

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**İŞBİRLİKÇİ FİLTRELEME YÖNTEMLERİNİN GELİŞTİRİLEN
BİR UYGULAMA İLE KARŞILAŞTIRILMASI**

GİZEM ZEYNEP PARİM

KOCAELİ 2018

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ


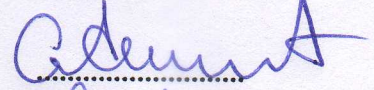
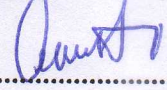
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

İŞBİRLİKÇİ FİLTRELEME YÖNTEMLERİNİN
GELİŞTİRİLEN BİR UYGULAMA İLE KARŞILAŞTIRILMASI

GİZEM ZEYNEP PARİM

Prof. Dr. Nevcihan DURU
Danışman, Kocaeli Üniversitesi
Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi
Prof. Dr. Cemil ÖZ
Jüri Üyesi, Sakarya Üniversitesi


.....

.....

.....

Tezin Savunulduğu Tarih: 11.07.2018

ÖNSÖZ VE TEŞEKKÜR

Öneri sistemleri günümüzde aktif araştırma alanlarından biridir. Seçimlerde alternatiflerin çoğalması, kişinin karar verme aşamasını zorlaştırabilir. Öneri sistemleri ise bu alanda kişinin geçmişteki tercihleri, topluluğun tercihleri vb. durumları göz önünde bulundurarak kişinin seçim yapmasını kolaylaştırmayı sağlar. Bu tez çalışması, işbirlikçi filtreleme ile oluşturulan öneri sistemlerinde kullanılan yöntemleri karşılaştırmak amacı ile gerçekleştirilmiştir.

Tez çalışmamda yardımlarını esirgemeyen ve bana her konuda destek olan değerli hocam Prof. Dr. Nevcihan DURU'ya, aileme ve araştırma aşamasında bana yardımcı olan herkese teşekkür ederim.

Haziran – 2018

Gizem Zeynep PARİM

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
TABLolar DİZİNİ	iv
SİMGELER VE KISALTMALAR DİZİNİ	v
ÖZET.....	vi
ABSTRACT	vii
GİRİŞ	1
1. GENEL BİLGİLER	3
1.1. Tez Çalışmasının Amacı	3
1.2. Literatür Taraması	3
1.3. Tezin Yapısı	7
2. ÖNERİ SİSTEMLERİ	9
2.1. Giriş	9
2.2. İşbirlikçi Filtreleme	10
2.2.1. İşbirlikçi filtrelemede karşılaşılan problemler	13
2.2.2. İşbirlikçi filtrelemede değerlendirme	14
2.3. İçerik Tabanlı Filtreleme	15
2.4. Hibrit Yaklaşım	16
3. VERİ ANALİZİ	18
3.1. Giriş	18
3.2. Çalışma Grubu.....	18
3.3. Kullanılan Verinin Analizi	19
4. ÖNERİ SİSTEMLERİNDE İŞBİRLİKÇİ FİLTRELEME	24
4.1. Giriş	24
4.2. Pearson Korelasyonu ile Öneri Sistemi Oluşturma	24
4.3. Kosinüs Benzerliği ile Öneri Sistemi Oluşturma	27
4.4. Matris Faktörizasyonu ile Öneri Sistemi Oluşturma	30
4.5. Sınırlı Pearson Korelasyonu ile Öneri Sistemi Oluşturma	37
4.6. Azure Machine Learning Platformu ile İşbirlikçi Filtreleme	40
4.6.1. Azure machine learning ile öneri sistemi oluşturma	41
5. ÖNERİ SİSTEMLERİNDE İŞBİRLİKÇİ FİLTRELEMENİN UYGULANMASI VE KARŞILAŞTIRILMASI.....	46
5.1. Giriş	46
5.2. İşbirlikçi Filtreleme ile Öneri Sistemi Oluşturulması.....	46
6. BULGULAR VE TARTIŞMA	54
7. SONUÇLAR VE ÖNERİLER	59
KAYNAKLAR	61
EKLER.....	70
KİŞİSEL YAYINLAR VE ESERLER	76
ÖZGEÇMİŞ	78

ŞEKİLLER DİZİNİ

Şekil 3.1. Ankette yer alan oylama arayüzünden bir kesit	18
Şekil 3.2. Eğitim düzeylerine göre elde edilen dağılım	21
Şekil 3.3. Katılımcılara ait yaş dağılımı	22
Şekil 3.4. Boğaz Turu ögesine ait kullanıcı oyları	23
Şekil 4.1. Pearson korelasyonu ile elde edilen ekran görüntüsü	26
Şekil 4.2. Gezilebilecek 50 yere ait oy ortalaması	27
Şekil 4.3. Kosinüs benzerliği ile elde edilen ekran görüntüsü	29
Şekil 4.4. Matris faktörizasyonu ile elde edilen ekran görüntüsü	35
Şekil 4.5. MovieLens verileri kullanıldığında elde edilen hata değerlerinin gösterimi.....	36
Şekil 4.6. Farklı K değerlerine bağlı olarak elde edilen hata değerleri	37
Şekil 4.7. Sınırlı Pearson korelasyonu ile elde edilen ekran görüntüsü	40
Şekil 4.8. AML üzerinde öneri sistemi temel çalışma prensibi	41
Şekil 4.9. Elde edilen öneriler	43
Şekil 4.10. Oluşturulan öneri sistemi	44
Şekil 5.1. Kosinüs benzerliği ile 100 numaralı kişi için elde edilen öneriler.....	47
Şekil 5.2. Girilen oy değerlerine göre elde edilen öneriler	49

TABLolar DİZİNİ

Tablo 2.1. Kullanıcı ve nesne oy tablosu	10
Tablo 2.2. Nesne tabanlı işbirlikçi filtreleme.....	12
Tablo 3.1. Katılımcıların oyladığı yerlerin listesi	19
Tablo 3.2. Kullanıcıdan istenen kişisel özellik soruları	20
Tablo 3.3. En çok ziyaret edilen ilk 10 yer	22
Tablo 3.4. En yüksek ortalama oya sahip ilk 10 yer	23
Tablo 4.1. Rastgele seçilen kullanıcılara ait oy değerleri	25
Tablo 4.2. Gradient Descent ile elde edilen oy değerleri	33
Tablo 4.3. P matrisini oluşturan değerler	33
Tablo 4.4. Q matrisini oluşturan değerler	33
Tablo 4.5. Matris bölündüğünde elde edilen oy değerleri	34
Tablo 4.6. MovieLens veri setinde yer alan oy değerleri.....	36
Tablo 5.1. Diğer yöntemlerle 100 numaralı kişi için elde edilen öneriler	48
Tablo 5.2. Rastgele giriş verilerine göre elde edilen öneriler	50
Tablo 5.3. Öneride bulunurken geçen süre	50
Tablo 5.4. Oy verilen nesnelere gizlendiğinde elde edilen veriler	51
Tablo 5.5. Rastgele seçilen kişilere ait oluşturulan öneriler	52
Tablo 6.1. Geleneksel yöntemler ile elde edilen hata değerleri	54
Tablo 6.2. Gezi öneri sisteminde SGD ile elde edilen hata değerleri	55
Tablo 6.3. MovieLens ile elde edilen hata değerleri.....	55
Tablo 6.4. MovieLens verilerinde SGD ile elde edilen hata değerleri.....	56
Tablo 6.5. AML üzerinde oy tahmini ile elde edilen değerler.....	56
Tablo 6.6. AML üzerinde nesne önerisi ile elde edilen değerler.....	57
Tablo 6.7. MovieLens ile oy tahmininde elde edilen hata değerleri.....	57
Tablo 6.8. Nesne önerisi ile elde edilen değerler	57

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

AML	: Azure MachineLearning (Azure Makine Öğrenmesi)
ID	: Identifier (Tanımlayıcı değer)
IDF	: Inverse Document Frequency (Ters Doküman Frekansı)
CTR	: Click Through Rate (Tıklama Oranı)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
NDCG	: Normalized Discounted Cumulative Gain (İndirgenmiş Normalize Birikimli Kazanç)
NMF	: Non-negative Matrix Factorization (Negatif Olmayan Matris Faktörizasyonu)
RBM	: Restricted Boltzmann Machines (Kısıtlı Boltzmann Makineleri)
RMSE	: Root Mean Squared Error (Ortalama Karesel Hatanın Karekökü)
SGD	: Stochastic Gradient Descent (Stokastik Meyilli Azalım)
SSE	: Sum of Squared Errors (Karesel Hatanın Toplamı)
SVD	: Singular Value Decomposition (Tekil Değer Ayrışımı)
TF	: Term Frequency (Terim Frekansı)

İŞBİRLİKÇİ FİLTRELEME YÖNTEMLERİNİN GELİŞTİRİLEN BİR UYGULAMA İLE KARŞILAŞTIRILMASI

ÖZET

Günümüzde alternatiflerin çoğalması kişilerin karar vermesini oldukça zorlaştırmaktadır. Öneri sistemleri ise karar verme aşamasında, kişilerin geçmiş alışkanlıklarını göz önünde bulundurarak kişiye uygun seçim yapmayı oldukça kolaylaştırmıştır. Öneri sistemlerinin temel amacı olan doğru öneride bulunmak için yürütülen çalışmalar günümüzde de devam etmektedir. Gerçekleştirilen çalışmanın amacı ise işbirlikçi filtreleme ile oluşturulan öneri sistemlerinde kullanılan yöntemlerin, sistemdeki doğru tahmin oranına olan etkisini ölçmektir.

Bu çalışmada Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği, matris faktörizasyonu yöntemlerinden biri olan Stochastic Gradient Descent ve veri bilimi platformlarından Azure Machine Learning kullanılmış ve elde edilen sonuçlar karşılaştırılmıştır. Elde edilen sonuçlara göre ayrıca bir öneri sistemi tasarlanmıştır.

Anahtar Kelimeler: İşbirlikçi Filtreleme Tekniği, Kosinüs Benzerliği, Matris Faktörizasyonu, Pearson Korelasyonu, Öneri Sistemleri.

COMPARISON OF COLLABORATIVE FILTERING METHODS WITH AN APPLICATION DEVELOPED

ABSTRACT

Today, the multiplication of alternatives makes it very difficult for people to make decisions. Recommender systems have made it easier to make the right choice for the person with taking into account the past habits of the person in the decision making stage. The researches carried out in order to make the right suggestion which is the main aim of the recommender systems continues today. The aim of the study is to measure the effect of the methods used in the recommender systems created with collaborative filtering on the correct prediction rate in the system.

In this study, Pearson correlation, constrained Pearson correlation, cosine similarity, Stochastic Gradient Descent which is one of the matrix factorization methods and Azure Machine Learning which is one of the data science platforms were used and the results obtained were compared. According to the results obtained, a recommendation system is also designed.

Keywords: Collaborative Filtering Technique, Cosine Similarity, Matrix Factorization, Pearson Correlation, Recommender Systems.

GİRİŞ

Günümüzde kişiler ne izlemeli, ne okumalı gibi sorularına yanıt ararken büyük ölçüde interneti kullanmaktadır. Kişilerin ilgilenebilecekleri ürünleri sunan bu sistemlere öneri (tavsiye) sistemi denmektedir.

Öneri sisteminin temeli sayılabilecek ilk sistem, Grundy adı verilen bir kütüphane görevlisi oluşturma amacıyla gerçekleştirilmiştir. Bu yazılımda, kullanıcılara ilk olarak isimleri sorulup kişi bir cinsiyet kalıbına dâhil edilir. Sonrasında ise kişiden kendini tanıtan sözcükler yazması istenir. Kişinin vermiş olduğu cevaplara göre kişi, Hristiyan, Yahudi, Protestan, Katolik, sportif kişilik, feminist, eğitilmiş kişi vb. kalıplardan uygun olanlarına dâhil edilir ve sistem bu kalıba uyabilecek kitap önerilerinde bulunur [1]. Ekstrand vd. bu yazılımın ilkel olduğunu ancak bu sistemlerin temelini oluşturması açısından da önemli olduğunu belirtmişlerdir [2].

Günümüzde özellikle e-ticaret sitelerinin kullandığı öneri sistemlerini Netflix dizi veya film önerirken, sosyal medya siteleri arkadaş önerirken, Udemy ve YouTube video önerirken, Spotify şarkı önerirken, Amazon.com ise ürün önerirken kullanmaktadır.

Öneri sistemlerinin temel amacı kullanıcılara uygun bir öneride bulunmaktır. Bunu yapmak için genellikle kullanıcı görüşlerinden faydalanılır. Öneri sistemleri oluşturulurken pek çok yöntem kullanılır. Burke, kullanılan yöntemleri işbirlikçi filtreleme (collaborative filtering), içerik tabanlı filtreleme (content-based filtering), fayda tabanlı (utility-based), bilgi temelli (knowledge-based) ve demografik (demographic) olmak üzere beş ana başlık altında toplanabileceğini belirtmiştir [3]. Ayrıca çalışmasında günümüzde yaygın olarak kullanılan hibrit yaklaşım ile öneri sistemi oluşturma hakkında da bilgi vermiştir. Kimi çalışmalarda, bu yöntemlere ek olarak veri madenciliği tekniği ile oluşturulan öneri sistemleri ve içerik farkında (context-aware) oluşturulan öneri sistemleri de yer almaktadır [4].

Gerçekleştirilen tez çalışmasında, öneri sistemleri oluşturulurken günümüzde yaygın olarak kullanılan işbirlikçi filtreleme tekniği, içerik tabanlı filtreleme tekniği ve hibrit yaklaşım yöntemlerinden bahsedilmiştir.

Öneri sistemi oluştururken kullanıcı oylarının temel alındığı yönteme işbirlikçi filtreleme denir. İşbirlikçi filtrelemede kullanıcı görüşlerinin yanı sıra kullanıcılara ait demografik veriler de göz önünde bulundurulabilir [5]. İşbirlikçi filtrelemede kullanıcı oyları ile benzer kullanıcılar bulunur ve benzer kullanıcının sevdiği bir nesne asıl kullanıcıya önerilir. Benzer kullanıcının bulunması için farklı hesaplama yöntemleri mevcuttur. Kullanıcıların hangi hesaplama yöntemini kullanacağı ile ilgili bir sınırlama yoktur.

Öneriler her zaman kullanıcı görüşlerine dayalı olarak yapılmayabilir, kimi zaman da nesne içeriğine bağlı olarak yapılır. Haber metinleri için içeriğe bağlı olarak bir öneri sistemi geliştirilecekse esas olan doküman benzerliğidir [6]. İçeriğe bağlı olarak bir film öneri sistemi gerçekleştirilmek istendiğinde ise oyuncu adları, yönetmen adı, filmin türü gibi içerikler dikkate alınır. Bu yönteme ise içerik tabanlı filtreleme denmektedir.

Öneri sistemlerinde daha doğru sonuçlar elde edebilmek için işbirlikçi filtreleme ve içerik tabanlı filtrelemenin birleştirilebileceği belirtilmiştir. Bu yöntemlerin birleştirilmesi ile oluşan yeni yönteme ise hibrit yaklaşım denir.

Bu çalışmada işbirlikçi filtrelemede kullanılan benzerlik hesaplama yöntemlerinin, öneri sistemindeki doğruluk oranına etkisi araştırılmak istenmiştir. İşbirlikçi filtrelemede sık rastlanan Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve matris faktörizasyonu yöntemlerinden Stochastic Gradient Descent (SGD) ile Azure Machine Learning (AML) platformunda yer alan öneri sistemi oluşturma seçeneği ile öneri sistemi oluşturulmuştur. Oluşturulan sistemlerin doğruluk oranları karşılaştırılmış ve bu sistemlerde kullanılan benzerlik hesaplama yöntemlerinin doğruluk oranına nasıl etki ettiği gözlemlenmiştir.

1. GENEL BİLGİLER

1.1. Tez Çalışmasının Amacı

Tez çalışmasında, günümüzde birçok alanda kullanılan öneri sistemlerinden işbirlikçi filtreleme yöntemi ele alınmıştır. Öneri sistemlerinde amaç, kullanıcılara ilgilendiği nesneye göre uygun olabilecek yeni bir nesne önerisinde bulunmaktır. Bu öneri, kullanıcı veya nesne benzerliğine bağlı olarak yapılabilir. Bu tez çalışmasında kullanıcı benzerlikleri göz önünde bulundurularak bir öneri sistemi gerçekleştirilmek istenmiştir. İşbirlikçi filtreleme ile oluşturulan bir sistemde kullanıcıların benzerliğini hesaplayabilmek için birden fazla hesaplama yöntemi bulunmaktadır. Geliştirilen öneri sistemine bağlı olarak kişi, yüksek doğruluk elde edilebilecek hesaplama yöntemini seçmelidir. Gerçekleştirilen çalışmada, bu hesaplama yöntemlerinin oluşturulacak öneri sistemine nasıl etki edeceği gözlemlenmek istenmiştir. Ayrıca veri bilimi platformlarından biri ile öneri sistemi oluşturulup geleneksel hesaplama yöntemlerine karşı başarısı ölçülmek istenmiştir. Yapılacak öneri sisteminde kullanılacak oy verileri, online anket aracılığı ile elde edilen İstanbul ili içerisinde ziyaret edilebilecek yerlere ait kullanıcı görüşlerini bildiren oy verilerinden oluşmaktadır.

Bu çalışmada, geleneksel hesaplama yöntemlerinden Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve matris faktörizasyonu yöntemlerinden SGD; veri bilimi platformlarından ise Azure Machine Learning kullanılarak elde edilen veriler ile öneri sistemleri oluşturulup başarı oranları karşılaştırılmıştır.

1.2. Literatür Taraması

İşbirlikçi filtreleme ile gerçekleştirilen çok sayıda öneri sistemi mevcuttur. İşbirlikçi filtrelemede genel olarak kabul görmüş tek bir yöntem olmadığı için oluşturulan sistemlerde farklı hesaplama yöntemlerinin kullanıldığı gözlemlenmiştir. Bu sistemler incelendiğinde yaygın olarak Pearson korelasyonu, kosinüs benzerliği ve matris faktörizasyonu işlemlerinin kullanıldığı görülebilir.

Matris faktörizasyonu işleminin gerçekleştirilmesi için birden fazla yöntem bulunur. Matris faktörizasyonu yönteminin kullanıldığı uygulamalar incelendiğinde genellikle Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), Alternating Least Squares (ALS) ve Stochastic Gradient Descent (SGD) yöntemlerinin kullanıldığı görülür.

İşbirlikçi filtreleme tekniğinde öneri sistemleri kullanıcıların tüm nesnelere verdikleri oy değerleri üzerinden oluşturulduğunda kullanıcı tabanlı, nesnelere ait tüm kullanıcıların vermiş olduğu oylar ile oluşturulduğunda ise nesne tabanlı olarak adlandırılır. Amazon.com, nesne tabanlı işbirlikçi filtrelemenin farklı bir versiyonunu kullandıklarını belirtmişlerdir. Kullandıkları bu filtreleme tekniğinde hesaplama yöntemi olarak kosinüs benzerliğini tercih etmişlerdir [7].

Tian vd. güvene dayalı olarak önerdikleri çalışmalarında kullanıcı benzerliğini ölçmek için kosinüs benzerliği yöntemini kullanmışlardır [8]. Lvn vd. haber metinleri önerisinde bulunmak için kosinüs benzerliği yöntemini kullanmışlardır [9]. Fakat oy değerleri yerine binary değerler üzerinden hesaplamaları gerçekleştirmişlerdir.

Singh vd. film öneri sistemi üzerinde çalışmışlardır [10]. Yaptıkları çalışmada benzerlik hesaplama yöntemi olarak Pearson korelasyonunu kullanmışlardır. Bu çalışmada oy değerlerinin yanı sıra film türleri arasında da Pearson korelasyonu yöntemini uygulamışlardır. Burada oy değerleri kullanıldığında korelasyonu 1 olan kullanıcıları seçip film türlerine bağlı olarak da Pearson korelasyonu hesabı yapıp elde edilen korelasyon sonuçları karşılaştırılmıştır. Bu iki korelasyonun beraber kullanıldığında daha iyi sonuçlar verebileceğini belirtmişlerdir.

Netflix, kendi öneri sistemlerinde birden fazla algoritmayı kullanmaktadır. Burada kullanılan algoritmaların sayısı tam olarak bilinmemektedir. Fakat kullandığı algoritmalarından birinin, matris faktörizasyonu yöntemlerinden biri olan SVD algoritması olduğu bilinmektedir [11]. Netflix yarışması devam ederken BellKor's takımı 2008 yılında bulgularını paylaşmış ve çalışmalarında matris faktörizasyonu yöntemleriyle beklenenden az da olsa geliştirme kaydettiklerini paylaşmışlardır [12].

Literatürde işbirlikçi filtreleme ile gerçekleştirilen öneri sistemlerinin karşılaştırıldığı çalışmalar mevcuttur. Lee vd. Netflix veri setini kullanarak kullanıcı tabanlı, nesne tabanlı ve SVD, Non-negative Matrix Factorization (NMF) yöntemlerini karşılaştırmış ve çoğunlukla matris faktörizasyonu yönteminin başarılı olduğunu fakat bu yöntem kullanılmadan önce bazı parametrelerde ayarlama yapılması gerektiğini belirtmişlerdir [13]. Ayrıca kullanıcı ve nesne tabanlı olarak yapılan hesaplamalarda da sistemin yavaş çalıştığını belirtmişlerdir.

Cacheda vd. kullanıcı veya nesne benzerliklerinin hesaplanması yerine eğilim veya farklılıklarının belirlenmesini önermişlerdir [14]. Çalışmalarında bellek tabanlı olarak kullanılan yöntemlerin yoğun matrislerde model tabanlı yöntemlerden daha iyi sonuçlar verdiğini belirtmişlerdir. Ayrıca çalışmalarında bazı algoritmaları karşılaştırmış ve SVD yöntemi ile olumlu sonuçlar elde ettiklerini belirtmişlerdir.

Al-Bakri vd. film verilerini kullandıkları çalışmada işbirlikçi filtrelemede kullanılan Pearson korelasyonu, kosinüs benzerliği, sınırlı Pearson korelasyonu, Jaccard uzaklığı benzerlik hesaplama yöntemleri ile kullanıcı benzerliklerini hesaplamıştır [15]. Her bir filmi izleyen kişilerin sayısı ile filmlere ait kullanıcı frekansını da hesaplamışlardır. Elde ettikleri veriler doğrultusunda yeni bir hesaplama yöntemi önerisinde bulunmuşlardır. Yeni yöntem ile işbirlikçi filtrelemedeki doğruluk oranını arttırmak istemişlerdir. Bu yöntemde benzerlik hesaplaması için kullanılan sınırlı Pearson korelasyonu yöntemine kullanıcı ağırlığı verisini çarpan olarak eklemişlerdir. Değiştirdikleri sınırlı Pearson yöntemini ayrıca Jaccard benzerliği ile çarparak elde edilen sonucu yeni benzerlik katsayısı olarak sunmuşlardır. Bu yeni benzerlik hesabı yöntemi ile her bir çift kullanıcı için oluşturulan benzerlik değeri ile tüm kullanıcıların karşılaştırılabilir olduklarını belirtmişlerdir.

Al-Barznji vd. çalışmalarında büyük veri kullanarak kullanıcı tabanlı ve nesne tabanlı işbirlikçi filtrelemeyi karşılaştırmak istemişlerdir [16]. Kullanıcılar arasındaki benzerliği hesaplarken Pearson korelasyonu ve Log Likelihood yöntemlerini kullanmışlardır. Bu yöntemlerle elde edilen sonuçları karşılaştırmışlardır. Sonuç olarak Pearson korelasyonunun daha nitelikli olduğunu ve çalışmalarında nesne tabanlı olarak daha olumlu sonuçlar elde ettiklerini belirtmişlerdir.

