoritmanın ürettiği sonuçlara bakılarak doğruluk değerleri hesaplanmalıdır.

Gelecekte yapılabilecek çalışma “Kullanıcı” ve “Kullanıcı/ürün çiftleri” için en iyi sonucu veren algoritmanın tespit edilmesi için bir model veya sınıflandırıcı geliştirmektir. Bu çalışmada deney sonuçlarında gösterilen değerler, geliştirilecek model veya sınıflandırıcının %100 doğrulukta ve performansta çalışması durumunda elde edilecek sonuçlardır. Geliştirilecek sınıflandırıcı istenen performansta ve doğrulukta çalışmazsa burada sunulan sonuçlar yakalanamayabilir.

## KAYNAKÇA

- [1] Vijayasathy, L. R. (2004) Predicting Consumer Intentions to Use Online Shopping: The Case for an Augmented Technology Acceptance Model, *Information and Management*, 41: 747–762.
- [2] Barwise, P., Elberse, A. and Hammond, K. (2002). Marketing and the Internet: A research review (pp. 01-801). London: London Business School.
- [3] Kazienko, P., and Musiał, K. (2006). Recommendation Framework for Online Social Networks, Last M., *Advances in Web Intelligence and Data Mining*. Springer Berlin Heidelberg. 23(1), 111-120.
- [4] Ricci, F., Rokach, L. and Shapira, B. (2011). Introduction to Recommender Systems Handbook, Kantor P. B., *Recommender Systems Handbook* (1-35). Springer US.
- [5] Lamere, P. and Celma, O. (2007). Music Recommendation Tutorial, *8th International Conference on Music Information Retrieval*, ismir2007
- [6] Souvik, D., Niloy G. and Pabitra, M. (2008) Feature weighting in content based recommendation system using social network analysis. *In Proc. of the 17th International Conference on World Wide Web*, pages 1041–1042, ACM
- [7] Cacheda, F., Carneiro, V., Fernández, D. and Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, 5(1), 2. P(1-33)
- [8] Candillier, L., Meyer, F., Fessant, F. and Jack, K. (2009). State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling*, pages 1--22, 2009.
- [9] Bilge, A., Polat, H., (2013). A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. *Inform. Process. Manage.* 49 (4) 912–927.

- [10] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, New York, NY, 175–186.
- [11] Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M. and Vuduc, R. (2000). SWAMI (poster session): a framework for collaborative filtering algorithm development and evaluation, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, p.366-368
- [12] Goldberg, D., 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.
- [13] Miller, B. N., Konstan, J. A. and Riedl, J. (2004). PocketLens: toward a personal recommender system, *ACM Transactions on Information Systems*, vol. 22, no. 3, pp. 437–476.
- [14] Bilge, A., Kaleli, C., Yakut, I., Gunes, I. and Polat, H., (2013). A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, 23(8), pp. 1085–1108.
- [15] Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, pp. 109–132.
- [16] Shardanand, U. (1994). Social information filtering for music recommendation. *Doctoral dissertation, Massachusetts Institute of Technology*.
- [17] Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW'01)*. ACM, New York, NY, 285–295.
- [18] Breese, J., Heckerman, D., Kadie, C.(1998). Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman.p 43–52



- [19] Salton, G. and McGill, M. (1983). Introduction to Modern Information Retrieval. McGraw-Hill, New York.
- [20] Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2), 133-151.
- [21] Landauer, T. K. and Littman, M. L. (1994). Computerized cross-language document retrieval using latent semantic indexing, U.S. Patent, No: 5,301,109.
- [22] Ungar, L. and Foster, D. (1998). Clustering methods for collaborative filtering. In Proceedings of the Workshop on Recommendation Systems. AAAI Press, Menlo Park, CA
- [23] Kohrs, A. and M'erialdo, B. (1999). Clustering for collaborative filtering applications. In Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation
- [24] Canny, J. (2002). Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*. New York, NY, 238–245
- [25] Park, D.H., Kim, H.K., Choi, Y. and Kim, J.K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11), pp. 10059–10072.
- [26] Schein, A.I., Popescul, A., Ungar, L.H. and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*. ACM, New York, NY, 253–260.
- [27] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

- [28] Yu, K., Schwaighofer, A., Tresp, V., Xu, X. and Kriegel, H. P. (2004). Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1), 56-69.
- [29] 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 (WWW'04)*. ACM, New York, NY, 393-402
- [30] Chirita, P.A., Nejdl, W. and Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. *In Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management (WIDM'05)*. New York, NY, 67-74.
- [31] Hofmann, T. and Hartmann, D. (2005). Collaborative filtering with privacy via factor analysis. *Proceedings of the 2005 ACM Symposium on Applied Computing*, New York, USA: ACM, 791-795.
- [32] Miller, B. N., Konstan, J. A. and Riedl, J. T. (2004). PocketLens: Toward a personal recommender system. *ACM Transactions on Information Systems*, 22(3), 437-476.
- [33] Pennock, D. M., Horvitz, E., Lawrence, S. and Giles, C. L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. *In Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI '00)*, San Francisco, CA, USA, p(473-480).
- [34] Chee, S.H.S., Han, J. and Wang, K. (2001). RecTree: an efficient collaborative filtering method. *Proceedings of the Third Annual Conference on Data Warehousing and Knowledge Discovery*, Munich, September 5-7, pp. 141-151.
- [35] Rashid, A.M., Lam, S.K., Karypis, G. and Riedl, J. (2006). ClustKNN: A highly scalable hybrid model- & memory-based CF algorithm. *In Proceedings of WebKDD 2006 - Knowledge Discovery on the Web*
- [36] Acilar, A., and Arslan, A. (2009). A collaborative filtering method based on artificial immune network. *Expert Syst. Appl.* 36(4), 8324-8332.

