



**ÖĞRENCİ KAYIT YENİLEME
İŞLEMLERİNDE
ÖNERİ SİSTEMLERİ UYGULAMASI
Yüksek Lisans Tezi**

İsmail Burak YAVRU

Eskişehir 2019

**ÖĞRENCİ KAYIT YENİLEME İŞLEMLERİNDE ÖNERİ SİSTEMLERİ
UYGULAMASI**

İsmail Burak YAVRU

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Alper Bilge

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Aralık 2019

Bu tez çalışması BAP Komisyonu tarafından kabul edilen 1505F414 no.lu proje kapsamında desteklenmiştir.

JÜRİ VE ENSTİTÜ ONAYI

İsmail Burak YAVRU'nun ÖĞRENCİ KAYIT YENİLEME İŞLEMLERİNDE ÖNERİ SİSTEMLERİ UYGULAMASI başlıklı tezi 09/12/2019 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Eskişehir Teknik Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Bilgisayar Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

<u>Jüri Üyeleri</u>	<u>Unvanı Adı Soyadı</u>	<u>İmza</u>
Üye (Tez Danışmanı)	: Doç. Dr. Alper BİLGE	
Üye	: Doç. Dr. Cihan KALELİ	
Üye	: Dr. Öğr. Üyesi Melih ENGİN	

Prof. Dr. Murat TANIŞLI

Lisansüstü Eğitim Enstitüsü Müdürü

ÖZET
ÖĞRENCİ KAYIT YENİLEME İŞLEMLERİNDE ÖNERİ SİSTEMLERİ
UYGULAMASI

İsmail Burak YAVRU

Bilgisayar Mühendisliği Anabilim Dalı
Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Aralık 2019

Danışman: Doç. Dr. Alper BİLGE

Ders seçim aşaması üniversite öğrencileri için zorluklar teşkil etmektedir. Bu aşamadaki öğrenciler için ders tavsiyesi sunmak üzere akademik danışmanlık yapan bir öneri sistemi bu sorunların üstesinden gelebilir. Geleneksel yöntemlerle sözlü tavsiye olarak ya da ders kataloglarındaki bilgilere bakarak öğrencilerin kendi akademik profillerine uyan dersleri bulmaları oldukça zordur. Bunun yanı sıra bir akademisyenden akademik danışmanlık desteği alınsa bile ilgili akademisyenin her öğrencinin akademik profilini bilmesi mümkün değildir. Ayrıca akademisyenlerin bölümlerindeki bütün derslerin niteliklerini bilmeleri de mümkün değildir. Bu çalışma ile öğrencilerin akademik başarılarının yükseltilmesi ve öğrencilerin kendi profillerine uyan dersleri seçebilmeleri için bir ders öneri sistemi geliştirilmiştir. Geliştirilen öneri sistemi, bir kullanıcı tabanlı ortak filtreleme öneri sistemidir. Bu sistem öğrencilerin kendisiyle benzer akademik özellikteki diğer öğrencilerin önceden aldıkları dersleri inceleyerek, kullanıcılarına ders öneri listesi ve harf notu tahminleri sunar. Sistem kullanıcılarına, kendisiyle akademik benzerlik gösteren diğer öğrencilerin alıp başarılı olduğu dersler üzerinden almasını tavsiye ettiği dersleri sunar. Bu öneri listesini ve harf notu tahminlerini üretirken kullanıcılarından hiçbir veri talep etmez. Öneri üretilecek olan kullanıcının daha önceden almış olduğu derslerdeki performansını aynı bölümdeki mezun ve üst dönem öğrencilerinin geçmişteki performansları ile benzerlik kurarak önerilerini oluşturur. Böylelikle derslerle ilgili akademik özelliklerin tanımlanmasına ve içeriklerinin bilinmesine gerek duymaz. Sistem deneysel yöntemlerle geliştirilmiştir. Deneylerden elde edilen sonuçlar sistemin başarılı bir şekilde çalıştığını göstermektedir.

Anahtar Sözcükler: Öneri sistemleri, Kullanıcı tabanlı ortak filtreleme, Ders öneri sistemi

ABSTRACT

A RECOMMENDER SYSTEMS APPLICATION FOR STUDENTS REGISTRATION RENEWAL

İsmail Burak YAVRU

Department of Computer Engineering

Eskişehir Technical University, Institute of Graduate Programs, December 2019

Supervisor: Assoc. Prof. Dr. Alper BİLGE

It could be very challenging to choose a course at the registration period for university students. A system could overcome this problem by giving academic advice. System should recommend courses to students. It is very difficult for students to find courses that match their academic profile with traditional methods like word of mouth and information in the course catalogs. In addition, even if academic advice is provided from an academician, it is not possible for the relevant academician to know the academic profile of each student. Furthermore, it is not possible for academics to know the qualifications of all courses in their departments. In this study, a course recommender system has been developed in order to increase the academic success of the students and enable them to choose the courses that fit their own profiles. The proposed recommender system is a user-based collaborative filtering recommender system. This system examines the courses previously taken by other students who has similar academic profiles with active student and provides users recommended courses list and letter grade estimates. Proposed recommender system does not require any explicit data from its users when generating this list of recommendations and letter note estimates. System creates recommendations by establishing a similarity between the performance of the active user and the upper semester students/graduates in the same department. Thus, it does not need to define the academic features and content of the courses. The system was developed by experimental methods. The results obtained from the experiments show that the system works successfully.