Sincan vd. çalışmalarında film veri setini kullanarak işbirlikçi filtrelemede yeni bir benzerlik hesaplama şekli önerisinde bulunmuştur [17]. Önerdikleri benzerlik hesaplama yöntemini, Pearson korelasyonu, kosinüs benzerliği ve uzaklığa bağlı mesafe ile elde edilen değerler ile karşılaştırmışlardır. Önerdikleri bu benzerlik katsayısının uzaklığa bağlı mesafede elde edilen değerden daha üstün olduğunu ve diğer yöntemlerden elde edilen değerlere de yakın değerler elde ettiğini belirtmişlerdir.

Li vd. çalışmalarında işbirlikçi filtrelemeye oy değerleri için kullanıcı alışkanlıklarını da eklemeyi önermişlerdir [18]. Bunun için oy değerleri ile entropi ve Bhattacharyya katsayısını kullanarak film veri seti ile ölçümler gerçekleştirmişlerdir. Buradan elde ettikleri değerleri kosinüs benzerliği, Pearson korelasyonu ve sınırlı Pearson korelasyonu yöntemleri ile karşılaştırmışlardır. Oluşturdukları yeni yöntemin diğer yöntemlere kıyasla daha başarılı olduğunu belirtmişlerdir.

Dhawan vd. işbirlikçi filtreleme tekniği ile sosyal ağlarda bir ürün öneri sistemi tasarlamışlardır [19]. Bu çalışmada, hesaplama tekniği için farklı bir öneride bulunmuşlardır. Yaptıkları çalışma ile önerdiklerini hesaplama yönteminde kullanıcı oyları ile kullanıcılar arasındaki benzerliği hesaplarken Jacobian ve kosinüs benzerliği yöntemlerini kullanmayı ve oy tahmini yapılırken de Pearson korelasyonu yöntemini kullanmayı önermişlerdir. Önerdikleri bu yöntemin öneri sistemlerinde karşılaşılan temel sorunlardan biri olan seyreklik sorununun da üstesinden gelmeye yardımcı olabileceğini belirtmişlerdir.

Gezi konulu öneri sistemleri incelendiğinde, gezilerin çeşitli niteliklerine ve farklı ölçütlere göre geliştirildiği gözlenmiştir. Menk, hazırlamış olduğu Brezilya seyahat öneri sisteminde kişilerin merak seviyesine göre öneride bulunmak istemiştir [20]. Kişilerin merak seviyesini ölçmek için Merak ve Keşif Envanteri (CE I-II) [21] testini uygulamıştır.

Uçar, gerçekleştirdiği seyahat öneri sisteminde J48, Naive Bayes vb. algoritmaları ile sınıflandırma yöntemini kullanarak öneride bulunmuştur [22]. Kişilerden süre, mevsim, cinsiyet, otel ve uçak aralığı fiyat bilgileri istemiş ve sonuç olarak dâhil oldukları sınıfa göre bölge, havayolu firması ve gezi süresi önerisinde bulunmuştur.

Ge vd. turizm verilerinden yola çıkarak matris faktörizasyonu ile maliyet farkında bir sistem üzerinde çalışmışlardır. İlerideki çalışmalarında zaman özelliği üzerine de çalışarak bu durumun sistemi nasıl etkileyebileceğini de araştıracaklarını belirtmişlerdir [23].

Ravi vd. çalışmalarında seyahat öneri sistemini etkileyebilecek faktörlere değinmişlerdir. İklim, trafik, bölgenin güvenliği, araç park olanağı, kullanıcı yorumları, ziyaret edilecek yerin açılış ve kapanış saatinin de bir etken olabileceğini belirtmişlerdir [24].

Herzog vd. bölgeler üzerinde beşli likert ölçeği ile suç oranı, aylık olarak tavsiye edilen ziyaret zamanı, ortalama kalış süresi, günlük ortalama harcama miktarına dayanan bir öneri sistemi üzerine çalışmışlardır [25]. Hao vd. kişilerin seyahatleri ile ilgili yaptıkları yorum yazılarından bilgi çıkarımında bulunarak seyahat temelli bir öneri sistemi tasarlamışlardır [26]. Kişilerin yazdıkları yazılardan yer adı, otel, sahil, havaalanı, servis vb. terimleri ayıklamış, TF-IDF yöntemi ile gidilecek yer tavsiyesinde bulunulabileceğini belirtmişlerdir.

1.3. Tezin Yapısı

Tez 6 bölümden oluşmaktadır. İlk bölümde tez çalışmasının neden gerçekleştirildiği ve literatürde yer alan çalışmaların neler olduğundan bahsedilmiştir.

İkinci bölümde öneri sistemlerine giriş yapılmıştır. Öneri sisteminin ne olduğu ve en yaygın kullanılan işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımdan bahsedilmiştir. Tezde gerçekleştirilen çalışma işbirlikçi filtreleme ile oluşturulmuştur. Bu sebeple işbirlikçi filtrelemede kullanılan hesaplama yöntemlerine de değinilmiştir. Bu bölümde ayrıca işbirlikçi filtrelemede karşılaşılan problemlerden de bahsedilmiştir.

Üçüncü bölümde, kullanılan gezi veri seti hakkında bilgi verilmiştir. İlk olarak verilerin nasıl elde edildiği ve toplamda kaç adet kullanıcı ve oy değeri bulunduğu bilgisi verilmiştir. Daha sonra ise ankete katılan kişilerin, kişisel özelliklerine göre elde edilen sayısal veriler belirtilmiştir.

Dördüncü bölümde işbirlikçi filtrelemede popüler olarak kullanılan benzerlik hesaplama yöntemlerinden Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve matris faktörizasyonu yöntemlerinden biri olan SGD ile öneri sistemleri oluşturma hakkında bilgi verilmiş ve bu yöntemler ile öneri sistemi oluşturulduğunda elde edilen hata verileri, doğru tahmin oranı bilgileri paylaşılmıştır. Ayrıca bu bölümde veri bilimi platformlarından biri olan AML hakkında bilgi verilmiş ve burada bir öneri sisteminin nasıl oluşturulabileceği gösterilmiştir.

Beşinci bölümde, daha önceki bölümlerde anlatılan Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve Stochastic Gradient Descent ile oluşturulan matris faktörizasyonu yöntemleri kullanılarak oluşturulan masaüstü uygulaması anlatılmıştır.

Son bölümde ise diğer bölümlerden oluşturulan tüm sistemlerden elde edilen hata değerleri, doğru tahmin oranları karşılaştırılmış ve bu karşılaştırma sonuçlarından yola çıkılarak öneriler dile getirilmiştir.

2. ÖNERİ SİSTEMLERİ

Öneri sistemleri oluştururken en çok tercih edilen teknikler incelendiğinde işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımın öne çıktığı gözlenmiştir. Bu bölüm içerisinde, bu tekniklerden bahsedilmiştir.

2.1. Giriş

Öneri sisteminin temeli sayılabilecek ilk sistem, Grundy adı verilen bir kütüphane görevlisi oluşturma amacıyla gerçekleştirilmiştir. Bu yazılımda kullanıcılara sorulan sorular doğrultusunda kullanıcılar, cinsiyet, dini inanç vb. kalıplardan uygun olanlarına dâhil edilmiş ve belirlenen kalıplara göre öneride bulunma amaçlanmıştır [1]. Ekstrand vd. bu yazılımın günümüzde kullanılan öneri sistemleri ile karşılaştırıldığında oldukça ilkel olduğunu fakat bu sistemlerin temelini oluşturması açısından önemli olduğunu belirtmişlerdir [2].

Öneri sistemlerinde amaç kullanıcıya en uygun olan nesneyi bulup önermektir. Bunun için öneri sistemi dikkatli bir şekilde hazırlanmalıdır. Amazon.com, sistemlerinde meydana gelen bir problem sebebi ile hiç çocuğu olmayan kişilere çocukları için alabilecekleri bebek bezi önerisinde bulunmuştur [27]. Ayrıca sıkça birlikte alınan ürün önerisinde de bomba yapım malzemelerini sunmuştur [28].

Kullanıcılarına dizi ve film içeriği sunan Netflix platformu 2006 yılında sundukları önerileri geliştirmek için bir yarışma başlatmış, hata oranlarında %10 oranında iyileştirme sağlayana 1 milyon dolar ödül verileceğini duyurmuştur. Yarışmayı 2009 yılında %10,06 oranında bir geliştirme ile BellKor's Pragmatic Chaos takımı kazanmıştır [29]. Yarışmayı 2. sırada tamamlayan The Ensemble takımı, ilk sırada yer alan BellKor's Pragmatic Chaos ile aynı oranda geliştirme sağlamış fakat çalışmalarını BellKor's Pragmatic Chaos takımından 20 dakika geç gönderdikleri için kazanamamışlardır [29].

Netflix, düzenlediği yarışmada en iyi sonuçlar veren algoritmalara baktığında bir matris faktörizasyonu yöntemi olan SVD ve Restricted Boltzmann Machines (RBM) algoritmaları karşılına çıkmıştır. Ayrıca, SVD ve RBM yöntemlerinin beraber kullanıldığında daha olumlu sonuçlar elde edildiğini görmüşlerdir [11].

Öneri sistemleri üzerinde yapılan çalışmalar günümüzde halen devam etmektedir. Kişilere daha iyi öneri sunmak için ontoloji tabanlı öneri sistemleri [30 – 32], duygu analizine dayalı sistemler [33] vb. konular üzerinde çalışmalar yapılmaktadır.

2.2. İşbirlikçi Filtreleme

İşbirlikçi filtreleme, öneri sistemlerinde yaygın olarak kullanılan yöntemlerden biridir. İşbirlikçi filtrelemede çoğunlukla kullanıcılardan elde edilen oy değerlerine göre birbirine benzer diğer kullanıcılar bulunup bu doğrultuda ürün önerisi gerçekleştirilir. Temelinde “Eğer 2 kullanıcı birbirlerine benziyor ise birinin sevdiği ürünü diğeri de sevebilir.” düşüncesi yer alır. Tablo 2.1’de k oy veren kişiler, n oy verilen nesnelere olmak üzere oy veren kişi ve oy verilen nesneye ait oy değerleri tablosu gösterilmiştir.

Tablo 2.1. Kullanıcı ve nesne oy tablosu

Nesneler Kullanıcılar	n_1	n_2	n_{i-1}	n_i
k_1					
k_2					
.					
.					
.					
k_{j-1}					
k_j					

İşbirlikçi filtreleme tekniği, model tabanlı (model based) ve bellek tabanlı (memory based) olmak üzere iki ana dal altında incelenebilir.

Model tabanlı yöntemde ilk olarak bir model oluşturulup öneriler, bu model üzerinden sağlanır. Model tabanlı işbirlikçi filtreleme tekniği Netflix yarışmasından sonra büyük bir ivme kazanmıştır [34]. Bu yöntemde çeşitli makine öğrenmesi ve veri madenciliği tekniği kullanılır. Model tabanlı işbirlikçi filtrelemede kullanılan yöntemlere örnek olarak Bayes ağları (Bayesian network), kümeleme (clustering) ve

kural tabanlı (rule based) yaklaşımlar gösterilebilir. Bir matrisin çarpanlarına ayrılması işlemi olan matris faktörizasyonu da model tabanlı bir yöntemdir. Gerçekleştirilen tez çalışmasında matris faktörizasyonu yöntemlerinden biri olan SGD yöntemi kullanılmıştır [34, 35].

Bellek tabanlı yöntemde ise kullanıcı ve nesneye ait oy değerleri kullanılarak bir tahmin oluşturulur. Bu yöntemde ilk olarak kullanıcının komşuları bulunur ve komşuları ile arasında olan benzerlik hesaplaması yapılır. Burada amaç komşu ile kişi arasında bir desen yakalayabilmektir. Benzerlik hesabında elde edilen değere göre kişiye bir öneride bulunulur [35, 36]. Kullanıcı veya nesnelere arasında benzerlik hesaplaması yapılırken yaygın olarak kullanılan yöntemler Pearson korelasyonu ve kosinüs benzerliği yöntemleridir [34].

Gerçekleştirilen tez çalışmasında Pearson korelasyonu ve kosinüs benzerliği yöntemleri kullanılmış ve bu iki yönteme ek olarak ayrıca sınırlı Pearson korelasyonu da kullanılmıştır. Sınırlı Pearson yönteminde, oy değerlerini pozitif ve negatif olarak ayrılır. Oy değerlerinin bu şekilde ayrılmasının oluşturulan sistem üzerindeki etkisini araştırmak için bu yöntem tercih edilmiştir.

Bellek tabanlı işbirlikçi filtreleme, kullanıcı oylarının benzerliği ile kullanıcı tabanlı; nesne oylarının benzerliği ile nesne tabanlı olarak oluşturulabilir [2].

Nesne tabanlı işbirlikçi filtreleme fikri 2000'li yılların başında Sarwar vd. tarafından ortaya atılmıştır [35]. Nesne tabanlı işbirlikçi filtreleme yöntemi içerik tabanlı filtreleme ile karıştırılabilir. Burada dikkat edilmesi gereken işbirlikçi filtrelemede kullanılan değerlerin oy değerleri olmasıdır.

Tablo 2.1'de yer alan tablo dikkate alındığında iki nesne arasındaki benzerliği hesaplamak için o nesnelere ait tüm kullanıcıların verdiği oy değerleri kullanılmalıdır. Bu durumda, n_2 ve n_1-1 nesnelere arasında benzerlik hesabı yapılmak istendiğinde kullanılması gereken oy değerleri Tablo 2.2'de gösterilmiştir.

Tablo 2.2. Nesne tabanlı işbirlikçi filtreleme

Nesneler Kullanıcılar	n ₁	n ₂	n _{i-1}	n _i
k ₁		x		x	
k ₂		x		x	
.		x		x	
.		x		x	
.		x		x	
k _{j-1}		x		x	
k _j		x		x	

Bellek tabanlı işbirlikçi filtrelemede kullanılan benzerlik hesaplama yöntemlerinden biri olan sınırlı Pearson korelasyonu, Shardanand vd. tarafından ortaya atılan bir hesaplama yöntemidir [37]. Kullanıcı tabanlı bir işbirlikçi filtrelemede, sınırlı Pearson yöntemi kullanılarak 7 puanlı bir ölçekte 1,2 ve 3 negatif, 4 tarafsız (nötr), 5,6 ve 7 pozitif oy değeri olmak üzere benzerlik hesaplanması Denklem (2.1)'de gösterilmiştir.

Denklem (2.1)'de I_u ve I_v, u ile v kullanıcılarının oy vermiş olduğu nesnelere, r_{u,i} ve r_{v,i} ise u ve v kullanıcılarının i nesnesine vermiş olduğu oy değerleri olmak üzere benzerlik hesabı;

$$\text{sim}(u,v)_{\text{sınırlı pearson}} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - 4)(r_{v,i} - 4)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - 4)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - 4)^2}} \quad (2.1)$$

şeklinde hesaplanır.

Pearson korelasyonu ve kosinüs benzerliği vb. hesaplama yöntemlerinden elde edilen benzerlik oranı sim(u, u'), u kullanıcısının ortalama oy değeri \bar{r}_u , u' kullanıcısının ortalama oy değeri $\bar{r}_{u'}$, u' kullanıcısının i nesnesine vermiş olduğu oy değeri r_{u',i} olmak üzere tahmini oy değerinin hesaplanması [38];

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} \text{sim}(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |\text{sim}(u,u')|} \quad (2.2)$$

Denklem (2.2)'de gösterilen şekilde yapılır.

İşbirlikçi filtreleme ile oluşturulan sistemlerde doğruluk oranını arttırmak amacı ile günümüzde bu alanda çeşitli çalışmalar yapılmaya devam edilmektedir. Bu çalışmalara örnek olarak, veri madenciliği teknikleri kullanılarak gerçekleştirilen işbirlikçi filtreleme ile oluşturulmuş öneri sistemleri gösterilebilir [39].

2.2.1. İşbirlikçi filtrelemede karşılaşılan problemler

Öneri sistemlerinde karşılaşılan bazı problemler vardır. Bu problemlerden biri seyrekliktir (sparsity). Az sayıda kullanıcının olması değerlendirme açısından yeterli olmayabilir. Oluşturulan sistemde çok sayıda nesne varsa ve kullanıcılar bu ürünlerden az sayıda ürün değerlendirdiyse ürüne ait az sayıda değerlendirme de sistemi olumsuz etkiler. Sistemde yeterli miktarda veri bulunmadığı durumlarda seyreklik problemi meydana gelir ve bunun sonucunda öneri sistemleri doğru sonuçlar veremeyebilir [40].

Soğuk başlangıç (cold start), öneri sistemlerinde karşılaşılan bir diğer büyük problemdir. Sisteme yeni bir ürün eklendiğinde ona ait herhangi bir oy verisi mevcut olmaz. Aynı şekilde sisteme yeni üye olan bir kişinin de diğer ürünlere ait herhangi bir oy değerlendirmesi mevcut değildir. Verinin mevcut olmaması, başlangıçta benzer kullanıcı veya benzer ürün bulmada probleme sebep olur. Bu problem soğuk başlangıç olarak adlandırılır. Soğuk başlangıç probleminin önüne geçmek için yapılan çeşitli çalışmalar mevcuttur. Soğuk başlangıç probleminin önüne geçmek için, genellikle sosyal medya vb. sitelerden kişi veya ürünlere ait veri toplama işlemi önerilmiştir [41 – 43].

İşbirlikçi filtreleme ile oluşturulan bir öneri sisteminde çok sayıda kullanıcı ve nesne bilgisi bulunabilir. Çok sayıda kullanıcının bulunması işbirlikçi filtrelemede temel bir işlem olan benzer kullanıcıların bulunmasını zorlaştırabilir. Çok fazla işlem gerekmesi sonucu sistemde yavaşlık meydana gelebilir. Bu sorunun önüne geçmek için SVD temelli bir boyut indirgeme tekniği, kümeleme temelli bir teknik ve kural çıkarımı vb. yöntemler önerilmiştir [44 – 46].

Öneri sisteminde bir nesne için aynı anlama gelebilecek farklı sözcükler kullanılabilir. Bu durumda her bir sözcük farklı bir nesne gibi düşünülebilir ve sistemin doğruluk oranı da bu durumdan olumsuz etkilenebilir. “Film” ve “movie”

aynı anlama gelen iki farklı sözcüktür. Eğer sistemde bu şekilde aynı anlama gelen farklı sözcükler bulunuyorsa bu durumun sisteme olan olumsuz etkisi en aza indirgenmeye çalışılmalıdır. Eşanlamlılık sorununun önüne geçebilmek için ontoloji, SVD ve Latent Semantic Indexing yöntemlerinden faydalanılabileceği belirtilmiştir [47 – 49].

2.2.2. İşbirlikçi filtrelemede değerlendirme

İşbirlikçi filtrelemede doğruluk kontrolü yapabilmek için eğitim ve test verisinin ayrılması gereklidir. Sistemdeki verileri, test için kullanılmak üzere ayırmak için elde bulunan oy matrisinin bazı verilerini gizlemek ve test aşamasında bu verilere göre ölçüm yapmak yaygın olarak kullanılan bir yöntemdir. Gizlenecek test verileri rastgele seçilebilir [50, 51]. Oy verilerini eğitim ve test olarak ayırmak için belirli bir oran belirtilmemiştir. Bu oran sistemi tasarlayan kişi tarafından belirlenir.

İşbirlikçi filtrelemede kurulan sistemin doğruluğunun belirlenmesi için kabul görmüş tek bir yöntem mevcut değildir. Öneri sistemleri çevrimiçi ve çevrimdışı çalışacak şekilde tasarlanabilir. Çevrimdışı çalışmalarda yaygın olarak mean absolute error (MAE) ve root mean square (RMSE) kullanılır. Çevrimiçi çalışan öneri sistemlerinde kullanıcının geri bildirim veya tıklama oranını hesaplaması olan Click Through Rate (CTR) işlemleri uygulanabilir [4, 52].

MAE hesaplanması, n toplam eksik veri sayısı, $p_{u,i}$ u kullanıcısının i nesnesine vereceği tahmin edilen oy değeri, $r_{u,i}$ ise u kullanıcısının i nesnesine ait vermiş olduğu gerçek oy değeri olmak üzere Denklem (2.3)'te gösterilen şekilde gerçekleştirilir.

$$MAE_{\text{öneri sistemi}} = \frac{1}{n} \sum_{u,i=1}^n |p_{u,i} - r_{u,i}| \quad (2.3)$$

RMSE hesaplanması n toplam eksik veri sayısı, $p_{u,i}$ u kullanıcısının i nesnesine vereceği tahmin edilen oy değeri, $r_{u,i}$ ise u kullanıcısının i nesnesine ait vermiş olduğu gerçek oy değeri olmak üzere;

$$\text{RMSE}_{\text{öneri sistemi}} = \sqrt{\frac{1}{n} \sum_{u,i=1}^n (p_{u,i} - r_{u,i})^2} \quad (2.4)$$

Şeklinde gerçekleştirilir.

2.3. İçerik Tabanlı Filtreleme

İçerik tabanlı filtreleme, öneri sistemlerinde sıklıkla karşılaşılan bir diğer filtreleme tekniğidir. Bu filtreleme tekniği ile öneri sistemi, nesnelerin içerik bilgisine dayandırılarak oluşturulur. İçerik tabanlı filtreleme kullanılan bir öneri sisteminde eğer bir kişi aksiyon filmlerini yüksek oy ile değerlendirmişse bu kişiye yapılan öneriler bu doğrultuda olacaktır.

Bir film öneri sistemi, içerik tabanlı filtreleme tekniği ile oluşturulmak istendiğinde aktör, aktris, yönetmen, senarist, tür, film adı vb. bilgileri kullanılabilir. Bir kitap öneri sistemi bu teknikle oluşturulacaksa yazar, kitap adı, yayın evi, türü ve konusu vb. bilgileri kullanılabilir. İçerik tabanlı filtreleme de ayrıca kullanıcıların demografik verileri üzerinden oluşturulan benzerlik de kullanılabilir. Her bir nesnenin bir diğeri ile benzerliğini ölçmek için genellikle kosinüs benzerliği ve tf-idf hesaplama yöntemleri kullanılır [53 – 58].

Son yıllarda içerik tabanlı filtreleme tekniği ile haber metinleri önerileri, kitap tavsiye sistemi ile ilgili çalışmalar mevcuttur. Mooney vd. içerik tabanlı filtreleme ve metin sınıflama ile kitap öneri sistemi gerçekleştirmişlerdir [59]. Çalışmalarında kullanıcılar hakkında bilgi olmadığında da içerik tabanlı filtremenin oldukça iyi sonuçlar verdiğini belirtmişlerdir.

Taşçı, çalışmasında içerik tabanlı filtreleme tekniğinin kullanıldığı bir haber sitesi gerçekleştirmiştir [60]. Bu haber sitesinde, kişilerin haberleri okuma zamanlarına göre kelime ağırlıkları zayıflatılmış veya güçlendirilmiştir. Bu sayede kişilere yakın zamanda okuduğu türe uygun olarak haber önerisinde bulunma amaçlanmıştır.

Lu vd. haber konusu önerisi ile ilgili bir öneri sistemi tasarlamışlardır [61]. Çalışmalarında, yeni bir yaklaşımda bulunarak içerik tabanlı işbirlikçi filtreleme

gerçekleştirmişlerdir. Ayrıca bu yaklaşım ile içerik tabanlı filtrelemedeki hata oranından daha düşük bir hata elde etmişlerdir.

2.4. Hibrit Yaklaşım

Öneri sistemlerinde kullanılan filtreleme yöntemlerinin zaman zaman yetersiz kaldığı görülmüştür. Bu sebeple içerik tabanlı veya işbirlikçi filtreleme arasında seçim yapmak yerine bu yöntemlerin birleştirildiği bir yöntem geliştirilmiştir. Bu yönteme hibrit yaklaşım denir.