- [37] Ju, C. and Xu, C. (2013). A New Collaborative Recommendation Approach Based on Users Clustering Using Artificial Bee Colony Algorithm. *The Scientific World Journal*, vol. 2013, 9 pages.
- [38] Cheng, J., Liu, Y., Zhang, H., Wu, X. and Chen, F. (2015). A New Recommendation Algorithm Based on User's Dynamic Information in Complex Social Network. *Mathematical Problems in Engineering*, vol. 2015
- [39] Lee, J., Sun, M. and Lebanon, G. (2012) A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*.
- [40] Yakut, I. and Polat, H. (2012). Arbitrarily distributed data-based recommendations with privacy. *Data Knowl. Eng.*, 72, pp. 239–256
- [41] Resnick, P. and Varian, H. R. (1997). Recommender Systems. *Communications of the ACM*, 40(3), 56-58.
- [42] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, J.L. and Riedl, J. (1987). GroupLens: applying collaborative filtering to Usenet news, *Communications of the ACM*, vol. 40, no. 3, pp. 77–87.
- [43] Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating “word of mouth”. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, USA: ACM Press/Addison-Wesley Publishing Co., 210-217.
- [44] Hill, W., Stead, L., Rosenstein, M. and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, USA: ACM Press/Addison-Wesley Publishing Co., 194-201.
- [45] Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J.T.(1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 22<sup>nd</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, pp. 230–237.

- [46] Liu, H., Hu, Z., Mian, A., Tian, H. and Zhu, X., (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, Volume 56, Pages 156-166.
- [47] Chen, T. and He, L. (2009). Collaborative filtering based on demographic attribute vector. *In International Conference on Future Computer and Communication (FCC'09)* .p225-229.
- [48] Wang, J., P. de Vries, A. and Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06)*. New York, NY, USA, 501-508.
- [49] Robert, M. B. and Koren, Y. (2007). Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. *In Proceedings of the 2007 Seventh IEEE International Conference on Data Mining (ICDM '07)*. IEEE Computer Society, Washington, DC, USA, 43-52.
- [50] Ma, H., King, I. and Lyu, M.R. (2007). Effective missing data prediction for collaborative filtering. *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07)*. ACM, New York, NY, USA, 39-46.
- [51] Luo, H., Niu, C., Shen, R. and Ullrich, C. (2008). A collaborative filtering framework based on both local user similarity and global user similarity. *Mach. Learn.* 72, 3, 231-245.
- [52] Polatidis, N. and Georgiadis, C. K. (2016). A multi-level collaborative filtering method for improved recommendations. *Expert Systems with Applications* Volume 48, 15 April 2016, Pages 100–110
- [53] Koohi, H. and Kiani, K. (2016) User based Collaborative Filtering using fuzzy C-means. *Measurement*, Volume 91, September 2016, Pages 134–139

- [54] Kaleli, C. (2014) An entropy-based neighbor selection approach for collaborative filtering. *Knowledge-Based Systems*, Volume 56, Pages 273–280
- [55] Vozalis, E. G. and Margaritis, K. G. (2003). Recommender systems: An experimental comparison of two filtering algorithms. In Proceedings of the 9th Panhellenic Conference in Informatics - PCI 2003.
- [56] Huang, Z., Zeng, D. and Chen, H. (2007). A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce. *IEEE Intelligent Systems* 22, 5 (September 2007), 68-78.
- [57] Lemire, D. & Maclachlan, A. (2005). Slope One predictors for online rating-based collaborative filtering. *Proceedings of the SIAM Data Mining*, Newport Beach, CA, USA, pp. 471–475.
- [58] Terzi, I., Polat, H. (2016), Privacy-preserving Normalized Ratings-based Weighted Slope One Predictor, *International Journal of Design & Nature and Ecodynamics*, Volume 11, Issue 3 p(284 – 294)
- [59] Bilge, A. and Polat, H. (2013). A comparison of clustering-based privacy-preserving collaborative filtering schemes. *Appl. Soft Comput.* 13, 5, 2478-2489.
- [60] Avazpour, I., Pitakrat, T., Grunske, L., and Grundy, J. (2014). Dimensions and metrics for evaluating recommendation systems. In Recommendation Systems in Software Engineering (pp. 245-273).

## ÖZGEÇMİŞ

Adı-Soyadı : İsmail TERZİ

Yabancı Dil : İngilizce

Doğum Yeri ve Yılı : Devrek/25.06.1982

E-Posta : iterzi@anadolu.edu.tr

### Eğitim ve Mesleki Geçmişi:

- 2007 Orta Doğu Teknik Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü (Çift Anadal)
- 2004, Orta Doğu Teknik Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği
- 2017, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
- 2014, Araştırma Görevlisi, Bülent Ecevit Üniversitesi, Bilgisayar Mühendisliği Bölümü
- 2014, Araştırma Görevlisi, Anadolu Üniversitesi, Bilgisayar Mühendisliği Bölümü
- 2004-2013 Milli Eğitim Bakanlığı Devrek Hamidiye Anadolu Lisesi, Bilişim Teknolojileri Öğretmeni