Hibrit yaklaşımda, bir filtreleme tekniğinin yetersiz kaldığı noktada diğer bir filtreleme tekniğinin avantajı kullanılabilir. İşbirlikçi filtrelemede sisteme yeni bir ürün eklendiğinde hiçbir oy verisinin olmaması bu nesnenin önerilmesi oranını oldukça düşürür. Fakat eklenen ürünün içerik olarak sistemde önceden yer alan ürünlerden içerik olarak hangisi ile benzerlik gösterdiği işe yarar bir bilgi olabilir [62]. Lealos vd. bu durumu göz önünde bulundurarak hibrit yaklaşım ile film öneri sistemi [63], Dwivedi vd. ise restoran öneri sistemi tasarlamışlardır [64].

Hibrit yaklaşım ile tasarlanan öneri sistemlerinde içerik tabanlı ve işbirlikçi filtrelemenin yanı sıra daha doğru sonuçlar elde etmek için çeşitli veri madenciliği tekniklerinin kullanıldığı gözlemlenmiştir.

Christakou vd. çalışmalarında yapay sinir ağlarını da kullanarak hibrit bir film öneri sistemi oluşturmuşlardır [65]. İşbirlikçi filtreleme tekniği kısmında kullanıcı oylarını kullanmış ve Pearson korelasyonu metodu ile kullanıcılar arasındaki benzerliği hesaplamışlardır. İçerik tabanlı filtreleme kısmında ise filmin türü, özeti, aktörü, yönetmeni ve senaristi bilgilerini kullanmışlardır.

Alsalama, çalışmasında birliktelik kurallarının da kullanıldığı hibrit bir öneri sistemi oluşturmuştur [66]. İlk olarak kullanıcı nesne oy matrisi üzerinde Apriori algoritmasını uygulamıştır. Bu işlemden sonra kullanıcı oylarından yola çıkarak nesnelere favori olan ve olmayan olarak ayırmıştır. Favori olan nesnelere, birliktelik kurallarını oluşturarak; favori olmayan nesnelere ise içerik tabanlı filtreleme kullanarak önerilerde bulunmuştur.

Kuzelewska, kümeleme yönteminde küme sayısına bağı olarak bu teknikteki metrikleri karşılaştırdığı hibrit bir öneri sistemi gerçekleştirmiştir [67]. Çalışmasında MovieLens oylarını kullanmıştır. Mojtaba vd. MovieLens veri setini ile bulanık kümeleme (fuzzy clustering) yöntemini kullanarak hibrit bir öneri sistemi gerçekleştirmiştir [68].

Soni vd. ise gerçekleştirdikleri hibrit öneri sisteminde içerik tabanlı ve işbirlikçi filtreleme ile elde ettikleri sonucu, metin madenciliği ile birleştirip öneride bulunan bir sistem oluşturmuşlardır [69].

Ali vd. kitap öneri sisteminde içindekiler bilgisini de kullanan birliktelik kuralına dayalı hibrit bir öneri sistemi tasarlamışlardır [70]. Campos vd. ise bayes ağlarının kullanıldığı hibrit öneri sistemi ile ilgili bir çalışma gerçekleştirmiştir [71].

3. VERİ ANALİZİ

3.1. Giriş

Bu bölümde, çalışmada kullanılan gezi oyları hakkında bilgi verilmiştir. Oyların nasıl elde edildiği, oy verilen yerlerin neler olduğu ve oy veren katılımcıların demografik özelliklerine ait bilgilerden bu bölümde bahsedilmiştir.

3.2. Çalışma Grubu

Çalışmada kullanılacak veriler online anket ile elde edilmiştir. Araştırma için İstanbul'da ziyaret edilebilecek 50 yer belirlenmiştir. Oluşturulan anket ile Türkiye genelindeki katılımcılardan veri toplanmıştır. Katılımcılardan belirlenen bu yerlerden daha önce ziyaret ettiklerine 1 – 10 arasında bir puan vermeleri istenmiştir. Anket üzerindeki oylama alanından bir kesit Şekil 3.1'de gösterilmiştir.

1- Topkapı Sarayı *

0 1 2 3 4 5 6 7 8 9 10

2- Dolmabahçe Sarayı *

0 1 2 3 4 5 6 7 8 9 10

3- Çırağan Sarayı *

0 1 2 3 4 5 6 7 8 9 10

4- Galata Kulesi *

0 1 2 3 4 5 6 7 8 9 10

Şekil 3.1. Ankette yer alan oylama arayüzünden bir kesit

Ayrıca cinsiyet, medeni hal, yaş grubu, gelir ve eğitim düzeyi, meslek, yaşadıkları şehir ve bölge bilgilerini de paylaşımları istenmiştir. Anket ile 213 katılımcıya ait 7800 oy verisi elde edilmiştir.

3.3. Kullanılan Verinin Analizi

Araştırma için katılımcılardan değerlendirilmesi istenen yerler Tablo 3.1’de gösterilmiştir.

Tablo 3.1. Katılımcıların oyladığı yerlerin listesi

Sıra No	Yerin Adı
1	Topkapı Sarayı
2	Dolmabahçe Sarayı
3	Çırağan Sarayı
4	Galata Kulesi
5	Kız Kulesi
6	Eyüp Sultan Cami
7	Ortaköy Cami
8	Mihrimah Sultan Cami
9	Süleymaniye Cami
10	Sultanahmet Cami
11	Ayasofya Müzesi
12	İstanbul Arkeoloji Müzesi
13	Rahmi Koç Müzesi
14	Masumiyet Müzesi
15	Pera Müzesi
16	Aşiyen Müzesi
17	Oyuncak Müzesi
18	Panorama 1453
19	Miniatürk
20	Yerebatan Sarnıcı
21	Yedikule Surları
22	Cağaloğlu Hamamı
23	Kapalı Çarşı
24	Mısır Çarşısı
25	Kuzguncuk Evleri
26	Pierre Loti Tepesi
27	Çamlıca Tepesi
28	Otağtepe
29	Atatürk Arboretumu
30	Emirgan Korusu
31	Fethi Paşa Korusu
32	Mihrabat Korusu
33	Cemile Sultan Korusu

Tablo 3.1 (Devam). Katılımcıların oyladığı yerlerin listesi

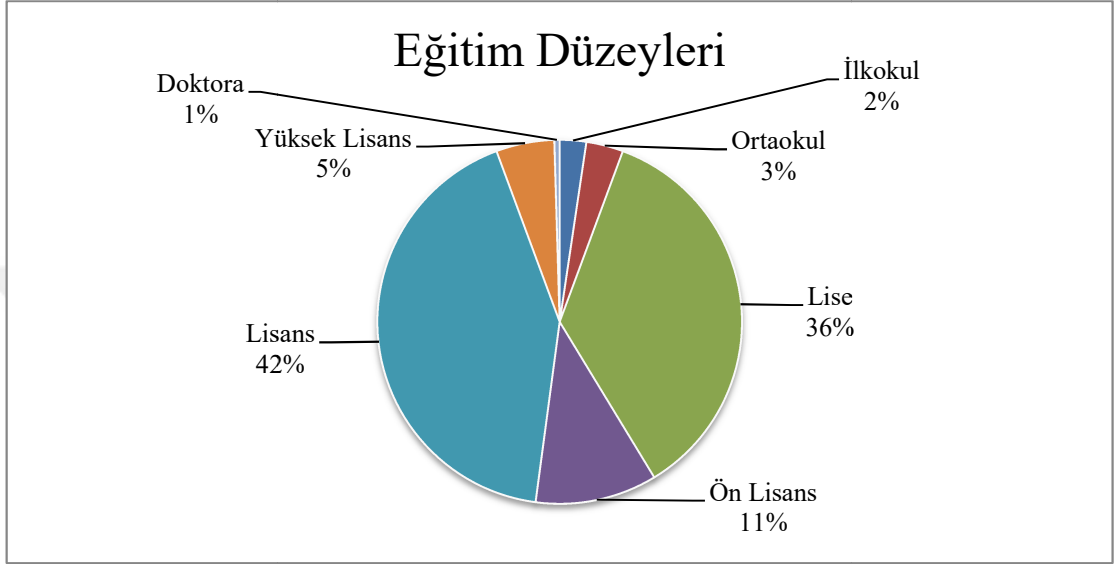
34	Gülhane Parkı
35	Gezi Parkı
36	Yoğurtçu Parkı
37	Yıldız Parkı
38	Rumeli Hisarı
39	Anadolu Hisarı
40	Haydarpaşa Garı
41	Taksim Meydanı
42	Büyükkada
43	Heybeli Ada
44	Kınalı Ada
45	Burgazada
46	Bebek Sahili
47	Boğaz Turu
48	Eminönü Balıkçıları
49	İstanbul Akvaryum
50	Vialand

Katılımcıların kişisel özelliklerine ait verileri elde etmek için Tablo 3.2’de yer alan ifadelere yanıt vermesi istenmiştir. Tablo 3.2’deki sorulara ek olarak katılımcılara meslekleri, yaşadıkları şehir ve bölge bilgisi de sorulmuştur. Elde edilen verilerde 29 farklı meslek bilgisi yer almaktadır. Ayrıca bazı katılımcılar mesleklerini “Diğer” olarak belirtmişlerdir. Türkiye genelindeki katılımcılardan elde edilen veriler doğrultusunda 19 farklı şehir ve 6 farklı bölgeden katılım gözlenmiştir. Ankete katılan kişilerden hiçbiri Güneydoğu Anadolu Bölgesi’nde ikamet etmemektedir.

Tablo 3.2. Kullanıcıdan istenen kişisel özellik soruları

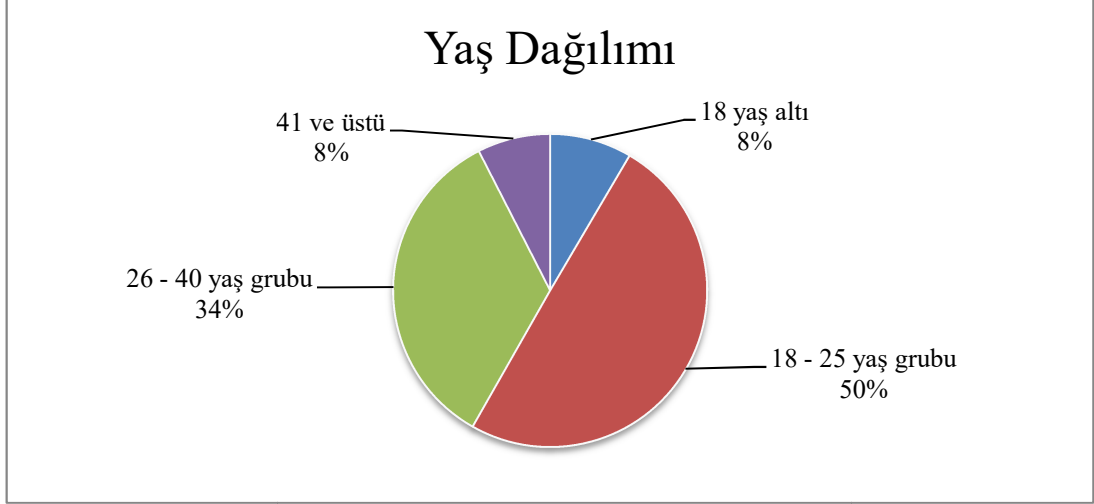
Nitelik	1	2	3	4	5	6	7
Cinsiyet	Erkek	Kadın					
Medeni Hal	Evli	Bekâr					
Eğitim Düzeyi	İlkokul	Ortaokul	Lise	Ön lisans	Lisans	Yüksek Lisans	Doktora
Gelir Düzeyi	Geliri yok.	1000TL altı	1000TL-2500TL	2500-5000TL	5000TL ve üstü		
Yaş	18 yaş altı	18-25 yaş grubu	26-40 yaş grubu	41 ve üstü			
Bölge	Marmara	Ege	Akdeniz	Karadeniz	İç Anadolu	Doğu Anadolu	Güneydoğu Anadolu

Katılımcıların özelliklerine göre ankete katılan kişiler incelendiğinde 66 erkek,147 kadın katılımcının ankete katıldığı; 48 kişinin evli, 165 kişinin bekâr olduğu; 5 ilkokul, 7 ortaokul, 76 lise, 23 ön lisans, 90 lisans, 11 yüksek lisans ve 1 doktora mezunu olduğu tespit edilmiştir. Eğitim düzeylerine göre elde edilen dağılım Şekil 3.2’de verilmiştir.



Şekil 3.2. Eğitim düzeylerine göre elde edilen dağılım

Gelir düzeylerine göre katılımcılar incelendiğinde ise 64 kişinin herhangi bir gelirin olmadığı, 18 kişinin gelirin 1000 TL altı olduğu, 45 kişinin 1000 – 2500 TL arası gelirin olduğu, 75 kişinin 2500 – 5000 TL arası gelirin olduğu ve 11 kişinin 5000 TL ve üzeri gelir elde ettiği gözlemlendi. Ayrıca 18 kişinin 18 yaş altı, 106 kişinin 18 – 25 yaş grubuna dâhil olduğu, 73 kişinin 26 – 40 yaş grubuna dâhil olduğu ve 16 kişinin 41 yaş ve üstü olduğu gözlemlendi. Yaş değerlerine göre elde edilen dağılım Şekil 3.3’te verilmiştir.



Şekil 3.3. Katılımcılara ait yaş dağılımı

Kullanıcılardan elde edilen oy verilerine göre en çok ziyaret edilen ilk 10 yer belirlenmiş ve bu veriler Tablo 3.3'te verilmiştir.

Tablo 3.3. En çok ziyaret edilen ilk 10 yer

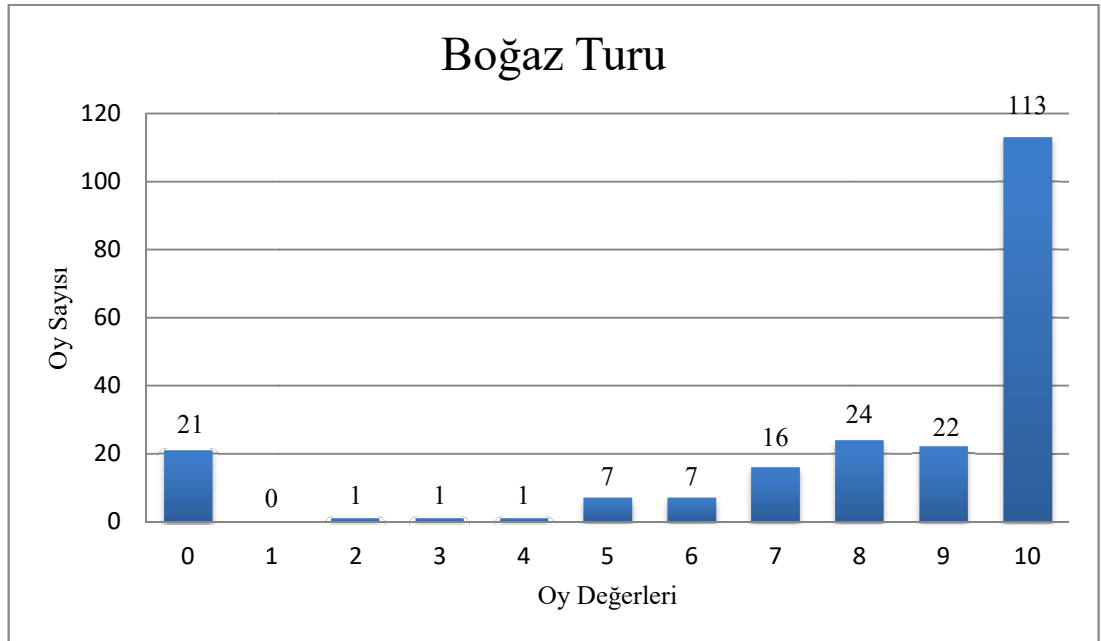
Sıra No	Yer Adı	Ziyaretçi Sayısı
1	Eminönü Balıkçıları	207
2	Kapalı Çarşı	203
3	Taksim Meydanı	203
4	Gülhane Parkı	196
5	Mısır Çarşısı	194
6	Boğaz Turu	192
7	Sultanahmet Cami	190
8	Haydarpaşa Garı	187
9	Çamlıca Tepesi	184
10	Topkapı Sarayı	183

Kullanıcılardan elde edilen oy verilerine göre ziyaret edilen yerlere ait ortalama oy değerleri hesaplanmıştır. Bu hesaplama yapılırken ziyaret edilmeyen yerlere ait 0 olan oy verisi işleme dâhil edilmeyip sadece ziyaret eden kullanıcılara ait oy verileri kullanılmıştır. Elde edilen sonuçlar Tablo 3.4'te gösterilmiştir.

Tablo 3.4. En yüksek ortalama oya sahip ilk 10 yer

Sıra No	Yer Adı	Ortalama Oy
1	Boğaz Turu	8,94791
2	Ayasofya Müzesi	8,85955
3	Dolmabahçe Sarayı	8,74566
4	Galata Kulesi	8,60335
5	Topkapı Sarayı	8,59016
6	Sultanahmet Cami	8,53684
7	Yerebatan Sarnıcı	8,45508
8	Çamlıca Tepesi	8,45108
9	Büyükkada	8,44692
10	Atatürk Arboretumu	8,4031

Katılımcılardan elde edilen oy doğrultusunda en yüksek ortalama oy değerine sahip olan boğaz turu, ziyaret eden kullanıcılardan % 59 oranında 10 puan almıştır. Boğaz turuna ait tüm kullanıcı oyları Şekil 3.4'te verilmiştir. Ziyaret etmeyen kullanıcılara ait oy değerleri grafikte 0 olarak yer almıştır.



Şekil 3.4. Boğaz Turu ögesine ait kullanıcı oyları

4. ÖNERİ SİSTEMLERİNDE İŞBİRLİKÇİ FİLTRELEME

4.1.Giriş

Bu bölümde, işbirlikçi filtreleme tekniğinde kullanılan yöntemlerden Pearson korelasyonu, kosinüs benzerliği, sınırlı Pearson korelasyonu, matris faktörizasyonu ve AML ile öneri sistemi oluşturmada bahsedilmiştir. Tez kapsamında bu hesaplama yöntemleri kullanılarak gerçekleştirilen sistem hakkında bilgi verilmiştir.

4.2. Pearson Korelasyonu ile Öneri Sistemi Oluşturma

Pearson korelasyonu öneri sistemlerinde kullanıcı benzerliklerini hesaplamak için kullanılan popüler hesaplama yöntemlerinden biridir. Pearson korelasyonu katsayısı ilk olarak Resnick vd. tarafından ortaya atılmıştır [72]. Benzerlik hesabı sonuçuna göre benzerlik minimum -1, maksimum 1 değerini alabilir. Benzerlik sonucu -1 ise kişiler birbirine benzer değildir; 1 ise kişiler birbirlerine tam uyum içerisindedir denebilir. Kişilerin herhangi bir nesneye verdikleri oy değeri r , ortalama oy değerleri \bar{r} , nesnelere I olmak kaydıyla u ve v kullanıcılarının benzerliğinin Pearson korelasyonu yöntemi ile hesaplanması Denklem (4.1)'de gösterilen şekilde gerçekleştirilir.

$$\text{sim}(u,v)_{\text{pearson}} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (4.1)$$

Pearson korelasyonu ile benzerlik hesaplaması yapılırken dikkat edilmesi gereken nokta bu hesaplama, benzerlik oranları hesaplanan kullanıcıların her ikisinin de oyladığı nesnelere üzerinden gerçekleştirilir.

Pearson korelasyonu birkaç ortak oy verisi olan kullanıcılar arasındaki benzerliği hesaplarken zorluk çekmektedir. Bu gibi durumlardan kaçınmak için yapılan çalışmalar sonucu bir eşik değerinin kullanılması önerilmiştir [36]. Bu fikir ilk olarak Herlocker vd. tarafından ortaya atılmıştır [73]. Bazı çalışmalarda bu eşik değerinin

50 olmasının uygun olduğu belirtilmiştir [2]. Belirlenen eşik değeri β olmak üzere hesaplanan benzerlik değeri Denklem (4.2)'de yer alan ifade ile çarpılmalıdır.

$$\frac{\min_{\beta} \{f_{\beta}(I_u \cap I_v, \beta)\}}{\beta} \quad (4.2)$$

Denklem (4.2)'de yer alan ifadenin sonuç aralığı $[0,1]$ 'dir. Rastgele 5 adet kullanıcıdan elde edilen ilk 5 nesneye ait oy değerleri Tablo 4.1'de gösterilmiştir. Bu değerler üzerinden B kullanıcıasına ait n_5 oy değerinin kullanıcı tabanlı işbirlikçi filtreleme nasıl bulunacağı gösterilmiştir.

Tablo 4.1. Rastgele seçilen kullanıcılara ait oy değerleri

Nesne Kullanıcı	n_1	n_2	n_3	n_4	n_5
A	7	10	5	6	4
B	8	9	-	6	?
C	8	10	-	10	-
D	10	10	-	6	-
E	7	9	-	8	8

B kullanıcıasının 5.sırada yer alan kayıp oy verisini bulmak için bu nesneye ait oy değeri bulunan kullanıcılar tespit edilir. Bu nesneye daha önce oy veren kullanıcılar A ve E kişileridir. Bu kullanıcılara ait ortalama oy değeri hesaplanmalıdır. Bazı çalışmalarda sadece ortak oylar kullanılarak ortalama oy hesabı yapılırken bazı çalışmalarda da ortak olmayan oy verileri de işleme katılarak ortalama oy hesabı yapılmıştır. Bu şekilde farklı hesaplamaların, hesaplamada çok önemli bir fark oluşturmadığı belirtilmiştir [74]. Kullanıcının ortak olmayan oy verileri de kullanılarak ortalama oy değeri hesaplandığında, B kullanıcıasının ortalama oy değeri 7,67; komşuları olan A ve E kişilerinin ortalama oy değerleri ise sırası ile 6,4 ve 8 olarak hesaplanmıştır. Komşu kullanıcıların ayrıca B nesnesiyle olan benzerliği Denklem (4.1) ile hesaplanmış ve $\text{sim}(B,A) = 0,7132$ ve $\text{sim}(B,E) = 0,3283$ olarak hesaplanmıştır.

$$p_{B,5} = \bar{r}_B + \frac{\text{sim}(B,A)(r_{A,5} - \bar{r}_A) + \text{sim}(B,E)(r_{E,5} - \bar{r}_E)}{|\text{sim}(B,A)| + |\text{sim}(B,E)|}$$

$$= 7,67 + \frac{0,7132(4 - 6,4) + 0,3283(8 - 8)}{|0,7132| + |0,3283|}$$

$$= 7,67 + \frac{-1,71168}{1,0415} = 6,03 \quad (4.3)$$

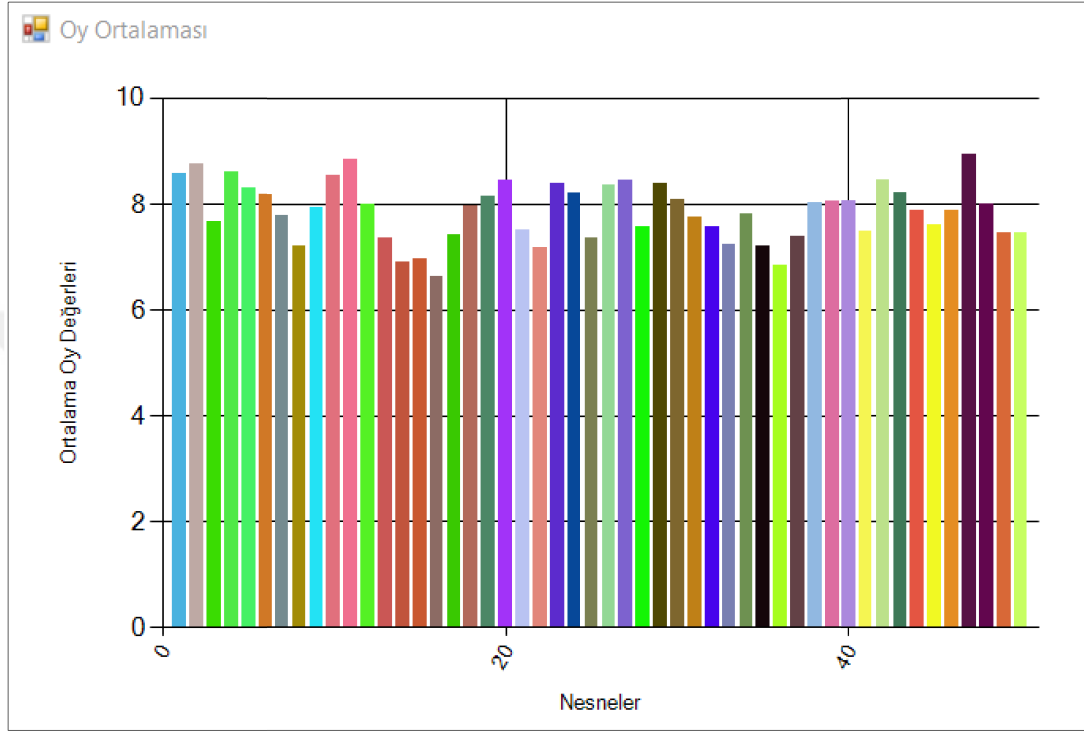
Buradan elde edilen sonuç doğrultusunda, Denklem (4.3)'te tahmini oy değeri hesaplanmıştır. B kullanıcısının 5. nesne olan Kız Kulesi için tahmini oy değeri 6 olarak hesaplanır.

Gerçekleştirilen öneri sisteminde bulunan oy verileri %80 eğitim, %20 test verisi olmak üzere ikiye ayrılmıştır. Test için kullanılacak oy verilerinin değerleri sistemde yer alan oy değerlerinden ayırmak için “-1” olarak değiştirilmiştir. Pearson benzerlikleri hesaplandıktan sonra bu değerler Denklem (4.2)'de yer alan ifade ile çarpılmıştır. Bu ifade ile çarpılmadan önce “doğru tahmin edilen veri sayısı/toplam test veri sayısı” oranı %24,7 iken bu ifade ile çarpım işlemi gerçekleştirildikten sonra %26,3 oranına ulaşmıştır.

Gerçekleştirilen uygulamaya ilk olarak csv uzantılı tüm oy verileri yüklenip hesaplama türü Pearson korelasyonu olarak seçilmiştir. Hesaplama türü belirlendikten sonra test için oluşturulan csv uzantılı oy verileri dosyası sisteme yüklenmiştir. Daha sonra “Uygula” butonuna basılmış ve Şekil 4.1’de yer alan ekran görüntüsü elde edilmiştir.

Şekil 4.1. Pearson korelasyonu ile elde edilen ekran görüntüsü

Uygulama ekranında yer alan “Oy Dağılımını Görüntüle” butonuna basıldığında sisteme yüklenen tüm oy verilerine ait ziyaret eden kullanıcıların vermiş olduğu oylar ile hesaplanan ortalama oy değerleri görüntülenir. Gezilebilecek 50 yere ait ortalama oy değerleri Şekil 4.2’de verilmiştir.



Şekil 4.2. Gezilebilecek 50 yere ait oy ortalaması

4.3. Kosinüs Benzerliği ile Öneri Sistemi Oluşturma

Kosinüs benzerliği, öneri sistemlerinde kullanıcı benzerliklerini hesaplamak için kullanılan bir diğer popüler hesaplama yöntemlerindedir. Pearson korelasyonu hesaplamasında olduğu gibi sonuç aralığı $[-1, +1]$ 'dir. Kişilerin herhangi bir nesneye verdikleri oy değerlerinin vektörü r olmak üzere u ve v kullanıcılarının benzerliğinin kosinüs benzerliği yöntemi ile hesaplanması Denklem (4.4)'te yer alan şekilde gerçekleştirilir.

$$\text{sim}(u,v)_{\text{kosinüs}} = \frac{r_u \cdot r_v}{\|r_u\| \|r_v\|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2} \sqrt{\sum_i r_{v,i}^2}} \quad (4.4)$$

Kosinüs benzerliğinde, kişilerin değerlendirmedikleri nesneye ait oy verisi 0 ise her iki kullanıcının da oyladığı nesnelere bulma işlemine gerek yoktur. Çarpma işleminde

ortak oylanmayan değerler sıfır ile çarpılacağı için hesaplamaya etki etmeyecektir. Denklem (4.4)'te bulunan benzerlik değeri, Denklem (2.2)'de yer alan benzerlik ifadesinin yerine yazılarak tahmini oy değeri hesaplaması yapılır.

Tablo 4.1'de yer alan oy verileri kullanılarak B kişisine ait 5. nesnenin tahmini oy değerini hesaplamak için ilk olarak 5. nesneye oy veren diğer kullanıcılar belirlenir. Komşu kullanıcılar belirlendikten sonra B kişisi ile aralarındaki benzerlik oranı Denklem (4.4)'te gösterildiği gibi hesaplanır. A kullanıcısı ile B kullanıcısının benzerlik hesabının yapılması Denklem (4.5)'te yer alan şekilde gerçekleştirilir.

$$\begin{aligned}
 \text{sim}(B,A) &= \frac{7.8+10.9+5.0+6.6}{\sqrt{(7)^2+(10)^2+(5)^2+(6)^2} \sqrt{(8)^2+(9)^2+(6)^2}} \\
 &= \frac{56+90+36}{\sqrt{49+100+25+36} \sqrt{64+81+36}} = \frac{182}{\sqrt{210} \sqrt{181}} \\
 &= \frac{182}{(14,5)(13,45)} = \frac{182}{195,025} = 0,93 \tag{4.5}
 \end{aligned}$$

Denklem (4.4)'te yer alan ifadeye göre E kullanıcısı ile B kullanıcısının benzerlik hesabının yapılması Denklem (4.6)'da yer alan şekilde gerçekleştirilir.

$$\begin{aligned}
 \text{sim}(B,E) &= \frac{7.8+9.9+8.6}{\sqrt{(7)^2+(9)^2+(8)^2} \sqrt{(8)^2+(9)^2+(6)^2}} \\
 &= \frac{56+81+48}{\sqrt{49+81+64} \sqrt{64+81+36}} = \frac{185}{\sqrt{194} \sqrt{181}} \\
 &= \frac{185}{(13,93)(13,45)} = \frac{185}{187,3585} = 0,98 \tag{4.6}
 \end{aligned}$$

A ve E kullanıcılarının B kullanıcısı ile benzerlik oranı hesaplandıktan sonra tahmini oy değeri hesaplanabilir. $\text{Sim}(B,A)$ ve $\text{sim}(B,E)$ benzerlik oranları; \bar{r}_A , \bar{r}_B ve \bar{r}_E ise A, B ve E kullanıcılarının ortalama oy değeri; $r_{A,5}$ ve $r_{E,5}$ ise A ve E kişilerinin 5.nesneye vermiş oldukları oy değeri olmak üzere B kullanıcısının 5. Nesneye verebileceği tahmini oy değerinin hesaplanması Denklem (4.7)'de yer alan şekilde gerçekleştirilir.

$$\begin{aligned}
P_{B,5} &= \bar{r}_B + \frac{\text{sim}(B,A)(r_{A,5} - \bar{r}_A) + \text{sim}(B,E)(r_{E,5} - \bar{r}_E)}{|\text{sim}(B,A)| + |\text{sim}(B,E)|} \\
&= 7,67 + \frac{(0,93)(4 - 6,4) + (0,98)(8 - 8)}{|0,93| + |0,98|} \\
&= 7,67 + \frac{-2,232}{1,91} = 6,5
\end{aligned} \tag{4.7}$$

Yapılan çalışmada Pearson korelasyonunda kullanılan eğitim ve test verisi burada da kullanılmıştır. Kosinüs benzerliğinde “doğru tahmin edilen veri sayısı/toplam test veri sayısı” oranı %27,6 olarak ölçülmüştür.

Gerçekleştirilen uygulamaya ilk olarak csv uzantılı tüm oy verileri yüklenip hesaplama türü kosinüs benzerliği olarak seçilmiştir. Hesaplama türü belirlendikten sonra test için oluşturulan csv uzantılı oy verileri dosyası sisteme yüklenmiştir. Daha sonra “Uygula” butonuna basılmış ve Şekil 4.3’te yer alan ekran görüntüsü elde edilmiştir.

Şekil 4.3. Kosinüs benzerliği ile elde edilen ekran görüntüsü

4.4. Matris Faktörizasyonu ile Öneri Sistemi Oluşturma

Matris faktörizasyonu, bir matrisi çarpanlarına ayırma işlemidir. İşbirlikçi filtrelemede tercih edilen hesaplama yöntemlerinden biridir [75 – 80].

Matris faktörizasyonu yapmak için birden fazla yöntem bulunur. Başlıca kullanılan yöntemler SVD, NMF, ALS ve Stochastic Gradient Descent (SGD) olarak sayılabilir. Bu yöntemlerden en bilineni Netflix yarışmasında da kullanılan SVD yöntemidir.

Yapılan çalışmada SGD kullanılmıştır. Bu çalışmada en popüler matris faktörizasyonu yöntemi olan SVD yönteminin tercih edilmemesinin birkaç sebebi bulunmaktadır. SVD yönteminde oy matrisi üç farklı matrisin çarpımı ile oluşturulur. Bu matrislerden biri ortogonal matris, biri köşegen matris bir diğeri ise bir ortogonal matrisin transpozunun alınması ile oluşan matristir [81]. SGD yönteminde ise oy matrisi iki farklı matrisin çarpımı ile oluşturulur ve bu iki matrisin ortogonal veya köşegen olması gibi bir zorunluluk yoktur. Ayrıca SVD yönteminde elde edilen üç matrisin çarpımı oy matrisini verirken, SGD yöntemindeki iki matris çarpıldığında oy değeri bulunmayan (sıfır değerine eşit olan) nesnelere için de bir değer oluşturulur. Bu nesnelere de sistemdeki yüksek oy değerlerine göre öneri olarak sunulabilir. SGD yöntemi, daha az işlem yükü gerektirdiği ve oy değeri bulunmayan nesnelere için oluşturduğu tahmini oy değeri ile bir avantaj sağladığı için tercih edilmiştir.

SGD yönteminde, R oy değerlerinin bulunduğu matris olmak üzere; R matrisinin iki matrisin çarpımı olarak ifade edilmesi Denklem (4.8)'de gösterilmiştir.

$$R \approx PQ^T = \hat{R} \quad (4.8)$$

P ve Q matrisleri başlangıçta rastgele oluşturulur. Burada dikkat edilmesi gereken nokta, bu matrislerdeki hiçbir değer sıfıra eşit olmaması gerektiğidir. R matrisi m kişi ve n sayıda nesneden oluşmak üzere $m \times n$ boyutlu ise P $m \times K$, Q $n \times K$ olarak ayrılır. K değeri, gizli özellikler (latent features) olarak adlandırılır. K değeri m ve n değerlerinden küçük bir değer olmalıdır [82]. Bir film öneri sistemi için gizli

özellikler film türleri, cinsiyet vb. olabilir. P matrisi kişiler ve özellikler, Q matrisi ise nesnelere ve özellikler arasındaki bağı temsil eder.

Kişilerin verebileceği tahmini oy değeri \hat{r}_{ij} , kişilerin nesneye vermiş olduğu gerçek oy değeri r_{ij} , hata değeri e_{ij} , gizli özellik sayısı K olmak üzere tahmini oy değeri hesaplaması, bulunan tahmini oy değeri ile gerçek oy değerinin farkı ile hata hesaplaması ve sisteme ait karesel hatanın toplamı (SSE) hesabı [80];

$$\hat{r}_{ij} = \sum_{k=1}^K p_{ik} q_{kj} \quad (4.9a)$$

$$e_{ij} = r_{ij} - \hat{r}_{ij} = r_{ij} - \sum_{k=1}^K p_{ik} q_{kj} \quad (4.9b)$$

$$SSE = \sum_{(i,j) \in \tau} e_{ij}^2 \quad (4.9c)$$

Şeklinde gerçekleştirilir.

Hata negatif veya pozitif olabileceği için hesaplamalarda genellikle karesi alınarak kullanılır. Hata değerini küçültmek için hangi yönde azaltılma yapılmasının bilinmesi gerekir [83]. Başka bir deyişle o noktadaki eğim bilinmelidir. Bu noktada devreye türev işlemi girer. Kişilerin verebileceği tahmini oy değeri \hat{r}_{ij} , kişilerin nesneye vermiş olduğu gerçek oy değeri r_{ij} , hata değeri e_{ij} olmak üzere türev işleminin yapılması Denklem (4.10) ve Denklem (4.11)'de gösterilmiştir.

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij}q_{kj} \quad (4.10)$$

$$\frac{\partial}{\partial q_{kj}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik} \quad (4.11)$$

Türev işleminden elde edilen değerlere göre p_{ik} ve q_{kj} değerleri her adımda güncellenir. Yeni değerlerin elde edilmesi Denklem (4.12) ve Denklem (4.13)'te gösterilmiştir.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj} \quad (4.12)$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik} \quad (4.13)$$

Denklem (4.12) ve (4.13)'te yer alan α ifadesi sabit ifadedir ve çok küçük bir değer seçilmelidir. Büyük bir değer seçilmesi, uygun olan değer atlanmasına sebep olabilir. Burada ayrıca aşırı uyuma (overfitting) sorunu yaşanabilir. Aşırı uyum, sistemdeki verilerin ezberlenip bu doğrultuda tek düze sonuçlar üretmesidir. Bu sorundan kaçınmak bir düzenleme yapılmalıdır. Bu sorundan kaçınmak için farklı eğitim ve test verileriyle sistemde deneme yapılabilir. Eğer bu sorun karar ağaçlarında yaşanıyorsa, budama denilen işlem gerçekleştirilebilir. SGD ile gerçekleştirilen matris faktörizasyonunda bu durumun önüne geçmek için sisteme yeni bir parametre eklenmelidir [83]. Yeni parametre değerinin eklenmesi hata değerinin hesaplanması ve güncellenin yapılması;

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^K p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^K (||P||^2 + ||Q||^2) \quad (4.14)$$

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij} q_{kj} - \beta p_{ik}) \quad (4.15)$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij} p_{ik} - \beta q_{kj}) \quad (4.16)$$

Şeklinde gerçekleştirilir.

Tablo 4.1.'de yer alan oy matrisinin yukarıda yer alan işlemler ile 5000 iterasyonda, $K=2$, $\alpha=0,0002$ ve $\beta=0.2$ değerleri ile elde edilen sonuç Tablo 4.2'de gösterilmiştir. Tablo 4.1 ve Tablo 4.2'de yer alan oy verileri karşılaştırıldığında birbirlerine oldukça yakın değerler elde edildiği gözlenmiştir. Bu yöntemle ayrıca oy değeri bulunmayan nesnelere için de oy değerlerinin hesaplandığı görülebilir. Elde edilen bu değerler, kişinin o nesneye verebileceği tahmini oy değeridir. Öneride bulunurken kişinin oy

değerinin bulunmadığı nesnelere arasında en yüksek oya sahip olanlar sıralanmış ve bunlardan belirli sayıda seçilip öneri olarak sunulmuştur.

Tablo 4.2. Gradient Descent ile elde edilen oy değerleri

Nesne Kullanıcı	n ₁	n ₂	n ₃	n ₄	n ₅
A	7,9169	9,1121	4,9937	5,969	4,182
B	7,8682	9,0806	4,9822	6,0145	4,2711
C	7,8637	10,2107	5,8728	9,8209	9,5923
D	9,428	10,504	5,6735	5,9428	3,3518
E	6,9763	8,9146	5,0969	8,23	7,8335

Tablo 4.2’de yer alan tahmini oy değerlerinin elde edildiği P ve Q^T matrisleri, kişi ve nesnelere, gizli faktör değerine göre faktörizasyon işlemi sonrası elde edilen gizli özellik katsayılarının bulunduğu matrislerdir. İşlemler sonrası elde edilen P matrisi Tablo 4.3’te verilmiştir.

Tablo 4.3. P matrisini oluşturan değerler

1,009	2,343
1,047	2,314
3,124	1,607
0,566	3,006
2,508	1,515

Tablo 4.1’deki verilere faktörizasyon işlemleri uygulandıktan sonra, nesnelere ait gizli özelliklerin bulunduğu Q^T matrisi ise Tablo 4.4’te verilmiştir.

Tablo 4.4. Q^T matrisini oluşturan değerler

1	1,628	1,006	2,354	2,764
2,946	3,186	1,697	1,532	0,593

Tablo 4.1’deki veriler, gizli özellik değerine bağlı olarak ayrıldığında burada yer alan özelliklerin ne olduğu hakkında bir bilgi edinilememektedir. Burada yer alan nesnelere üzerinden örnek verildiğinde K=2 değeri 1900 yılı öncesi veya sonrası yapılan eserler, denize yakın olan veya uzak olan yerler olabilir. Matrislerde yer alan değerler de bu gizli özellikler ile kişi arasındaki katsayı değerleridir.

Bir film öneri sistemi düşünüldüğünde, P ve Q^T matrislerindeki ifadeler aksiyon, dram, komedi veya belgesel gibi türleri temsil ediyor olabilir ve bu değişkenleri beğenme durumuna göre bir katsayı değeri içeriyor denebilir.

P ve Q^T matrislerindeki katsayılar ayrıca X adlı oyuncunun başrol oynadığı filmler veya Y adlı senaristin yazdığı eserler gibi özellikleri de temsil ediyor olabilir [77, 84, 85].

Gezilecek yer önerisi için oy matrisi $K=15$, $\alpha=0,0002$ ve $\beta=0,2$ değerleri kullanılmak üzere başlangıçta iki matrise bölünmüş ve hesaplama işlemi bu matrisler üzerinden ayrı ayrı gerçekleştirildikten sonra bu matrisler ile elde edilen tahmin değerleri tekrar birleştirilmiştir. Oy matrisi bölünmeden “doğru tahmin edilen oy sayısı/tüm oy sayısı” oranı %55 - %57 iken bu oran oylar bölündükten sonra aynı ile %76 - %79 arasında değişim göstermiştir. Burada ikiye bölünen matrislerde K değerinin sistemin doğruluğuna en büyük etkiyi sağladığı görülebilir. Tablo 4.1’de yer alan oy matrisinin girişte ikiye bölünüp ayrı birer matris olarak değerlendirildiği işlemler ile 5000 iterasyonda, $K=2$, $\alpha=0,0002$ ve $\beta=0,2$ değerleri ile elde edilen sonuç Tablo 4.5’te gösterilmiştir.

Tablo 4.5. Matris bölündüğünde elde edilen oy değerleri

Nesne Kullanıcı	n_1	n_2	n_3	n_4	n_5
A	7,021	9,9627	4,9933	5,9746	4,0115
B	7,9518	9,0211	4,2421	5,9904	5,107
C	8,0094	9,9713	7,1648	9,9842	8,3779
D	9,9741	9,9987	4,251	5,9904	5,0944
E	6,9947	8,9861	4,8433	7,996	7,9794

Yapılan çalışmada elde edilen oy verileri sisteme yüklenmiş ve hesaplama işlemi gerçekleştirilmiştir. Tablo 4.2’de yer alan değerler matris bir bütün olarak ele alındığında, Tablo 4.5’te yer alan değerler ise matris girişte iki farklı matris olarak ele alındığında elde edilen sonuçlardır. Burada bütün matriste ve bölünmüş matrislerde aynı α ve β değerleri kullanılarak işlem yapılmıştır. Bu sonuçlar incelendiğinde matris bölündüğünde daha iyi sonuçlar elde edildiği ve bu şekilde iyi sonuçlar elde edilmesinin en büyük sebebinin ise K değeri olduğu söylenebilir.

Matris bölündüğünde, tek matris ile belirli bir sayıda iterasyon ile gerçekleştirmek yerine, iki ayrı matris ile iterasyonların gerçekleştirilmesi her ne kadar olumlu sonuçlar verse de sistemin iş yükünü arttırıp, sistemi yavaşlatabilir.

Matrisi bölme işlemi gerçekleştirilmeden K değerini iki katına çıkarmak da aynı şekilde sistemdeki iş yükü üzerinde negatif bir etki oluşturup sistemi yavaşlatabilir.

Gerçekleştirilen uygulamaya ilk olarak csv uzantılı tüm oy verileri yüklenip hesaplama türü matris faktörizasyonu olarak seçilmiştir. Bu yöntemde test verisi yüklemeye gerek duyulmamaktadır. Oy verileri yüklenip yöntem olarak matris faktörizasyonu seçildikten sonra “Uygula” butonuna basılmış ve Şekil 4.4’te yer alan ekran görüntüsü elde edilmiştir. Burada elde edilen sonuçlar matris faktörizasyonu başlangıçta bölünmeden bir bütün olarak ele alındığında elde edilen sonuçlardır.

Öneri Sistemi

Dosya Adı : tüm oy verileri.csv Dosya Seç

Pearson Korelasyonu Kosinüs Benzerliği Matris Faktörizasyonu Sınırlı Pearson Korelasyonu

Dosya Adı : Test Verisi Yükle

Uygula

Oy Dağılımını Görüntüle

Sonuç :

Eksik veri: 7800 Doğru tahmin edilen veri: 4317 Doğru tahmin oranı: 0,5535

Adm:5000 RMSE : 1,02565 MSE : 1,05196 MAE : 0,64168

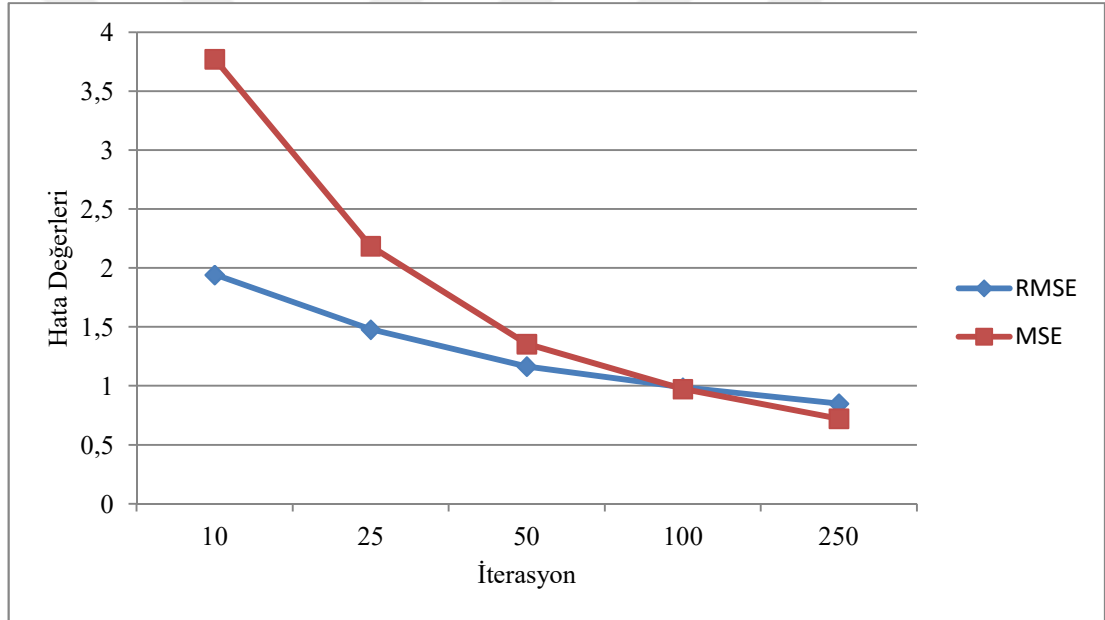
Şekil 4.4. Matris faktörizasyonu ile elde edilen ekran görüntüsü

Matris faktörizasyonu işlemi ayrıca 671 kullanıcı ve 9066 film ile elde edilen 100.005 oy verisi bulunan MovieLens [86] veri seti kullanılarak da gerçekleştirilmiştir. MovieLens veri setinde yer alan oy verilerinin örnek gösterimi Tablo 4.6’da gösterilmiştir. MovieLens oy verileri dosyasında “UserID, MovieID ve Rating” verileri yer almaktadır. Tablo 4.6’da bu sütun başlıklarına ek olarak “Sıra Numarası ve Film Adı” sütunları eklenmiştir. Burada yer alan “UserID, MovieID ve Rating” verileri Tablo 2.1’de yer alan gösterime uygun olarak yeniden düzenlenmiş ve sisteme yüklenmiştir.

Tablo 4.6. MovieLens veri setinde yer alan oy değerleri

Sıra Numarası	UserID	MovieID	Rating	Film Adı
1	1	31	2,5	Dangerous Minds
2	1	1029	3,0	Dumbo
3	1	1061	3,0	Sleepers
.
.
.
100005	671	6565	3,5	Seabiscuit

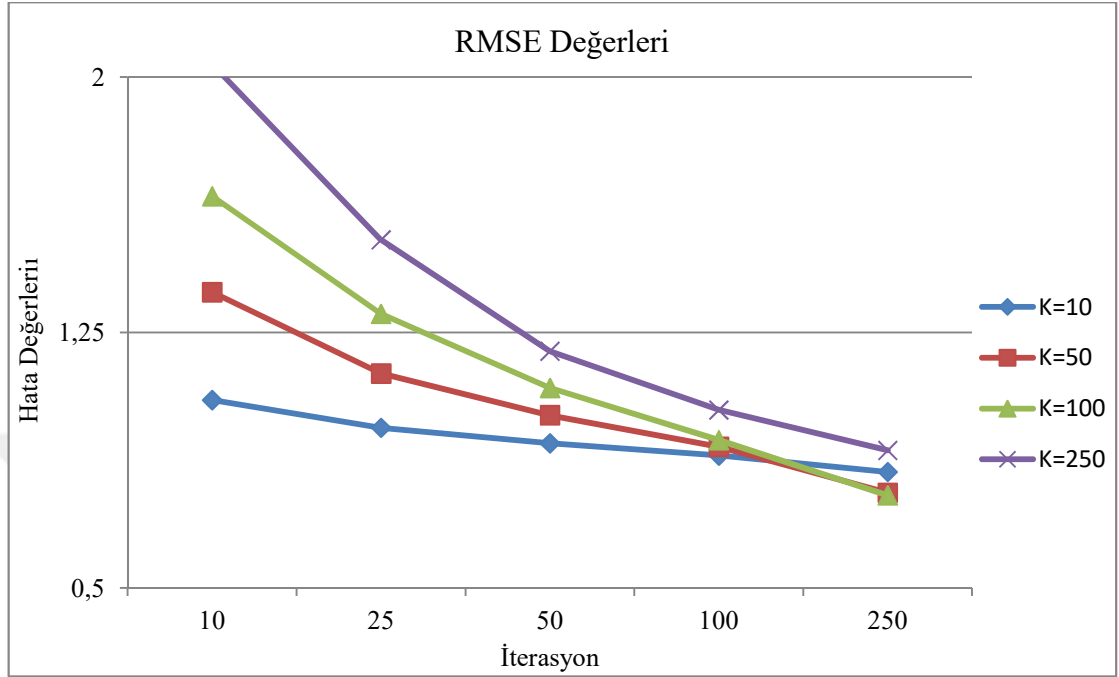
MovieLens veri seti kullanıldığında, $K=201$, $\alpha=0,0002$ ve $\beta=0,2$ değerleri ile elde edilen RMSE ve MSE değerleri Şekil 4.5'te gösterilmiştir. Bu veri seti ile yapılan çalışmalar incelendiğinde RMSE değerinin 0,815992 – 3,4 aralığında değiştiği gözlenmiştir [87 – 91].



Şekil 4.5. MovieLens verileri kullanıldığında elde edilen hata değerlerinin gösterimi

MovieLens verileri ile matris faktörizasyonu işlemleri gerçekleştirildiğinde seçilen K değerinin hata oranına nasıl bir etki yaptığı gözlenmek istenmiştir. Bu sebeple sistemde K değerleri sırası ile 10, 50, 100 ve 250 olarak seçildiğinde oluşan RMSE değerlerinin gösterimi Şekil 4.6'da gösterilmiştir. Şekil 4.6 incelendiğinde K değerinin başlangıçta büyük bir değer olması sistemdeki hata değerinin de başlangıçta yüksek olmasına yol açmıştır. Bu sebeple iterasyonlar arasındaki değer farkı, K değişkeninin büyük olduğu yerde daha fazla değişim göstermiştir.

Gerçekleştirilen gezi öneri uygulamasında K değeri, kişi veya nesne sayılarından küçük olanın % 30'u olarak seçilmiştir.



Şekil 4.6. Farklı K değerlerine bağlı olarak elde edilen hata değerleri

Kullanılan matris faktörizasyonu yöntemi ile bulunan RMSE değerleri, diğer çalışmalardan elde edilen RMSE değerleri ile karşılaştırıldığında hata aralığının diğer çalışmalarla elde edilen veriler ile aynı aralıkta olduğu gözlenmiştir. Bu durumda kullanılan yöntemin diğer yöntemler kadar başarılı olduğu söylenebilir.

4.5. Sınırlı Pearson Korelasyonu ile Öneri Sistemi Oluşturma

Sınırlı Pearson korelasyonu, genellikle ayrık oy değerleri ile yapılan öneri sistemlerinde kullanılan bir benzerlik hesaplama yöntemidir. Shardanand vd. tarafından ortaya atılan bir hesaplama yöntemidir [37]. 5 ölçekli bir değerlendirme sisteminde 1,2 oy değerleri negatif, 3 tarafsız 4,5 oy değerleri ise pozitif oy değeri olmak üzere benzerlik hesaplanması Denklem (4.17)'de gösterilmiştir. Denklem (4.17)'de I_u ve I_v , u ile v kullanıcılarının oy vermiş olduğu nesnelere, $r_{u,i}$ ve $r_{v,i}$ ise u ve v kullanıcılarının i nesnesine vermiş olduğu oy değerleri olmak üzere benzerlik hesabı;

$$\text{sim}(u,v)_{\text{sınırlı pearson}} = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i}-3)(r_{v,i}-3)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i}-3)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i}-3)^2}} \quad (4.17)$$

şeklinde gerçekleştirilir.

Bu yöntemde, Pearson korelasyonundan farklı olarak ortalama oy değeri yerine tarafsız olarak belirlenen oy değeri ile benzerlik katsayısı hesabı yapılır. Sınırlı Pearson korelasyonunda her iki kullanıcının da pozitif değerlendirmede bulunması, aralarındaki benzerlik oranlarını yükseltir.

Yapılan çalışmada tarafsız oy değeri 5 olarak kabul edildiğinde “doğru tahmin edilen veri sayısı/toplam test veri sayısı” oranı %28 olarak ölçülmüştür. Nötr oy değeri olarak 6 seçildiğinde ise bu oran %28,4 olarak hesaplanmıştır.

Tablo 4.1’de yer alan oy verileri kullanılarak B kişisine ait 5. nesnenin tahmini oy değerini hesaplamak için ilk olarak 5. nesneye oy veren diğer kullanıcılar belirlenir. Komşu kullanıcılar belirlendikten sonra B kişisi ile aralarındaki benzerlik oranı Denklem (4.17)’de gösterildiği gibi hesaplanır. Tarafsız oy değeri 5 olarak alındığında A kullanıcısı ile B kullanıcısının benzerlik hesabının yapılması Denklem (4.18)’de gösterilmiştir.

$$\begin{aligned} \text{sim}(B,A) &= \frac{(7-5)(8-5)+(10-5)(9-5)+(6-5)(6-5)}{\sqrt{(7-5)^2+(10-5)^2+(6-5)^2} \sqrt{(8-5)^2+(9-5)^2+(6-5)^2}} \\ &= \frac{(2)(3)+(5)(4)+(1)(1)}{\sqrt{(2)^2+(5)^2+(1)^2} \sqrt{(3)^2+(4)^2+(1)^2}} = \frac{6+20+1}{\sqrt{30} \sqrt{26}} \\ &= \frac{27}{(5,48)(5,1)} = \frac{27}{27,95} = 0,97 \end{aligned} \quad (4.18)$$

Tarafsız oy değeri 5 olarak alındığında E kullanıcısı ile B kullanıcısının benzerlik hesabının yapılması Denklem (4.19)’da gösterilmiştir.

$$\text{sim}(B,E) = \frac{(7-5)(8-5)+(9-5)(9-5)+(8-5)(6-5)}{\sqrt{(7-5)^2+(9-5)^2+(8-5)^2} \sqrt{(8-5)^2+(9-5)^2+(6-5)^2}}$$

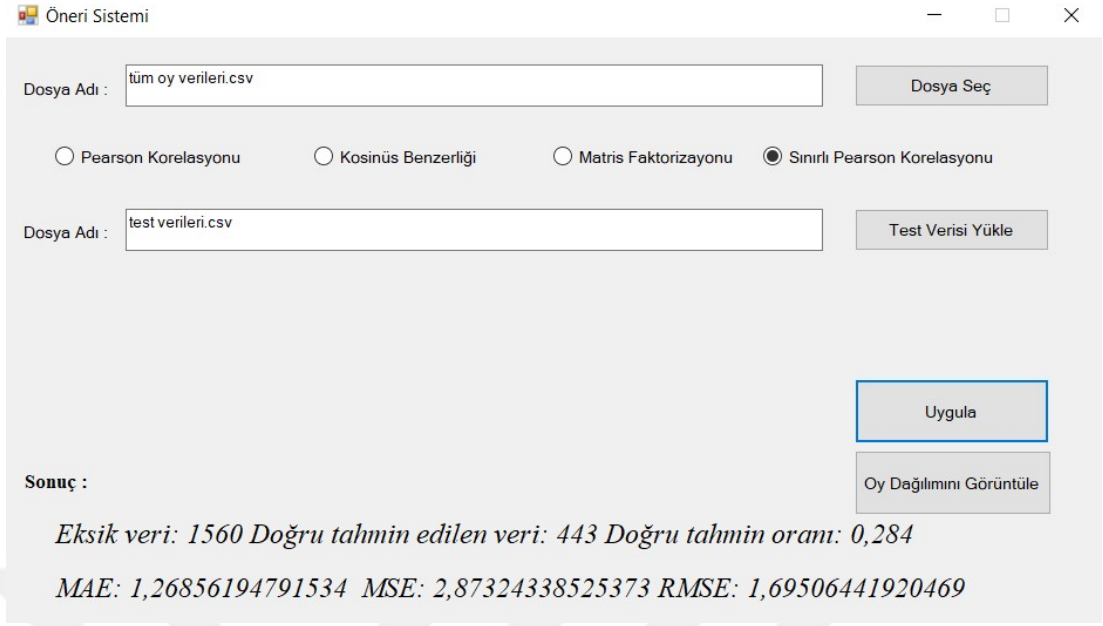
$$\begin{aligned}
&= \frac{(2)(3)+(4)(4)+(3)(1)}{\sqrt{(2)^2+(4)^2+(3)^2} \sqrt{(3)^2+(4)^2+(1)^2}} = \frac{6+16+3}{\sqrt{29} \sqrt{26}} \\
&= \frac{25}{(5,39)(5,1)} = \frac{25}{27,49} = 0,91 \tag{4.19}
\end{aligned}$$

A ve E kullanıcılarının B kullanıcısı ile benzerlik oranı hesaplandıktan sonra tahmini oy değeri hesaplanabilir. Sim(B,A) ve sim(B,E) benzerlik oranları; \bar{r}_A , \bar{r}_B ve \bar{r}_E ise A, B ve E kullanıcılarının ortalama oy değeri; $r_{A,5}$ ve $r_{E,5}$ ise A ve E kişilerinin 5. nesneye vermiş oldukları oy değeri olmak üzere B kullanıcısının 5. Nesneye verebileceği tahmini oy değerinin hesaplanması Denklem (4.20)'de gösterilmiştir.

$$\begin{aligned}
p_{B,5} &= \bar{r}_B + \frac{\text{sim}(B,A)(r_{A,5} - \bar{r}_A) + \text{sim}(B,E)(r_{E,5} - \bar{r}_E)}{|\text{sim}(B,A)| + |\text{sim}(B,E)|} \\
&= 7,67 + \frac{(0,97)(4 - 6,4) + (0,91)(8 - 8)}{|0,97| + |0,91|} \\
&= 7,67 + \frac{-2,328}{1,88} = 6,43 \tag{4.20}
\end{aligned}$$

Tarafsız oy değeri 6 olarak benzerlik oranları hesaplandığında sim(B,A)=0,94; sim(B,E) ise 0,82 olarak hesaplanmıştır. Bu değerler kullanılarak B kullanıcısının 5. Nesneye verebileceği tahmini oy değeri 6,39 olarak hesaplanmıştır.

Gerçekleştirilen uygulamaya ilk olarak csv uzantılı tüm oy verileri yüklenip hesaplama türü sınırlı Pearson korelasyonu olarak seçilmiştir. Hesaplama türü belirlendikten sonra test için oluşturulan csv uzantılı oy verileri dosyası sisteme yüklenmiştir. Daha sonra ‘‘Uygula’’ butonuna basılmış ve Şekil 4.7’de yer alan ekran görüntüsü elde edilmiştir.



Şekil 4.7. Sınırlı Pearson korelasyonu ile elde edilen ekran görüntüsü

4.6. Azure Machine Learning Platformu ile İşbirlikçi Filtreleme

Tez çalışmasında, geleneksel yöntemlerin yanı sıra bir veri bilimi platformunda da öneri sistemi gerçekleştirilmek istenmiş ve mevcut platformlardan Azure Machine Learning tercih edilmiştir.

Azure Machine Learning, Microsoft tarafından geliştirilen bir veri bilimi platformudur [92]. İçerisinde birçok örnek veri setlerini barındıran bu platform ile sınıflandırma, kümeleme gibi makine öğrenmesi işlemi yapılabilmektedir. Kullanıcılar isterlerse kendi verilerini de kullanabilmektedir. Bu platform üzerinde ayrıca öneri sistemi de oluşturulabilmektedir. Azure Machine Learning ile işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımda bulunulabilir. Her bir kullanıcıya 10 GB ücretsiz depolama alanı verilir. Platform, tarayıcı üzerinden çevrimiçi olarak çalışır. İçerisindeki modüller sürükle bırak yöntemi ile çalıştığı için kullanıcılar kod yazma bilgisi olmadan da kolay bir şekilde uygulama geliştirebilir. Şekil 4.8’de AML üzerinde yer alan öneri sisteminin çalışma prensibi verilmiştir.



Şekil 4.8. AML üzerinde öneri sistemi temel çalışma prensibi

Kullanıcılar, okul elektronik postası (edu uzantılı mail adresi) ile üye olduklarında platformu 1 yıl süresince ücretsiz kullanabilmektedirler. Kullanıcılar sisteme üye olduklarında ilk ay hesaplarında tanımlanan kredi ile ücretli servisleri de satın alıp 1 ay boyunca deneyimleyebilirler [93]. Platform üzerinde web servisleri de oluşturulabilir; ancak ücretsiz sürümde oluşturma sayısı ve aylık sorgu sayısı kısıtlıdır. Azure Machine Learning platformu kullanıcı oylarıyla oluşturulan veri bilimi platformları sıralamasında 2018 yılı Mayıs ayı listesinde 5. sırada yer almaktadır [94].

Bu çalışmada basit arayüz, kullanıcının kendi verilerini belirli bir sınıra kadar ücretsiz depolayabilme imkânı, okul elektronik postası ile ücretsiz üyelik sağlandığı için Azure Machine Learning platformu tercih edilmiştir.

Gerçekleştirilen çalışmada oluşturulan öneri sistemine katılımcılara ait oy verileri yüklenmiş ve oluşturulan sistemin doğruluk oranı ölçülmüştür.

4.6.1. Azure machine learning ile öneri sistemi oluşturma

AML ile öneri sistemi oluşturmak için sisteme giriş yaptıktan sonra New > Experiment > Blank Experiment ile boş bir çalışma ortamı oluşturulur. Eğer hazır veri seti kullanılacaksa açılan ekrandan sürükleyip bırak yöntemi ile proje alanına taşınır. Kişi kendi veri setini kullanmak isterse New > Dataset > From Local File ile veri setini sisteme yükleyebilir. Öneri sisteminde kullanılacak veride sıralama “kullanıcı ID, nesne ID, oy değeri” şeklinde olmalıdır. Yüklenecek olan dosya csv uzantılı ise tüm sütunlar noktalı virgül yerine virgül ile ayrılmalıdır.

Oy verisi sisteme eklendikten sonra “Edit Metadata” ile öneri sisteminde kullanılacak oy bilgisinin hangi sütun adı altında yer aldığı belirtilir. Ayrıca oy verisinin tam sayı veya ondalıklı bir sayı olup olmadığı bilgisi de bu alanda seçilir.

Oluşturulan sistemde, sisteme yüklenen tüm veriler sayısaldir. Bu nedenle, sistem öneride bulunduğunda nesneye ait tanımlayıcı değer (identifier, ID) yerine nesnenin adı görüntülenmek istenebilir. Bu işlem nesneye ait tanımlayıcı değer ile nesne adının bulunduğu bir dosya sisteme yüklenmiş ve oy verilerinin bulunduğu dosya ile birleştirilmiştir. Birleştirme işlemi için “Join Dataset” modülü kullanılmış ve iki dosyada da ortak olan nesne tanımlayıcı değerinin bulunduğu sütunlar seçilmiştir. “Select Columns in Dataset” modülü ile sisteme dâhil edilmek istenen sütunlar seçilmiştir. Seçilen sütunlar sırası ile kullanıcı ID, nesne adı ve oy değerlerinin yer aldığı sütunlardır.

Kullanılan veriler “Split Data” modülü ile eğitim ve test olmak üzere ikiye ayrılır. Bu modül üzerinde yer alan ilk çıkış eğitim verisi, diğer çıkış ise test verisini barındırır. Bu modül seçili iken özellikler kısmından “Recommender Split” seçilir. Bu seçimden sonra bölme oranları belirlenir. Veriler ilk olarak %75 eğitim verisi %25 test verisi, daha sonra %80 eğitim verisi %20 test verisi olarak bölünmüştür.

Öneri sisteminde eğitim işlemi için “Train Matchbox Recommender” modülü sisteme eklenir. Bu modülün ilk girişine eğitim verileri bağlanır. Yer alan diğer girişler, sırası ile kullanıcı özellikleri ve nesne özellikleri girişi için mevcuttur. Nesne veya kullanıcıya ait özellikler eklendiğinde, sistem hibrit yaklaşımda bulunur; bu özellikler eklenmeyip sadece oy verileri kullanıldığında işbirlikçi filtreleme gerçekleştirilmiş olur. Ankete katılan kullanıcıların oyları ve kişisel özellikleri kullanılarak hibrit bir öneri sistemi oluşturulmuş ve hangi özelliğin sisteme nasıl etki ettiği ayrıca araştırılmıştır [95]. Gerçekleştirilen tez çalışmasında ise kullanıcıya veya nesneye ait özellikler sisteme eklenmeyip işbirlikçi filtreleme gerçekleştirilmiştir.

Eğitim işlemi sonrası elde edilen veriler ile test verileri için öneride bulunulur. Bu işlem, “Score Matchbox Recommender” modülü ile gerçekleştirilir. Bu modülün ilk girişine eğitim işleminden elde edilen veriler, 2. sırada yer alan girişine test verileri, 5. Sırada yer alan girişe de eğitim için ayrılan veriler bağlanır. Burada ayrıca yapılacak işlem seçilir. Yapılabilecek işlemler; oy tahmini, nesne önerisi, benzer kullanıcılar ve benzer nesnelere önerisidir. Buradan, nesne önerisi seçilmiştir. Önerilecek nesnelere için iki seçenek yer almaktadır. Model oluşturmak için kullanıcıların oyladığı nesnelere veya kullanıcıya yeni öneride bulunmak için

oylanmamış nesnelere seçilebilir. Burada model oluşturmak için oylanmış oylar seçilmiştir. Bu modülün çıkışında sistemin oluşturduğu öneriler yer almaktadır. Kullanıcı isterse “Convert to CSV” modülü ile önerileri csv uzantılı dosya olarak da indirebilir. Önerileri görüntülemek için ise sistem çalıştırıldığında modül üzerine sağ tıklayarak Scored Dataset > Visualize yolu izlenmelidir. Sistem çalıştırıldığında, gezi için olan oy verisi kullanıldığında elde edilen öneriler Şekil 4.9’da gösterilmiştir. Burada yer alan User ifadesi kişinin ID değerini belirtir. Item 1’den Item 5’e kadar sıralanan yerler ise kişilere yönelik önerilerdir. Item 1 bölümünde yer alanlar, her kullanıcıya yönelik yapılan önerilerden ilk sırada yer alan nesnelere aittir.

User	Item 1	Item 2	Item 3	Item 4	Item 5
132	Bogaz Turu	Ayasofya Muzesi	Sultanahmet Cami	Topkapi Sarayi	Heybeli Ada
93	Galata Kulesi	Masumiyet Muzesi	Topkapi Sarayi	Pera Muzesi	Ciragan Sarayi
118	Masumiyet Muzesi	Topkapi Sarayi	Galata Kulesi	Ayasofya Muzesi	Sultanahmet Cami
205	Yildiz Parki	Yedikule Surlari	YogurtCu Parki	Yerebatan Sarnici	Vialand
4	Galata Kulesi	Masumiyet Muzesi	Topkapi Sarayi	Ayasofya Muzesi	Sultanahmet Cami
55	Galata Kulesi	Topkapi Sarayi	Masumiyet Muzesi	Ayasofya Muzesi	Sultanahmet Cami
133	Oyuncak Muzesi	Bogaz Turu	Ayasofya Muzesi	Sultanahmet Cami	Eyup Sultan Cami
92	Galata Kulesi	Masumiyet Muzesi	Pera Muzesi	Otagtepe	Topkapi Sarayi

Şekil 4.9. Elde edilen öneriler

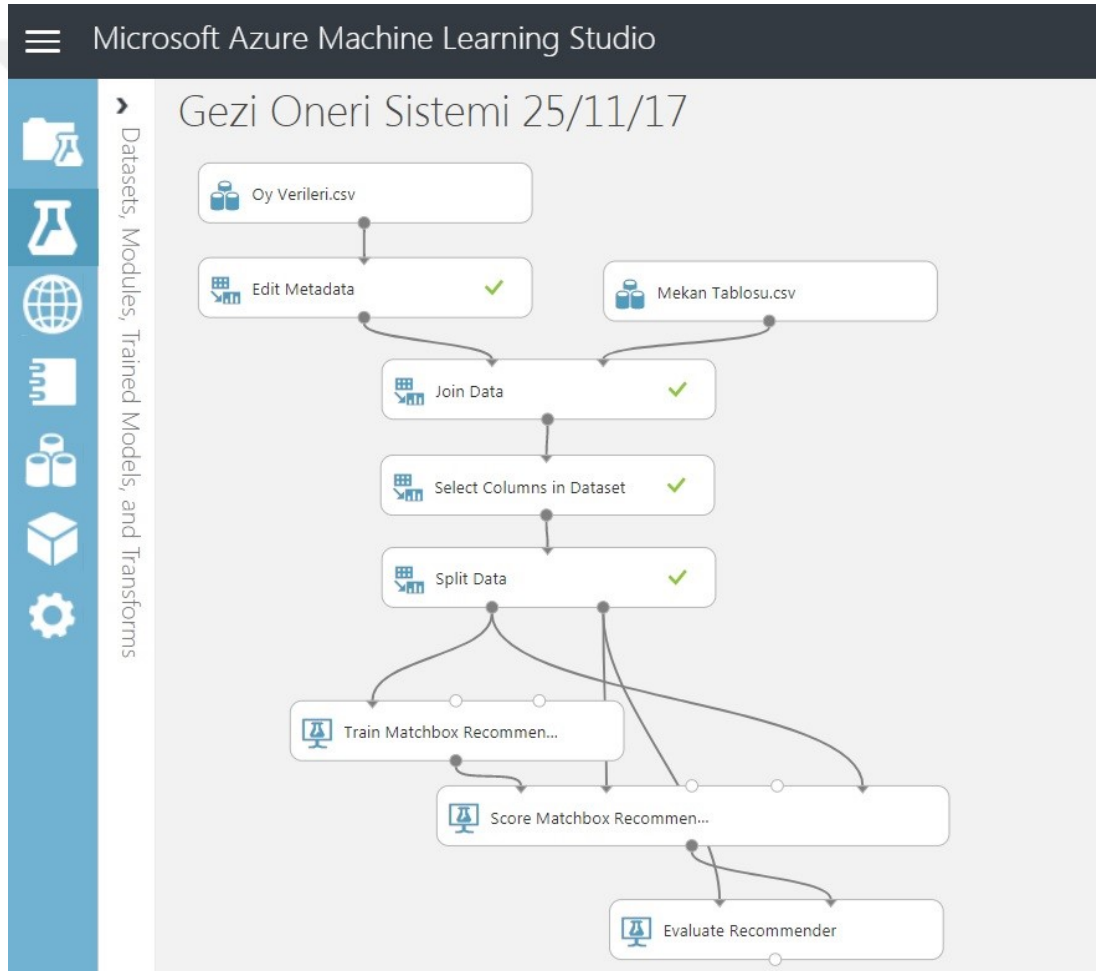
Oluşturulan modelin doğruluk oranını görüntüleyebilmek için “Evaluation Recommender” modülü eklenmiştir. Bu modülün ilk girişine test için ayrılan veriler, 2. girişine ise Scored Matchbox Recommender modülünden elde edilen veriler bağlanmıştır. Oluşturulan sistem çalıştırıldığında bu modül üzerine sağ tıklayıp Metric > Visualize seçenekleri seçildiğinde sistemin doğruluk oranı görüntülenir. Burada yer alan değer NDCG ile elde edilen sonuçtur. Sonuç aralığı [0,1]’dir [96]. Cumulative Gain ve Discounted Cumulative Gain hesaplanması Denklem (4.21) ve Denklem (4.22)’de verilmiştir [97, 98].

Bu ifadelerde yer alan relevance değeri her bir nesne için alakalı olma durumuna göre belirlenir. Denklem (4.22)'nin normalize edilmesi ile NDGC hesaplanır.

$$CG = \sum_{i=1}^n \text{relevance}_i \quad (4.21)$$

$$DCG = \sum_{i=1}^n \frac{\text{relevance}_i}{\log(i+1)} \quad (4.22)$$

Tüm modüller sisteme eklendiğinde oluşan ekran görüntüsü Şekil 4.10'da verilmiştir.



Şekil 4.10. Oluşturulan öneri sistemi

Burada oluşturulan modelin başarı ölçümü gerçekleştirildikten sonra, kullanıcılara daha önce oy vermedikleri nesnelere önermek için Score Matchbox modülü seçilip "from Unrated Items" seçilmelidir. Kullanıcılara yeni nesne önerisinde bulunurken model doğruluk ölçümü için kullanılan modül, oluşturulan sistemden

kaldırılmalıdır. Yeni nesne önerileri “Score Matchbox Recommender” modülüne sağ tıklanıp görüntülenebilir. Kullanıcı burada “Convert to CSV” modülünü “Score Matchbox Recommender” modülü çıkışına ekleyerek önerileri csv uzantılı dosya olarak da indirebilir. Oluşturulan sistemden elde edilen öneri sonuçları Ek – A1’de gösterilmiştir.

Kullanıcılara daha önce oy vermedikleri nesnelere önerildiğinde oy verilen nesnelere de aralarında olabileceği görülmüştür. Bu sorunun önüne geçmek için bir masaüstü uygulaması gerçekleştirilmiştir. Gerçekleştirilen uygulamada tüm oy verileri ve AML’den elde edilen öneri verileri sisteme yüklenir ve karşılaştırılır. Önerilen nesneye ait kullanıcının oy verisi varsa nesne önerilerden silinir. Bu işlem sonucunda bazı kullanıcılara önerdiği nesnelere tümünün kişinin oylarıyla oluşturulduğu veya sonuçlarda yer alan kişinin tüm nesnelere değerlendirdiği gözlenmiştir. Oy değeri bulunan nesnelere öneriden silinmesi sonucu elde edilen sonuçlar Ek – A2’de gösterilmiştir.

5. ÖNERİ SİSTEMLERİNDE İŞBİRLİKÇİ FİLTRELEMENİN UYGULANMASI VE KARŞILAŞTIRILMASI

5.1. Giriş

Bu bölümde önceki bölümlerde bahsedilen Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve SGD ile oluşturulan matris faktörizasyonu yöntemleri iki veri seti ile uygulanmış ve yöntemlerle ilgili karşılaştırılma verilmiştir.

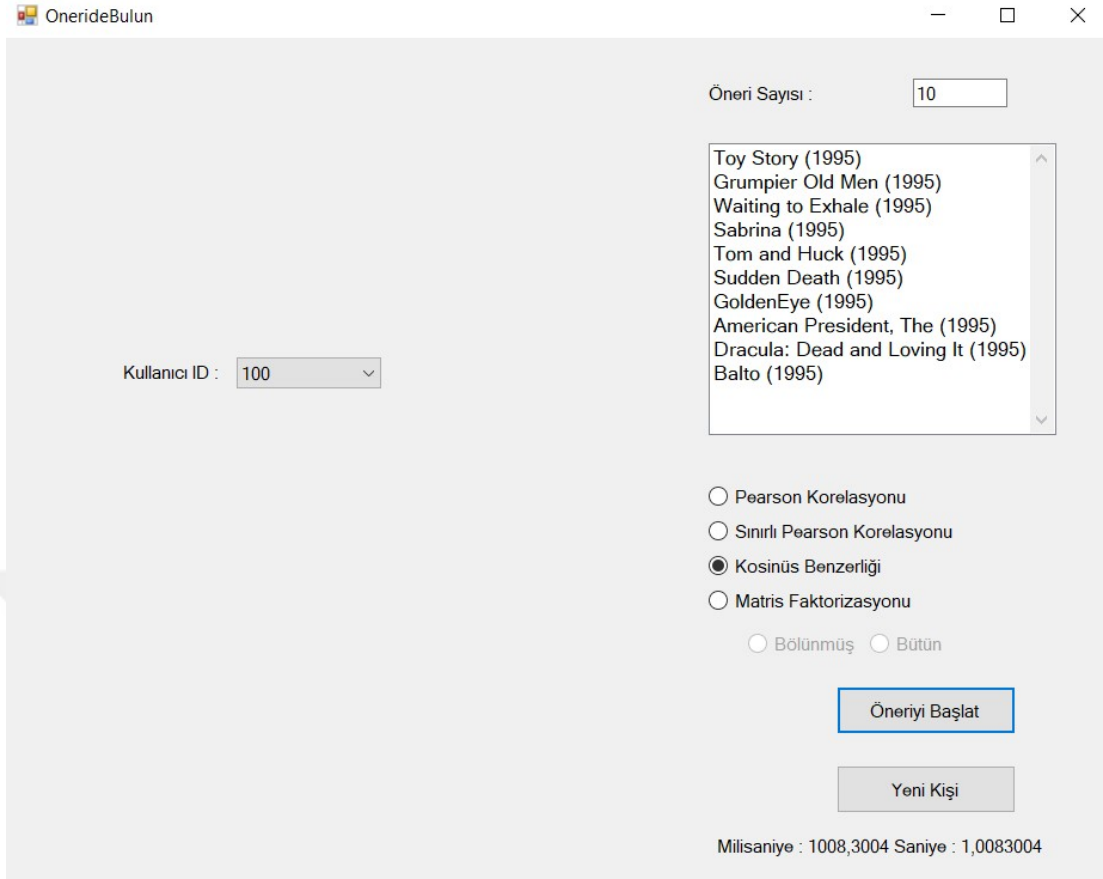
5.2. İşbirlikçi Filtreleme ile Öneri Sistemi Oluşturulması

Önceki bölümlerde bahsi geçen Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve SGD ile oluşturulan matris faktörizasyonu yöntemleri kullanılarak seyahat edilecek yer önerisinde bulunan bir masaüstü uygulaması gerçekleştirilmiştir.

Geliştirilen uygulamada oy verileri ve nesne verileri otomatik olarak sisteme yüklenir. Kullanılacak hesaplama yöntemini ve işlemler sonucunda kaç adet öneri istediğini belirtir. İşlemler sonunda, sistemin öneride bulunması için çalışmayı başlatır. Belirlediği öneri sayısı kadar yer, en yüksek puan verebileceği yerden itibaren sıralı olarak listelenir. Eğer kişinin daha önce ziyaret etmediği bir yer yok ise sistem öneride bulunamayacaktır.

Burada kişinin oy verebileceği nesne sayısı 50 veya 50'den az ise ekranda nesne isimleri görüntülenir ve kullanıcı ziyaret ettiği yerleri puanlandırır. Nesne sayısının 50'den fazla olması durumunda ise kullanıcıların ID değerleri görüntülenir ve kişi içlerinden birini seçer. Buradaki nesne sınırı, ekran kısıtlaması yüzünden maksimum 50 olarak seçilmiştir.

Oylanabilecek nesne sayısının 50'den fazla ve önerilecek nesne sayısının 10 olduğu durumda karşılaşılabilecek ekran görüntüsü Şekil 5.1'de gösterilmiştir. Burada yer alan ekran görüntüsü MovieLens veri seti ile 100 numaralı kullanıcıya ait oy verileri ile kosinüs benzerliği yöntemi seçildiğinde elde edilmiştir.



Şekil 5.1. Kosinüs benzerliği ile 100 numaralı kişi için elde edilen öneriler

MovieLens verilerinde yer alan 100 numaralı kullanıcı için diğer yöntemlerden elde edilen öneriler de hesaplanmıştır. Pearson korelasyonu, sınırlı Pearson korelasyonu yöntemleri ile elde edilen öneriler Şekil 5.1’de kosinüs benzerliği yöntemi ile elde edilen önerilerle tamamen aynıdır. Matris faktörizasyonu ile elde edilen öneriler de Tablo 5.1’de gösterilmiştir.

Tablo 5.1. Diğer yöntemlerle 100 numaralı kişi için elde edilen öneriler

Yöntem Sıra No	Matris Faktörizasyonu	
	Bölünmüş	Bütün
1	Counselor, The (2013)	Water for Elephants (2011)
2	Star 80 (1983)	With a Friend Like Harry... (Harry, un ami qui vous veut du bien) (2000)
3	Librarian, The: The Curse of the Judas Chalice (2008)	Porco Rosso (Crimson Pig) (Kurenai no buta) (1992)
4	Pek Yakında (2014)	Green Lantern (2011)
5	Tooth Fairy (2010)	As We Were Dreaming (2015)
6	Babies (Bébé(s)) (2010)	Waking the Dead (2000)
7	Valkyrie (2008)	Fingers (1978)
8	I Don't Know How She Does It (2011)	Bridge to Terabithia (2007)
9	Lightning in a Bottle (2004)	Clockwise (1986)
10	What We Do in the Shadows (2014)	Runaway (1984)

100 numaralı kişi için oluşturulan öneriler, Intel Core i5 işlemcili, 8 GB RAM kapasiteli bilgisayarda Pearson korelasyonu ile 0,646 saniye, sınırlı Pearson korelasyonu ile 0,679 saniye, kosinüs benzerliği ile 1,008 saniyede elde edilmiştir. Matris faktörizasyonu ile işlemler gerçekleştirildiğinde ise bölünmüş olarak hesaplamada 9641,82 saniyede tamamlanırken bütün olarak hesaplamada ise 24432,49 saniyede tamamlanmıştır.

Oylanabilecek nesne sayısının 50, önerilecek nesne sayısının 10 olduğu durumda Pearson korelasyonu ile elde edilen öneriler Şekil 5.2'de gösterilmiştir. Burada yer alan giriş verileri, rastgele oluşturulmuş oy verileridir.

OnerideBulun

Öneri Sayısı : 10

Topkapı Sarayı	0	Pierre Loti Tepesi	0
Dolmabahçe Sarayı	10	Çamlıca Tepesi	0
Çırağan Sarayı	0	Otağtepe	0
Galata Kulesi	0	Atatürk Arboretumu	0
Kız Kulesi	0	Emirgan Korusu	8
Eyüp Sultan Cami	0	Fethi Paşa Korusu	0
Ortaköy Cami	0	Mihrabat Korusu	0
Mihrimah Sultan	0	Cemile Sultan	0
Süleymaniye Cami	0	Gülhane Parkı	0
Sultanahmet Cami	10	Gezi Parkı	0
Ayasofya Müzesi	0	Yoğurtçu Parkı	0
İstanbul Arkeoloji	0	Yıldız Parkı	8
Rahmi Koç Müzesi	0	Rumeli Hisarı	0
Masumiyet Müzesi	0	Anadolu Hisarı	0
Pera Müzesi	0	Haydarpaşa Garı	0
Aşiyen Müzesi	0	Taksim Meydanı	0
Oyuncak Müzesi	0	Büyükkada	8
Panorama 1453	0	Heybeli Ada	0
Miniaturk	8	Kınalı Ada	0
Yerebatan Sarnıcı	0	Burgazada	0
Yedikule Surları	0	Bebek Sahili	0
Cağaloğlu Hamamı	0	Boğaz Turu	9
Kapalı Çarşı	9	Eminönü Balıkçıları	0
Mısır Çarşısı	0	İstanbul Akvaryum	0
Kuzguncuk Evleri	0	Vialand	0

Öneriler:

- Topkapı Sarayı
- Galata Kulesi
- Süleymaniye Cami
- Ayasofya Müzesi
- Çırağan Sarayı
- Kız Kulesi
- Eyüp Sultan Cami
- Ortaköy Cami
- Mihrimah Sultan Cami
- İstanbul Arkeoloji Müzesi

Pearson Korelasyonu
 Sınırlı Pearson Korelasyonu
 Kosinüs Benzerliği
 Matris Faktörizasyonu

Bölünmüş Bütün

Öneriyi Başlat

Yeni Kişi

Milisaniye : 15,6575 Saniye : 0,0156575

Şekil 5.2. Girilen oy değerlerine göre elde edilen öneriler

Şekil 5.2.'de yer alan aynı rastgele giriş verileri kullanılarak diğer yöntemler ile de öneri işlemleri gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.2'de gösterilmiştir. Burada yer alan verilere göre tüm yöntemler ile elde edilen 10 yer önerisinin 5'i ortaktır. Ortak olarak önerilen nesnelere Topkapı Sarayı, Galata Kulesi, Süleymaniye Cami, Ayasofya Müzesi ve Çırağan Sarayı'dır.

Tablo 5.2. Rastgele giriş verilerine göre elde edilen öneriler

Yöntem Sıra No	Sınırlı Pearson Korelasyonu	Kosinüs Benzerliği	Matris Faktörizasyonu	
			Bölünmüş	Bütün
1	Topkapı Sarayı	Ayasofya Müzesi	Süleymaniye Cami	Topkapı Sarayı
2	Çırağan Sarayı	Topkapı Sarayı	Ayasofya Müzesi	Çırağan Sarayı
3	Galata Kulesi	Çırağan Sarayı	Yerebatan Sarnıcı	Ayasofya Müzesi
4	Kız Kulesi	Galata Kulesi	Topkapı Sarayı	Süleymaniye Cami
5	Eyüp Sultan Cami	Kız Kulesi	Çırağan Sarayı	Galata Kulesi
6	Ortaköy Cami	Eyüp Sultan Cami	Anadolu Hisarı	Atatürk Arboretumu
7	Süleymaniye Cami	Süleymaniye Cami	Yedikule Surları	İstanbul Arkeoloji Müzesi
8	Ayasofya Müzesi	İstanbul Arkeoloji Müzesi	Eminönü Balıkçıları	Mısır Çarşısı
9	İstanbul Arkeoloji Müzesi	Panorama 1453	Galata Kulesi	Kız Kulesi
10	Panorama 1453	Yerebatan Sarnıcı	Panorama 1453	Pierre Loti Tepesi

Tablo 5.2’de yer alan öneriler elde edilirken işlemde geçen süre Tablo 5.3’te verilmiştir. Tablo 5.3’te yer alan verilere göre doğru tahmin/eksik veri sayısı oranı en yüksek yöntem olan matris faktörizasyonu, diğer yöntemler ile karşılaştırıldığında en yavaş öneri üreten yöntem olmuştur. Matris faktörizasyonu yönteminin diğer yöntemlerden daha yavaş öneri oluşturmasının sebebi girişteki oy değerlendirmesi bulunmayan nesnelere birlikte sistemdeki tüm oy değerlendirmesi bulunmayan nesnelere ait de oy hesaplaması yapması olabilir.

Tablo 5.3. Öneride bulunurken geçen süre

Yöntem Süre	Pearson Korelasyonu	Sınırlı Pearson Korelasyonu	Kosinüs Benzerliği	Matris Faktörizasyonu	
				Bölünmüş	Bütün
Saniye	0,0156575	0,0279248	0,0239137	39,282045	115,1192621
Milisaniye	15,6575	27,9248	23,9137	39282,045	115119,2621

Tüm oy verileri içerisinde rastgele 5 kullanıcı seçilmiştir. Seçilen kullanıcıların kullandıkları oylardan rastgele seçilen 5 tanesi ziyaret edilmedi anlamına gelen sıfır oy değeri ile değiştirilmiştir. Burada amaçlanan, bu kişilere yapılacak önerilerde oy değerleri sıfır olarak değiştirilen nesnelere önerilip önerilmeyeceğini gözlemektir. Bu amaç doğrultusunda rastgele seçilen kişiler 14, 73, 133, 171 ve 205 numaralı kullanıcılarıdır.

14 numaralı kullanıcıya ait 6, 12, 24, 36 ve 47; 73 numaralı kullanıcıya ait 6, 13, 26, 35 ve 48; 133 numaralı kullanıcıya ait 10, 20, 27, 36 ve 45; 171 numaralı kullanıcıya ait 8, 19, 37, 42 ve 48 ve 205 numaralı kullanıcıya ait 4, 11, 23, 35 ve 42 numaralı nesnelere verilen oylar sistemde ziyaret edilmedi anlamına gelen sıfır oy değeri ile değiştirilmiştir. Öneri sayısı ilk 5 nesne olarak seçildiğinde sıfır oy değeri verilerek çıkarılan nesnelere ilk 5 arasına giren nesnelere Tablo 5.4’te gösterilmiştir.

Tablo 5.4. Oy verilen nesnelere gizlendiğinde elde edilen veriler

Yöntem Kişi Sıra No	Pearson Korelasyonu	Sınırlı Pearson Korelasyonu	Kosinüs Benzerliği	Matris Faktörizasyonu	
				Bölünmüş	Bütün
14	6 - Eyüp Sultan Cami	47- Boğaz Turu	47- Boğaz Turu	47- Boğaz Turu	12 - İstanbul Arkeoloji Müzesi
	12 - İstanbul Arkeoloji Müzesi	6 - Eyüp Sultan Cami	6 - Eyüp Sultan Cami	-	24 - Mısır Çarşısı
	-	12 - İstanbul Arkeoloji Müzesi	12 - İstanbul Arkeoloji Müzesi	-	-
73	6 - Eyüp Sultan Cami	6 - Eyüp Sultan Cami	6 - Eyüp Sultan Cami	6 - Eyüp Sultan Cami	6 - Eyüp Sultan Cami
133	10 - Sultanahmet Cami	10 - Sultanahmet Cami	10 - Sultanahmet Cami	27 - Çamlıca Tepesi	-
	-	-	-	45 - Burgazada	-
171	8 - Mihrimah Sultan Cami	8 - Mihrimah Sultan Cami	42 - Büyükkada	-	48 -Eminönü Balıkçıları
205	4 - Galata Kulesi	4 - Galata Kulesi	4 - Galata Kulesi	4 - Galata Kulesi	-
	-	11 -Ayasofya Müzesi	11 -Ayasofya Müzesi	11 -Ayasofya Müzesi	-
	-	23 -Kapalı Çarşı	23- Kapalı Çarşı	23 - Kapalı Çarşı	-
	-	42 -Büyükkada	42 -Büyükkada	-	-

Tablo 5.4’te yer alan veriler incelendiğinde başlangıçta çıkarılan yerlerin genellikle öneriler arasında yer aldığı görülmüştür. Hiçbir kullanıcı için çıkarılan 5 nesnenin tamamının önerilerde yer almadığı görülmüştür. 205 numaralı kullanıcı incelendiğinde çıkarılan 5 nesneden maksimum 4 tanesinin önerilerde yer aldığı görülmüştür. Sistem öneride bulunurken oy verisi olmayan nesnelere için tahmin yapıp büyükten küçüğe olacak şekilde sıraladığı için ilk 5 nesne arasında en fazla 4 nesnenin eşleştiği görülmüştür. Önerilen nesne sayısı 5 yerine 10 olarak seçildiğinde birçok öneride başlangıçta çıkarılan 5 nesnenin de önerilerin bulunduğu listede yer aldığı gözlemlenmiştir.

Ayrıca tüm oy verilerinden, 5 kişiye ait tüm oy verileri sırayla sistemden çıkarılmış ve bu kişilere kullanılan yöntemle göre nasıl öneride bulunacağı gözlemlenmek istenmiştir. Seçilen kullanıcıların oy değerleri ile her bir yöntemden 3 adet öneri oluşturulmuştur. Elde edilen veriler Tablo 5.5'te gösterilmiştir. Sistem seçilen kişilere öneride bulunurken hesaplama, Intel Core i5 işlemcili, 8 GB RAM kapasiteli bilgisayarda 11 milisaniye ile 179 saniye arasında değişmiştir. Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliğinde işlemler milisaniye bazında gerçekleşirken matris faktörizasyonu ile yapılan hesaplamalarda geçen süre saniye bazında gerçekleşmiştir.

Tablo 5.5. Rastgele seçilen kişilere ait oluşturulan öneriler

Yöntem Kişi Sıra No	Pearson Korelasyonu	Sınırlı Pearson Korelasyonu	Kosinüs Benzerliği	Matris Faktörizasyonu	
				Bölünmüş	Bütün
14	Çırağan Sarayı	Çamlıca Tepesi	Kız Kulesi	Büyükkada	Büyükkada
	Kız Kulesi	Atatürk Arboretumu	Pierre Loti Tepesi	Haydarpaşa Garı	Haydarpaşa Garı
	Oyuncak Müzesi	Büyükkada	Çamlıca Tepesi	Anadolu Hisarı	Anadolu Hisarı
73	Topkapı Sarayı	Topkapı Sarayı	Boğaz Turu	Mihrimah Sultan Cami	Yedikule Surları
	Çırağan Sarayı	Kız Kulesi	Topkapı Sarayı	Büyükkada	Emirgan Korusu
	Kız Kulesi	Yerebatan Sarnıcı	Çırağan Sarayı	Çırağan Sarayı	Yerebatan Sarnıcı
133	Topkapı Sarayı	Topkapı Sarayı	Topkapı Sarayı	Fethi Paşa Korusu	Otağtepe
	Dolmabahçe Sarayı	Dolmabahçe Sarayı	Dolmabahçe Sarayı	Kınalı Ada	Atatürk Arboretumu
	Süleymaniye Cami	Atatürk Arboretumu	İstanbul Arkeoloji Müzesi	Atatürk Arboretumu	Mihrabat Korusu
171	Ayasofya Müzesi	Çırağan Sarayı	Ayasofya Müzesi	Vialand	İstanbul Arkeoloji Müzesi
	Çırağan Sarayı	Ayasofya Müzesi	Çırağan Sarayı	İstanbul Arkeoloji Müzesi	Vialand
	İstanbul Arkeoloji Müzesi	İstanbul Arkeoloji Müzesi	İstanbul Arkeoloji Müzesi	Rahmi Koç Müzesi	Yedikule Surları
205	Çırağan Sarayı	Kız Kulesi	Kız Kulesi	Panorama 1453	İstanbul Akvaryum
	Kız Kulesi	Süleymaniye Cami	İstanbul Arkeoloji Müzesi	Miniattürk	Panorama 1453
	Süleymaniye Cami	İstanbul Arkeoloji Müzesi	Miniattürk	Heybeli Ada	Kuzguncuk Evleri

Tablo 5.5'te yer alan veriler incelendiğinde elde edilen öneriler arasında genellikle benzer yerlerin bulunduğu söylenebilir. Örneğin; 171 numaralı kişi için tüm yöntemlerde İstanbul Arkeoloji Müzesi önerilerde ilk üç arasında yer almıştır. Ayrıca 171 numaralı kişinin oy verileri ile Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliği ile elde edilen önerilerde ilk üçte yer alan tüm yerlerin aynı olduğu, sadece sınırlı Pearson korelasyonunda ilk iki yerin sıralamasın kendi arasında yer değiştirdiği gözlenmiştir. 171 numaralı kişinin oy verileri ile matris faktörizasyonu ile matris bölünmüş ve bütün olarak ele alındığında elde edilen önerilerde ise Vialand ve İstanbul Arkeoloji Müzesi'nin ilk ikide yer aldığı fakat sıralamasının farklı olduğu gözlenmiştir.



6. BULGULAR VE TARTIŞMA

Çalışmada kullanılan bellek tabanlı işbirlikçi filtreleme yöntemleri olan Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliği yöntemleri ile eğitim ve test işlemleri sonrası gezi öneri sistemi için kullanılan oylar ile elde edilen hata değerleri Tablo 6.1’de gösterilmiştir. Burada yer alan “Doğru Tahmin/Eksik Veri Sayısı” oranı, oy değerlerinin [0,10] aralığında olduğu durumda hesaplanır. Tablo 6.1’de yer alan veriler incelendiğinde Pearson korelasyonunun oy tahmininde oldukça zayıf olduğu gözlenebilir. Bellek tabanlı filtrelemede kullanılan bu yöntemlerde, gezi öneri sistemi için kullanılan oylarda en başarılı yöntemin kosinüs benzerliği olduğu görülmüştür.

Tablo 6.1. Geleneksel yöntemler ile elde edilen hata değerleri

Yöntem	Doğru Tahmin/Eksik Veri Sayısı	MAE	MSE	RMSE
Pearson Korelasyonu	%26,4	1,828	9,006	3,001
Kosinüs Benzerliği	%27,6	1,289	2,938	1,714
Sınırlı Pearson Korelasyonu	%28,4	1,268	2,873	1,695

Gezi öneri sistemi için elde edilen oylar ile matris faktörizasyonu yöntemlerinden biri olan SGD kullanılarak 5000 iterasyonda $\alpha=0,0002$, $\beta=0,2$ iken farklı K değerleri ile elde edilen sonuçlar Tablo 6.2’de gösterilmiştir. Burada MAE değerinin gösterilmemesinin sebebi, regülasyon ile yapılan tahmini oy hesabında MAE hesabının nasıl değişeceği ile ilgili net bir görüş olmamasındandır. MAE hesabı regülasyon işlemi göz ardı edilip Denklem (2.3)’te verilen şekilde hesaplandığında, 5000 iterasyonda $\alpha=0,0002$, $\beta=0,2$ ve K değeri 15 iken 0,642 olarak; K değeri 30 iken 0,286 olarak hesaplanmıştır. Tablo 6.1 ve Tablo 6.2’de yer alan değerler incelendiğinde en başarılı yöntemin SGD olduğu gözlemlenmiştir.

Tablo 6.2. Gezi öneri sisteminde SGD ile elde edilen hata değerleri

Elde edilen oranlar / K değerleri	Doğru Tahmin/Eksik Veri Sayısı	MSE	RMSE
K=15	%55,35	1,051	1,025
K=30	%82,69	0,497	0,705
K=45	%99,53	0,3667	0,6056

Tablo 6.2 incelendiğinde K değeri arttıkça doğruluk oranının arttığı gözlemlenir. Fakat bu yöntemde kullanılan K değerinin düşük olması gerekmektedir. Yüksek K değeri iş yükünü arttıracak gibi başlangıçta sistemin hata değerinin de yüksek olmasına sebep olacaktır.

Çalışmada kullanılan Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliği ile gerçekleştirilen eğitim ve test işlemleri MovieLens veri seti kullanılarak da tekrarlanmıştır.

671 kullanıcı ve 9066 film içeren MovieLens 100K veri setinde kullanıcı oyları, 0,5'ten başlayarak maksimum 5 olacak şekilde 0,5 artmaktadır. Bu oy verileri kullanıldığında elde edilen hata değerleri Tablo 6.3'te gösterilmiştir. Sınırlı Pearson korelasyonu yönteminde elde edilen değerlerde tarafsız oy değeri 3 olarak seçilmiştir.

Tablo 6.3. MovieLens ile elde edilen hata değerleri

Hata Tipi / Yöntem	MAE	MSE	RMSE
Pearson Korelasyonu	0,7215	0,8788	0,9374
Sınırlı Pearson Korelasyonu	0,7306	0,9033	0,9504
Kosinüs Benzerliği	0,72	0,8729	0,9343

Gezi öneri sistemi için kullanılan oy verileri ile elde edilen hata değerlerinin yer aldığı Tablo 6.1 ile MovieLens veri setinde yer alan değerler ile elde edilen hata değerlerinin yer aldığı Tablo 6.3'te ortak olarak yer alan yöntemler olan Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliği ile elde edilen sonuçlar karşılaştırılmıştır.

Tablo 6.1 ve Tablo 6.3 incelendiğinde göre en yüksek hata değerinin gezi oy verilerinde Pearson korelasyonu; MovieLens oy verilerinde ise sınırlı Pearson korelasyonu ile elde edildiği gözlenmiştir.

MovieLens veri seti ile SGD yönteminde 250 iterasyonda $\alpha=0,0002$, $\beta=0,2$ iken farklı K değerleri ile elde edilen sonuçlar Tablo 6.4'te gösterilmiştir. Doğru tahmin/Eksik veri sayısı sadece 0-10 aralığındaki tüm oy değerleri tam sayı olan oy verileri için hesaplandığından tabloda bu değerler gösterilmemiştir.

Tablo 6.4. MovieLens verilerinde SGD ile elde edilen hata değerleri

Elde edilen oranlar K değerleri	MSE	RMSE
K=10	0,7062	0,8404
K=50	0,6075	0,7794
K=100	0,5958	0,7719
K=250	0,8169	0,9038

Gezi önerilerinde kullanılan oy verileri kullanılarak veri bilimi platformlarından biri olan AML ile oluşturulan öneri sistemine ait hata değerleri Tablo 6.5'te gösterilmiştir. Burada yer alan değerler, Score Matchbox Modülü üzerinde "Rating Prediction" seçildiğinde elde edilen değerlerdir. Tablo 6.1, Tablo 6.2 ve Tablo 6.5 incelendiğinde, gezi oy verileri için oy tahmininde en yakın sonuçları bulan yöntemin SGD olduğu görülmüştür.

Tablo 6.5. AML üzerinde oy tahmini ile elde edilen değerler

Hata Tipi Eğitim ve Test Veri Oranı	MAE	RMSE
%75 eğitim - %25 test	6,717593	7,216374
%80 eğitim - %20 test	4,395742	5,574288

Score Matchbox Modülü ile ayrıca oy tahmini işlemi yapan "Rating Prediction" yerine nesne önerisi anlamına gelen "Item Recommendation" seçildiğinde elde edilen sonuçlar Tablo 6.6'da gösterilmiştir. Bu tablo incelendiğinde eğitim verisinin oranı yüksek olduğunda doğru tahmin oranının da arttığı görülmüştür.

Tablo 6.6. AML üzerinde nesne önerisi ile elde edilen değerlerler

Değerlendirme Yöntemi	NDCG
Eğitim ve Test Veri Oranı	
%75 eğitim - %25 test	0,8508
%80 eğitim - %20 test	0,8649

MovieLens önerilerinde kullanılan oy verileri kullanılarak da AML üzerinde ölçümler gerçekleştirilmiştir. Oluşturulan öneri sistemine ait hata değerleri Tablo 6.7’de gösterilmiştir. Burada yer alan değerler, Score Matchbox Modülü üzerinde “Rating Prediction” seçildiğinde elde edilen değerlerdir. Tablo 6.3, Tablo 6.4 ve Tablo 6.7 incelendiğinde, MovieLens verileri için de oy tahmininde en yakın sonuçları bulan yöntemin SGD olduğu görülmüştür.

Tablo 6.7. MovieLens ile oy tahmininde elde edilen hata değerleri

Hata Tipi	MAE	RMSE
Eğitim ve Test Veri Oranı		
%75 eğitim - %25 test	0,675784	1,080404
%80 eğitim - %20 test	0,710048	1,127737

MovieLens oy verileri kullanılarak Score Matchbox Modülü ile ayrıca oy tahmini işlemini yapan “Rating Prediction” yerine nesne önerisi anlamına gelen “Item Recommendation” seçildiğinde elde edilen sonuçlar Tablo 6.8’de gösterilmiştir. Bu tablo incelendiğinde gezi oy verilerinde de olduğu gibi eğitim verisinin oranı yüksek olduğunda doğru tahmin oranının da arttığı görülmüştür.

Tablo 6.8. Nesne önerisi ile elde edilen değerlerler

Değerlendirme Yöntemi	NDCG
Eğitim ve Test Veri Oranı	
%75 eğitim - %25 test	0,867935
%80 eğitim - %20 test	0,866299

AML ile oluşturulan gezi öneri sisteminden elde edilen önerileri karşılaştırmak için matris faktörizasyonu işlemi ile 5000 iterasyonda, $K=15$, $\alpha=0,0002$, $\beta=0.2$ değerleri kullanılarak matrisin ikiye bölünüp ayrı ayrı faktörizasyon işleminin gerçekleştirilmesi ile öneriler elde edilmiştir. Elde edilen öneriler Ek – B’de verilmiştir. Matrisin bölünerek faktörizasyon işleminin yapılması sonucu elde edilen öneriler, AML sisteminden elde edilen öneriler ile karşılaştırıldığında çoğu kişide iki sistemin de benzer öneride bulunduğu gözlenmiştir. Bu durumda AML gezi oy verileri için her ne kadar oy tahmininde yetersiz kalsa da önerilerinin SGD ile benzer olması sebebiyle nesnelere alakalı olarak önerdiği söylenebilir.



7. SONUÇLAR VE ÖNERİLER

Öneri sistemleri günümüzde aktif araştırma konularından biridir. Öneri sistemlerinin giderek yaygınlaşması ile beraber kullanıcılara doğru öneriler sunmak da gün geçtikçe önem kazanmaktadır. Bu çalışmada işbirlikçi filtreleme tekniği ile oluşturulan öneri sistemlerinde, kullanılan hesaplama yöntemlerinin sisteme olan etkisi araştırılmıştır.

Öneri sistemleri oluştururken kullanılan benzerlik hesaplama yöntemlerinden Pearson korelasyonu, sınırlı Pearson korelasyonu ve kosinüs benzerliği ile ölçümler yapılmıştır. Model tabanlı işbirlikçi filtreleme yöntemlerinden biri olan ve literatür incelendiğinde de başarılı olduğu görülen matris faktörizasyonu ile de öneri sistemi oluşturulup ölçümler yapılmıştır. Çalışmada matris faktörizasyonu yöntemlerinden biri olan SGD kullanılmıştır. Ayrıca hazır bir platform olan AML ile de öneri sistemleri oluşturulmuş ve tüm bu yöntemlerden elde edilen sonuçlar karşılaştırılmıştır. Gerçekleştirilen çalışma ile her bir yöntem ile farklı sonuçlar elde edildiği gösterilmiştir.

Elde edilen sonuçlara göre, klasik yöntemler olan Pearson korelasyonu, sınırlı Pearson korelasyonu, kosinüs benzerliği ve SGD karşılaştırıldığında, en başarılı yöntemin SGD olduğu görülürken en zayıf yöntemin gezi oy verileri için Pearson korelasyonu, MovieLens oy verileri için ise sınırlı Pearson korelasyonu olduğu görülmüştür.

Veri bilimi platformundan oy tahmini için elde edilen değerler ile kullanılan yöntemlerden elde edilen değerler karşılaştırıldığında AML'nin yetersiz olduğu görülmüştür. Fakat AML üzerindeki nesne önerilerine bakıldığında nesne tahmin oranının %85 değerinin üzerinde olması oy tahmininde her ne kadar yetersiz olsa da nesnelere alakalı olarak önerdiği söylenebilir. Bu sebeple, bu çalışmada kullanılan veriler baz alındığında AML ile oy tahmini işleminden ziyade nesne önerisi işleminin kullanılmasının daha uygun olduğu söylenebilir.

SGD yönteminde yer alan gizli özellik değeri incelendiğinde ise bu değer yüksek seçildiğinde iterasyonlar sonucu elde edilen hata değeri her ne kadar düşük olsa da başlangıçta oluşturulan öneri sisteminin hata değerinin çok yüksek olduğu görülmüştür. Başlangıçta sistemin hata değeri çok yüksek olduğunda ise bu oranı düşürmek için iterasyon sayısının da fazla olması gerekebilir. Diğer çalışmalarda da belirtildiği gibi bu gizli özellik değerinin küçük olması, sistemin hata değerinin başlangıçta büyük olmasının önüne geçtiği için avantaj sağlamaktadır.



KAYNAKLAR

- [1] Rich E., User Modeling Via Stereotypes, *Cognitive Science*, 1979, **3**(4), 329-354.
- [2] Ekstrand M.D., Riedl J.T., Konstan J.A, Collaborative Filtering Recommender Systems, *Foundations and Trends® in Human-Computer Interaction*, 2010, **4**(2), 81-173.
- [3] Burke R., Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, 2002, **12**(4), 331 – 370.
- [4] Shani G., Gunawardana A., Evaluating Recommendation Systems, Editors: Ricci F., Rokach L., Shapira B., Kantor P.B., *Recommender Systems Handbook*, 1st ed., Springer US, USA, 257-297, 2011.
- [5] Vozalis M., Margaritis K. G., Collaborative filtering enhanced by demographic correlation, *18th World Computer Congress*, Toulouse, France, 22-27 Ağustos 2004.
- [6] Özgöbek Ö., Erdur R.C., Öneri Sistemleri ve Bir Uygulama Alanı Olarak Haber Öneri Sistemleri, *Akademik Bilişim Konferansları*, Eskişehir, Türkiye, 4-6 Şubat 2015.
- [7] Linden G., Smith B., York J., Amazon.com Recommendations: Item-to-item Collaborative Filtering, *IEEE Internet Computing*, 2003, **7**(1), 76-80.
- [8] Tian H., Liang P., Improved Recommendation Based on Trust Relationships in Social Networks, *Future Internet*, 2017, **9**(1), Article 9.
- [9] Lvn R., Wang Q., Raj J. D., Recommending News Articles Using Cosine Similarity Function, *SAS Global Forum*, Washington, USA, 23-26 March, 2014.
- [10] Singh A. K., Soundarabai P. B., Collaborative Filtering in Movie Recommendation System Based on Rating and Genre, *International Journal of Advanced Research in Computer and Communication Engineering*, 2017, **6**(3), 465 – 467.
- [11] Amatriain X., Basilico J., Netflix Recommendations: Beyond The 5 Stars(Part 1), Medium, <https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429> (Ziyaret Tarihi: 29 Ocak 2018).

- [12] Bell R. M., Koren Y., Volinsky C., The BellKor 2008 Solution to the Netflix Prize, Netflix Prize, https://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf (Ziyaret tarihi: 1 Aralık 2017).
- [13] Lee J., Sun M., Lebanon G., A Comparative Study of Collaborative Filtering Algorithms, 2012, arXiv: 1205.3193v1.
- [14] CACHEDA F., CARNEIRO V., FERNANDEZ D., FORMOSO V., Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques And Proposals for Scalable, High – Performance Recommender Systems, *ACM Transactions on The Web*, 2011, **5**(1).
- [15] Al-Bakri N. F., Hashim H. S., A Modified Similarity Measure for Improving Accuracy of User-Based Collaborative Filtering, *Iraqi Journal of Science*, 2018, **59**(2B), 934 – 945.
- [16] Al-Barznji K., Atanassov A., Comparison of Memory Based Filtering Techniques for Generating Recommendations on Large Data, *Engineering and Automation Problems*, 2018, **1**, 44 – 50.
- [17] Sincan O. M., Yıldırım Z., A New Similarity Coefficient for a Collaborative Filtering Algorithm, *Communications, Series A2-A3: Physical Sciences and Engineering*, 2017, **59**(2), 41 – 54.
- [18] Li M., Zheng K., A Collaborative Filtering Algorithm Combined with User Habits for Rating, *Proceedings of International Conference on Logistics Engineering, Management and Computer Science*, 2015, 1279 – 1282.
- [19] Dhawan S., Singh K., Kumar N., Collaborative Filtering Based Product Recommendation System for Online Social Networks, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2017, **2**(5), 126 – 131.
- [20] Menk A., A Hybrid Recommendation System Based on Human Curiosity, *9th ACM Conference on Recommender Systems*, Vienna, Austria, 16th-20th September 2015.
- [21] Kashdan T. B., Gallagher M. W., Silvia P. J., Winterstein B. P., Breen W. E., Terhar D., Steger M. F., The Curiosity And Exploration Inventory-II: Development, Factor Structure And Psychometrics, *Journal of Research in Personality*, 2009, **43**(6), 987-998.
- [22] Uçar T., Developing Intelligent Trip Recommender System, Doktora Tezi, Bahçeşehir Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2016, 453778.
- [23] Ge Y., Xiong H., Tuzhilin A., Liu Q., Cost-Aware Collaborative Filtering for Travel Tour Recommendations, *ACM Transactions on Information Systems*, 2014, **32**(1), 4:1 – 4:31.

- [24] Ravi L., Vairavasundaram S., A Collaborative Location Based Travel Recommendation Systems through Enhanced Rating Prediction for the Group of Users, *Computational Intelligence and Neuroscience*, 2016, Article ID: 1291358.
- [25] Herzog D., Wörndl W., A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip, *The 1st Workshop on New Trends in Content Based Recommender Systems*, California, USA, 6 October 2014.
- [26] Hao Q., Cai R., Wang C., Xiao R., Yang J. M., Pang Y., Zhang L., Equip Tourists with Knowledge Mined From Travelogues, *The 19th International Conference on World Wide Web*, Raleigh, USA, 26 – 30 April 2010.
- [27] Kılınç Ş., İnsanları Sebepsiz Yere Çocuk Sahibi Yapan Amazon'a Sosyal Medyadan Büyük Tepki, *WebTekno*, <http://www.webtekno.com/insanlari-sebepsiz-yere-cocuk-sahibi-yapan-amazon-a-sosyal-medyadan-buyuk-tepki-h33870.html> (Ziyaret Tarihi: 28 Ocak 2018).
- [28] Kennedy S., Potentially Deadly Bomb Ingredients Are 'Frequently Bought Together' on Amazon, *Channel 4 News*, <https://www.channel4.com/news/potentially-deadly-bomb-ingredients-on-amazon> (Ziyaret tarihi: 28 Ocak 2018).
- [29] Anonim, Leaderbord, Netflix Prize, <https://www.netflixprize.com/leaderboard.html> (Ziyaret tarihi: 26 Kasım 2017).
- [30] Özgöbek Ö., Ontolojiler ve Anlamsal Çıkarıma Kullanılarak Bir Öneri Sistemi Tasarlanması, *Doktora Tezi*, Ege Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, 2015, 405297.
- [31] Kapusuzoğlu H., Ontoloji Tabanlı İlişkisel Ürün Öneri Sistemi, *Yüksek Lisans Tezi*, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2011, 349601.
- [32] Milli M., Türkçe İçin E-Öğrenme Ortamlarında Ontoloji Tabanlı Öneri Sistemi, *Yüksek Lisans Tezi*, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, 2016, 427988.
- [33] Yengi Y., Büyük Veride Duygu Analizine Dayalı Öneri Sistemleri, *Yüksek Lisans Tezi*, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2016, 436205.
- [34] Levinas C. A., An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals, *Master's Thesis*, Universitat Politècnica de Catalunya, 2014.
- [35] Sarwar B., Karypis G., Konstan J., Riedl J., Item-Based Collaborative Filtering Recommendation Algorithms, *10th International Conference on World Wide Web*, Hong Kong, China, 01-05 May 2001.

- [36] Aggarwal C.C., *Recommender Systems*, 1st ed., Springer International Publishing, Switzerland, 2016.
- [37] Shardanand U., Maes P., Social Information Filtering: Algorithms for Automating “Word of Mouth”, *SIGCHI Conference on Human Factors in Computing Systems*, Denver, USA, 07-11 May 1995.
- [38] Ahn H. J., A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, *Information Science: An International Journal*, 2008, **178**(1), 37 – 51.
- [39] Jooa J., Bangb S., Parka G., Implementation of a Recommendation System Using Association Rules and Collaborative Filtering, *Procedia Computer Science*, 2016, **91**, 944-952.
- [40] Serradilla F., Bobadilla J., The Effect of Sparsity on Collaborative Filtering Metrics, *Proceedings of The 20th Australasian Conference on Australasian Database*, 2009, 92(1), 9 – 18.
- [41] Sedhain S., Sanner S., Braziunas D., Xie L., Christensen J., Social Collaborative Filtering for Cold Start Recommendations, *The 8th ACM Conference on Recommender Systems*, Foster City, USA, 6-10 October 2014.
- [42] Hu R., Pu P., Enhancing Collaborative Filtering Sytems with Personality Information, *The 5th ACM Conference on Recommender Systems*, Chicago, USA, 23-27 October 2011.
- [43] Lin J., Sugiyama K., Kan M. Y., Chua T. S., Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers, *The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 28 - August 1, 2013.
- [44] Sarwar B. M., Sparsity, Scalability and Distribution in Recommender Systems, Doktora Tezi, University of Minnesota, 2001. ISBN:0-493-04207-5.
- [45] Sarwar B. M., Karypis G., Konstan J., Riedl J., Application of Dimensionality Reduction in Recommender System – A Case Study, Access Number: ADA439541, 2000.
- [46] Sarwar B. M., Karypis G., Konstan J., Riedl J., Recommender Systems for Large Scale E-Commerce: Scalable Neighborhood Formation Using Clustering, *The 5th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, 27-28 December 2002.
- [47] Khusro S., Ali Z., Ullah I., Recommender Systems: Issues, Challenges And Research Opportunities, Editörler: Kim K. J., Joukov N., *Information Science and Applications(ICISA)*, 1st ed., Springer, Singapore, 1179 – 1189, 2016.

- [48] Lau A., Tsui E., Lee W. B., An Ontology Based Similarity Measurement for Problem Based Case Reasoning, *Expert Systems with Applications*, 2009, **36**(3), 6574 – 6579.
- [49] Isinkaye F. O., Folajimi Y. O., Ojokoh B. A., Recommendation Systems: Principles, Methods And Evaluation, *Egyptian Informatics Journal*, 2015, **16**(3), 261 – 273.
- [50] Barwicki A., Collaborative Filtering Recommendation Systems Simple Theory, <http://adrianbarwicki.com/2017/06/16/collaborative-filtering-recommendation-systems-simple-theory/> (Ziyaret tarihi: 28 Aralık 2017).
- [51] Steinweg-Woods J., A Gentle Introduction to Recommender Systems with Implicit Feedback, <https://jessesw.com/Rec-System/> (Ziyaret tarihi: 6 Şubat 2018).
- [52] Rehorek T., Evaluating Recommender Systems: Choosing The Best One for Your Business, Medium, <https://medium.com/recombee-blog/evaluating-recommender-systems-choosing-the-best-one-for-your-business-c688ab781a35> (Ziyaret tarihi: 10 Şubat 2018).
- [53] Hannon J., Bennett M., Smyth B., Recommending Twitter Users To Follow Using Content and Collaborative Filtering Approaches, The 4th ACM Conference on Recommender Systems, Barcelona, Spain, 26-30 September 2010.
- [54] Pazzani M. J., Billsus D., Content-Based Recommendation Systems, Editörler: Brusilovsky P., Kobsa A., Nejdl W., *The Adaptive Web Methods and Strategies of Web Personalization*, 1st ed., Springer-Verlag Berlin Heidelberg, Berlin, 325-341, 2007.
- [55] Vanetti M., Binaghi E., Carminati B., Carullo M., Ferrari E., Content-Based Filtering in On-Line Social Networks, Editörler: Dimitrakakakis C., Divanis A. G., Mitrokotsa A., Verykios V. S., Saygın Y., *Privacy and Security Issues in Data Mining and Machine Learning*, 1st ed., Springer-Verlag Berlin Heidelberg, Berlin, 127-140, 2011.
- [56] Zhang Y., Hong J., I., Cranor L. F., Cantina: A Content-Based Approach to Detecting Phishing Web Sites, *The 16th International Conference on World Wide Web*, Banff, Canada, 8-12 May 2007.
- [57] Meteren R. V., Someren M. V., Using Content Based Filtering for Recommendation, *Machine Learning in The New Information Age: MLnet/ECML2000 Workshop*, Barselona, Spain, 30 May 2000.
- [58] Cantador I., Bellogin A., Vallet D., Content-Based Recommendation in Social Tagging Systems, *4th ACM Conference on Recommender Systems*, Barcelona, Spain, 26-30 September 2010.

- [59] Mooney R. J., Roy L., Content Based Book Recommending Using Learning for Text Categorization, *5th ACM Conference on Digital Libraries*, San Antonio, USA, 02-07 June 2000.
- [60] Taşcı S., İçerik Bazlı Medya Takip ve Haber Tavsiye Sistemi, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2015, 392758.
- [61] Lu Z., Dou Z., Lian J., Xie X., Yang Q., Content Based Collaborative Filtering for News Topic Recommendation, *29th AAAI Conference on Artificial Intelligence*, Austin, USA, 25-30 January 2015.
- [62] Ricci F., Rokach L., Shapira B., Introduction to Recommender Systems Handbook, Editors: Ricci F., Rokach L., Shapira B., Kantor P.B., *Recommender Systems Handbook*, 1st ed., Springer US, USA, 1-35, 2011.
- [63] Lealos G., Caravelas P., A Hybrid Approach for Movie Recommendation, *Multimedia Tools and Applications*, 2008, **36**(1), 55-70.
- [64] Dwivedi P., Chheda N., A Hybrid Restaurant Recommender, *International Journal of Computer Applications*, 2012, **55**(16), 20-25.
- [65] Christakou C., Vrettos S., Stafylopatis A., A Hybrid Movie Recommender System Based on Neural Networks, *International Journal on Artificial Intelligence Tools*, 2007, **16**(5), 771-792.
- [66] Alsalama A. M. K., A Hybrid Recommendation System Based on Association Rules, *International Journal of Computer and Information Engineering*, 2015, **9**(1), 51-62.
- [67] Kuzelewska U., Clustering Algorithms in Hybrid Recommender System on MovieLens Data, *Studies in Logic, Grammar and Rhetoric*, 2014, **37**(1), 125-139.
- [68] Mojtaba B., Reza M., A New Hybrid Recommender System Using Dynamic Fuzzy Clustering, *3rd International Conference on Information and Knowledge Technology*, Mashad, İnan, 27-29 November 2007.
- [69] Soni K., Goyal R., Vadera B., More S., A Three Way Hybrid Movie Recommendation System, *International Journal of Computer Applications*, 2017, **160**(9), 29-32.
- [70] Ali Z., Khusro S., Ullah I., A Hybrid Book Recommender System Based on Table of Contents and Association Rule Mining, *10th International Conference on Informatics and Systems*, Giza, Egypt, 9-11 May 2016.
- [71] Campos L. M., Fernandez-Luna J. M., Huete J.F., Ruade-Morales M. A., Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks, *International Journal of Approximate Reasoning*, 2010, **51**(7), 785-799.

- [72] Resnick P., Iacovou N., Suchak M., Bergstrom, Riedl J., GroupLens: An Open Architecture for Collaborative Filtering of Networks, *1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, USA, 22-26 October 1994.
- [73] Herlocker J. L., Konstan J. A., Borchers A., Riedl J., An Algorithmic Framework for Performing Collaborative Filtering, *22nd Annual International ACM SIGIR Conference on Research And Development in Information Retrieval*, Berkeley, USA, 15-19 August 1999.
- [74] Ekstrand M., Similarity Functions for User – User Collaborative Filtering, GroupLens, <https://grouplens.org/blog/similarity-functions-for-user-user-collaborative-filtering/> (Ziyaret tarihi : 7 Şubat 2018).
- [75] Jamali M., Ester M., A Matrix Factorization Technique with trust Propagation for Recommendation in Social Networks, *The 4th ACM Conference on Recommender Systems*, Barselona, Spain, 26 – 30 September 2010.
- [76] Ma H., Yang H., Lyu M. R., King I., SoRec: Social Recoomendation Using Probabilistic Matrix Factorization, *The 17th ACM Conference on Information and Knowledge Management*, California, USA, 26 – 30 October 2008.
- [77] Koren Y., Bell R., Volinsky C., Matrix Factorization Techniques for Recommender Systems, *Computer*, 2009, **42**(8), 30 – 37.
- [78] Kawale J., Bui H. H., Kveton B., Thanh L. T., Chawla S., Efficient Thompson Sampling for Online Matrix Factorization Recommendation, *Advances in Neural Information Processing Systems*, 2015, 1297 – 1305.
- [79] Purushotham S., Liu Y., Kuo C. C. J., Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems, 2012, arXiv: 1206.4684.
- [80] Takacs G., Pilsz I., Nemeth B., Tik D., Matrix Factorization and Neighbor Based Algorithms for The Netflix Prize Problem, *The 2008 ACM Conference on Recommender Systems*, Lausanne, Switzerland, 23 - 25 October 2008.
- [81] Baker K., Singular Value Decomposition Tutorial, https://davetang.org/file/Singular_Value_Decomposition_Tutorial.pdf (Ziyaret Tarihi: 1 Haziran 2018).
- [82] Chen B. C., Latent Factor Models for Web Recommender Systems, Intelligent Data Exploration and Analysis Laboratory University of Texas, <http://www.ideal.ece.utexas.edu/seminar/LatentFactorModels.pdf> (Ziyaret Tarihi: 20 Mart 2018).
- [83] Yeung A. A., Matrix Factorization: A Simple Tutorial and Implementation in Phyton, Quuxlabs, <http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/> (Ziyaret tarihi: 8 Şubat 2018).

- [84] Ivarsson J., Lindgren M., Movie Recommendations Using Matrix Factorization, KTH Royal Institute of Technology, School of Computer Science and Communication, 2016.
- [85] Argerich L., Why does latent factor model (matrix factorization) better than neighborhood model in collaborative filtering?, Quora, <https://www.quora.com/Why-does-latent-factor-model-matrix-factorization-better-than-neighborhood-model-in-collaborative-filtering> (Ziyaret Tarihi: 16 Nisan 2018).
- [86] Anonim, MovieLens Latest Dataset, GroupLens <https://grouplens.org/datasets/movielens/latest/> (Ziyaret Tarihi: 26 Şubat 2018).
- [87] Foti R., Exploring Recommenders Systems, Romano Foti's Blog, http://blog.romanofoti.com/exploring_recommenders_movielens_dataset/ (Ziyaret Tarihi: 1 Mart 2018).
- [88] Son N. T., Dat D. H., Trung N. Q., Anh B. N., Combination of Dimensionality Reduction and User Clustering for Collaborative Filtering, 2017 International Conference on Computer Science and Artificial Intelligence, Jakarta, Indonesia, 5 - 7 December 2017.
- [89] Tseng G., Clustering and Collaborative Filtering – Implementing Neural Networks, Medium, <https://medium.com/@gabrielteng/clustering-and-collaborative-filtering-implementing-neural-networks-bccf2f9ff988> (Ziyaret Tarihi: 1 Mart 2018).
- [90] Grover P., Various Implementations of Collaborative Filtering, Towards Data Science, <https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0> (Ziyaret Tarihi: 1 Mart 2018).
- [91] Boström P., Filipsson M., Comparison of User Based and Item Based Collaborative Filtering Recommendation Services, <http://kth.diva-portal.org/smash/get/diva2:1111865/FULLTEXT01.pdf> (Ziyaret Tarihi: 28 Şubat 2018).
- [92] Anonim, <https://studio.azureml.net> (Ziyaret tarihi: 25 Kasım 2017).
- [93] Anonim, <https://azure.microsoft.com/tr-tr/offers/ms-azr-0044p/> (Ziyaret Tarihi: 15 Ocak 2018).
- [94] Anonim, https://www.itcentralstation.com/categories/data-science-platforms#top_rated.aspx (Ziyaret tarihi: 14 Mayıs 2018).
- [95] Parim G. Z., Duru N., Gezi Öneri Sistemine Kullanıcı Özelliklerinin Etkisi, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2018, 6(1), 152 – 161.
- [96] Wang Y., Wang L., Li, Y., He D., Chen W., Liu T., Y., A Theoretical Analysis of NDCG Ranking Measures, *26th Annual Conference on Learning Theory*, Princeton, USA, 12 - 14 June 2013.

- [97] Motl J., Normalized Discounted Cumulative Gain, MathWorks, <https://www.mathworks.com/matlabcentral/fileexchange/65570-normalized-discounted-cumulative-gain--ndcg-> (Ziyaret tarihi: 16 Şubat 2018).
- [98] Mhaskar V., Measuring Search Relevance Using NDCG, The Digital Group Blog, <http://blog.thedigitalgroup.com/vijaym/measuring-search-relevance-using-ndcg/> (Ziyaret tarihi: 17 Şubat 2018).





EK – A1

Tablo A1. Azure Machine Learning ile elde edilen öneriler

User	Item 1	Item 2	Item 3	Item 4	Item 5
132	Çırağan Sarayı	Masumiyet Müzesi	Boğaz Turu	Pera Müzesi	Otağtepe
93	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Pera Müzesi	Çırağan Sarayı
118	Masumiyet Müzesi	Topkapı Sarayı	Galata Kulesi	Ayasofya Müzesi	Sultanahmet Cami
205	Çırağan Sarayı	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Masumiyet Müzesi
4	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Ayasofya Müzesi	Sultanahmet Cami
55	Galata Kulesi	Topkapı Sarayı	Masumiyet Müzesi	Ayasofya Müzesi	Sultanahmet Cami
133	Masumiyet Müzesi	Fethi Paşa Korusu	Mihrabat Korusu	Otağtepe	Pera Müzesi
92	Galata Kulesi	Masumiyet Müzesi	Pera Müzesi	Otağtepe	Topkapı Sarayı
78	Masumiyet Müzesi	Galata Kulesi	Topkapı Sarayı	Ayasofya Müzesi	Çırağan Sarayı
99	Çırağan Sarayı	Boğaz Turu	Topkapı Sarayı	Ayasofya Müzesi	Sultanahmet Cami
95	Masumiyet Müzesi	Pera Müzesi	Otağtepe	Fethi Paşa Korusu	Cemile Sultan Korusu
183	Masumiyet Müzesi	Galata Kulesi	Topkapı Sarayı	Ayasofya Müzesi	Sultanahmet Cami
39	Masumiyet Müzesi	Galata Kulesi	Pera Müzesi	Otağtepe	Topkapı Sarayı
27	Masumiyet Müzesi	Çırağan Sarayı	Otağtepe	Fethi Paşa Korusu	Pera Müzesi
198	Masumiyet Müzesi	Mısır Çarşısı	Otağtepe	Pera Müzesi	Galata Kulesi
114	Boğaz Turu	Ayasofya Müzesi	Sultanahmet Cami	Topkapı Sarayı	Eyup Sultan Cami
167	Boğaz Turu	Ayasofya Müzesi	Sultanahmet Cami	İstanbul Akvaryum	Topkapı Sarayı
70	Rahmi Koç Müzesi	Yıldız Parkı	Kınalı Ada	Burgazada	Atatürk Arboretumu
171	Çırağan Sarayı	Masumiyet Müzesi	Otağtepe	Pera Müzesi	Cemile Sultan Korusu
87	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Pera Müzesi	Otağtepe
50	Çırağan Sarayı	Otağtepe	Masumiyet Müzesi	Pera Müzesi	Oyuncak Müzesi
160	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Ayasofya Müzesi	Boğaz Turu
42	Masumiyet Müzesi	Galata Kulesi	Topkapı Sarayı	Ayasofya Müzesi	Sultanahmet Cami
155	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
121	Çırağan Sarayı	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Masumiyet Müzesi

Tablo A1 (Devam). Azure Machine Learning ile elde edilen öneriler

138	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
38	Masumiyet Müzesi	Galata Kulesi	Topkapı Sarayı	Pera Müzesi	Ayasofya Müzesi
165	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
91	Boğaz Turu	Ayasofya Müzesi	Sultanahmet Cami	Topkapı Sarayı	İstanbul Akvaryum
37	Galata Kulesi	Masumiyet Müzesi	Pera Müzesi	Otağtepe	Topkapı Sarayı
72	Masumiyet Müzesi	Pera Müzesi	Otağtepe	Galata Kulesi	Oyuncak Müzesi
60	Çırağan Sarayı	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Cemile Sultan Korusu
137	Masumiyet Müzesi	Mısır Çarşısı	Pera Müzesi	Galata Kulesi	Otağtepe
127	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi	Boğaz Turu	Topkapı Sarayı
48	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Pera Müzesi	Otağtepe
84	Boğaz Turu	Ayasofya Müzesi	Sultanahmet Cami	Topkapı Sarayı	Eyüp Sultan Cami
19	Topkapı Sarayı	Boğaz Turu	Sultanahmet Cami	Ayasofya Müzesi	Eyüp Sultan Cami
1	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi	Otağtepe	Pera Müzesi
143	Galata Kulesi	Masumiyet Müzesi	Topkapı Sarayı	Ayasofya Müzesi	Sultanahmet Cami
211	Boğaz Turu	Ayasofya Müzesi	İstanbul Akvaryum	Sultanahmet Cami	Topkapı Sarayı
180	Otağtepe	Masumiyet Müzesi	Çırağan Sarayı	Pera Müzesi	Oyuncak Müzesi
176	Masumiyet Müzesi	Fethi Pasa Korusu	Mihrabat Korusu	Otağtepe	Pera Müzesi
65	Masumiyet Müzesi	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Çırağan Sarayı
144	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
7	Çırağan Sarayı	Masumiyet Müzesi	Pera Müzesi	Oyuncak Müzesi	Otağtepe
45	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
159	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
66	Masumiyet Müzesi	Çırağan Sarayı	Pera Müzesi	Otağtepe	Oyuncak Müzesi
10	Çırağan Sarayı	Cemile Sultan Korusu	Oyuncak Müzesi	Otağtepe	Cağaloğlu Hamamı
184	Masumiyet Müzesi	Otağtepe	Boğaz Turu	Pera Müzesi	Cemile Sultan Korusu
185	Mihrabat Korusu	Masumiyet Müzesi	Boğaz Turu	Ayasofya Müzesi	Galata Kulesi
58	Boğaz Turu	Sultanahmet Cami	Ayasofya Müzesi	Topkapı Sarayı	Eyüp Sultan Cami
179	Masumiyet Müzesi	Pera Müzesi	Otağtepe	Mısır Çarşısı	Oyuncak Müzesi

EK – A2

Tablo A2. Sadece oy verilmeyen nesnelere ile elde edilen öneriler

User	1. Öneri	2. Öneri	3. Öneri	4. Öneri	5. Öneri
132	Çırağan Sarayı	Masumiyet Müzesi	Pera Müzesi	Otağtepe	
93					
118					
205	Çırağan Sarayı	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Masumiyet Müzesi
4					
55					
133	Masumiyet Müzesi	Fethi Pasa Korusu	Mihrabat Korusu	Otağtepe	Pera Müzesi
92					
78					
99	Çırağan Sarayı				
95	Pera Müzesi	Otağtepe	Fethi Pasa Korusu	Cemile Sultan Korusu	
183					
39					
27	Masumiyet Müzesi	Çırağan Sarayı	Otağtepe	Fethi Pasa Korusu	Pera Müzesi
198	Masumiyet Müzesi	Pera Müzesi			
114					
167	İstanbul Akvaryum				
70	Rahmi Koç Müzesi	Yıldız Parkı	Kınalı Ada	Burgazada	Atatürk Arboretumu
171	Çırağan Sarayı	Masumiyet Müzesi	Otağtepe	Cemile Sultan Korusu	
87					
50	Çırağan Sarayı	Otağtepe	Masumiyet Müzesi	Pera Müzesi	Oyuncak Müzesi
160					
42					
155	Pera Müzesi	Masumiyet Müzesi			
121	Otağtepe	Pera Müzesi	Oyuncak Müzesi	Masumiyet Müzesi	
138					
38					
165	Çırağan Sarayı	Pera Müzesi			
91	İstanbul Akvaryum				
37					
72					
60	Otağtepe	Pera Müzesi	Cemile Sultan Korusu		
137	Masumiyet Müzesi	Pera Müzesi	Galata Kulesi		
127	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi		
127	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi		
48					
84					
19	Topkapı Sarayı	Boğaz Turu	Sultanahmet Cami	Ayasofya Müzesi	
1	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi	Otağtepe	Pera Müzesi

Tablo A2 (Devam). Sadece oy verilmeyen nesnelere ile elde edilen öneriler

48					
84					
19	Topkapı Sarayı	Boğaz Turu	Sultanahmet Cami	Ayasofya Müzesi	
1	Fethi Pasa Korusu	Taksim Meydani	Masumiyet Müzesi	Otağtepe	Pera Müzesi
143					
211	İstanbul Akvaryum				
180	Masumiyet Müzesi	Çırağan Sarayı	Pera Müzesi	Oyuncak Müzesi	
176	Masumiyet Müzesi	Fethi Pasa Korusu	Mihrabat Korusu		
65					
144					
7	Masumiyet Müzesi	Pera Müzesi	Oyuncak Müzesi		
45	Çırağan Sarayı	Otağtepe	Pera Müzesi	Masumiyet Müzesi	Oyuncak Müzesi
159					
66					
10	Çırağan Sarayı	Cemile Sultan Korusu	Oyuncak Müzesi	Cağaloğlu Hamamı	
184	Masumiyet Müzesi	Otağtepe	Pera Müzesi		
185	Mihrabat Korusu	Masumiyet Müzesi	Galata Kulesi		
58					
179	Masumiyet Müzesi	Otağtepe	Oyuncak Müzesi		

EK – B

Tablo B1. Matris bölündüğünde elde edilen öneriler

User	1. Öneri	2. Öneri	3. Öneri	4. Öneri	5. Öneri
132	Dolmabahçe Sarayı	Otağtepe	Masumiyet Müzesi	Pera Müzesi	Oyuncak Müzesi
93					
118					
205	Otağtepe	Miniatürk	Çamlıca Tepesi	Heybeli Ada	Rumeli Hisarı
4					
55					
133	Kınalı Ada	Mihrabat Korusu	Fethi Paşa Korusu	Topkapı Sarayı	Yıldız Parkı
92					
78					
99	Kınalı Ada	Atatürk Arboretumu	Oyuncak Müzesi	Heybeli Ada	Yedikule Surları
95	Cemile Sultan Korusu	Fethi Paşa Korusu	Cağaloğlu Hamamı	Rumeli Hisarı	Emirgan Korusu
183	Bebek Sahili				
39					
27	Süleymaniye Cami	Çırağan Sarayı	Eyüp Sultan Cami	Cağaloğlu Hamamı	Rumeli Hisarı
198	Burgazada	Cağaloğlu Hamamı			
114					
167	Bebek Sahili	Atatürk Arboretumu	Masumiyet Müzesi	Mihrabat Korusu	Aşıyan Müzesi
70	Miniatürk	Çamlıca Tepesi	Fethi Paşa Korusu	Otağtepe	Çırağan Sarayı
171	Cağaloğlu Hamamı	Yedikule Surları	Vialand	İstanbul Arkeoloji Müzesi	İstanbul Akvaryum
87					
50	Kız Kulesi	Topkapı Sarayı	Galata Kulesi	Otağtepe	Mihrimah Sultan Cami
160					
42					
155	Dolmabahçe Sarayı	İstanbul Arkeoloji Müzesi	Cağaloğlu Hamamı	Pierre Loti Tepesi	Atatürk Arboretumu
121	İstanbul Akvaryum	Ayasofya Müzesi	Otağtepe	Dolmabahçe Sarayı	Mihrabat Korusu
138					
38					
165	Pierre Loti Tepesi	Cağaloğlu Hamamı	Heybeli Ada	Yedikule Surları	Dolmabahçe Sarayı

Tablo B1 (Devam). Matris bölündüğünde elde edilen öneriler

91	Miniatürk	Kınalı Ada	Burgazada	Çırağan Sarayı	Vialand
37					
72					
60	Kız Kulesi	Büyükada	Masumiyet Müzesi	Rumeli Hisarı	Dolmabahçe Sarayı
137	Aşiyân Müzesi	Rahmi Koç Müzesi	Kuzguncuk Evleri	Bebek Sahili	Ortaköy Cami
127	Mihrimah Sultan Cami	Oyuncak Müzesi	Cağaloğlu Hamamı	Kuzguncuk Evleri	Çırağan Sarayı
48					
84					
19	Kınalı Ada	Ortaköy Cami	Anadolu Hisarı	Ayasofya Müzesi	Boğaz Turu
1	Atatürk Arboretumu	Galata Kulesi	Rumeli Hisarı	Anadolu Hisarı	Kınalı Ada
143					
211	Çırağan Sarayı	Burgazada	Fethi Paşa Korusu	Mihrabat Korusu	Kınalı Ada
180	Eyüp Sultan Cami	Yerebatan Sarnıcı	Galata Kulesi	Topkapı Sarayı	Dolmabahçe Sarayı
176	Kuzguncuk Evleri	Yedikule Surları	Aşiyân Müzesi	Mihrabat Korusu	Fethi Paşa Korusu
65	Vialand				
144					
7	İstanbul Akvaryum	Oyuncak Müzesi	Pierre Loti Tepesi	Rahmi Koç Müzesi	Pera Müzesi
45	Yerebatan Sarnıcı	Çamlıca Tepesi	Heybeli Ada	Yıldız Parkı	Mihrimah Sultan Cami
159					
66	Kuzguncuk Evleri				
10	Atatürk Arboretumu	Rumeli Hisarı	Cemile Sultan Korusu	Yerebatan Sarnıcı	Pierre Loti Tepesi
184	Panorama 1453	Miniatürk	Vialand	Fethi Paşa Korusu	Dolmabahçe Sarayı
185	İstanbul Akvaryum	Mihrabat Korusu	Emirgan Korusu	Cemile Sultan Korusu	Yedikule Surları
58					
179	Oyuncak Müzesi	İstanbul Arkeoloji Müzesi	Panorama 1453	Eyüp Sultan Cami	Masumiyet Müzesi

KİŞİSEL YAYINLAR VE ESERLER

[1]. **Parim G. Z.**, Duru N., Gezi Öneri Sistemine Kullanıcı Özelliklerinin Etkisi, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2018, **6**(1), 152 – 161.



ÖZGEÇMİŞ

Gizem Zeynep PARİM, 1992 yılında İstanbul'da doğdu. İlköğretim ve lise eğitimini İstanbul'da tamamladı. 2010 yılında girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2014 yılında mezun oldu. 2014 – 2015 yılı bahar yarıyılından itibaren Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine devam etmektedir.