Keywords: Recommender systems, User based collaborative filtering, Course recommendation system

TEŞEKKÜR

Tez çalışmam boyunca değerli bilgi ve birikimlerini benimle paylaşan, çalışmam için beni motive eden ve tecrübeleriyle hem akademik hem de kişisel olarak bana her zaman destek olan saygıdeğer tez danışmanım Doç. Dr. Alper Bilge'ye sonsuz teşekkür ve saygılarımı sunarım.

Yüksek lisans çalışmam süresince desteklerini benden esirgemeyen çok değerli mesai arkadaşlarım Anadolu Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi Örgün Öğrenci İşleri Yazılım Ekibi'ne teşekkürlerimi sunarım.

Tez kapsamında gerçekleştirdiğim çalışmaya Bilimsel Araştırma Projeleri desteği ile maddi destek sağlayan Anadolu Üniversitesine teşekkürlerimi sunarım.

Son olarak, hayatımın her anında yanımda olan, beni hem akademik hem de kişisel olarak bugünlere getiren, desteklerini her an hissettiğim çok değerli babam Hasan Yavru, annem Pembe Yavru ve kardeşim Hazar Berkay Yavru'ya ve son 11 yılda başıma gelen en güzel şey, tüm iyi zamanlarda yanımda olduğu gibi geçirdiğim en kötü zamanlarda da bir gün bile sevgisini ve desteğini eksik hissetmediğim değerli eşim Özge Güneş Yavru'ya sonsuz sevgi ve destekleri için şükranlarımı sunarım. Ailem olmadan bu çalışma da olmazdı.

İsmail Burak YAVRU

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

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

İsmail Burak YAVRU

İÇİNDEKİLER

Sayfa

ÖĞRENCİ KAYIT YENİLEME İŞLEMLERİNDE ÖNERİ SİSTEMLERİ UYGULAMASI.....	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜR	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vi
İÇİNDEKİLER	vii
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
KISALTMALAR DİZİNİ	xi
1. GİRİŞ	1
1.1. Öneri Sistemleri Algoritmaları.....	3
1.1.1. Ortak filtreleme	3
1.1.2. İçerik tabanlı filtreleme	7
1.1.3. Bilgi Tabanlı Filtreleme	9
1.1.4. Melez Algoritmalar	11
1.2. Ortak Filtreleme Algoritma Yaklaşımları.....	12
1.2.1 Hafıza Tabanlı Ortak Filtreleme Algoritması.....	12
1.2.1.1. Kullanıcı tabanlı ortak filtreleme	14
1.2.1.2. Ürün tabanlı ortak filtreleme.....	14
1.2.2. Model tabanlı ortak filtreleme algoritması.....	14
1.2.3. Karma algoritmalar	15
1.3. Amaç, Konu, Kapsam ve Katkılar	16
2. İLGİLİ ÇALIŞMALAR.....	18
3. ORTAK FİLTRELEME TABANLI DERS ÖNERİ SİSTEMİ	21
4. KULLANILAN TEKNOLOJİLER	24

4.1. Ön Uç Teknolojileri	24
4.1.1. HTML5.....	25
4.1.2. Basamaklı Stil Şablonları ve LESS.....	25
4.1.3. JavaScript ve AngularJS	26
4.2. Arka Uç Teknolojileri.....	27
4.2.1. Java ve Spring Web MVC	27
4.2.2. Microsoft SQL Server	27
5. DENEYSEL ÇALIŞMALAR.....	29
5.1. Veri Kümesi	29
5.2. Değerlendirme Ölçütleri.....	30
5.3. Metodoloji.....	31
5.3.1. Komşuluk seçme ve öneri üretme yöntemi	31
5.3.1.1. K en yakın komşu yöntemi.....	32
5.3.1.2. Benzerlik eşiği yöntemi.....	32
5.3.1.3. Öneri üretimi.....	32
5.4. Deney Sonuçları ve Analizi.....	33
5.4.1. Ortak ders eşiği deneyleri.....	33
5.4.2. Komşuluk belirleme yöntemi deneyleri.....	35
6. SONUÇLAR VE DEĞERLENDİRME	38
KAYNAKÇA.....	39
ÖZGEÇMİŞ	44

TABLolar DİZİNİ

	<u>Sayfa</u>
Tablo 5.1. ESTÜBM Örnek Öğrenci Transkript Verisi.....	29
Tablo 5.2. Harf notları ve karşılık gelen derecelendirme puanları.....	30
Tablo 5.3. Veri Seti ile İlgili Bilgiler.....	30
Tablo 5.4. K komşuluk değerleri için deney sonuçları(Ortak ders sayısı eşiği 5).....	34
Tablo 5.5. Benzerlik Eşiği için deney sonuçları(Ortak ders eşiği 5).....	34
Tablo 5.6. K komşuluk değerleri için deney sonuçları(Ortak ders sayısı eşiği 3).....	34
Tablo 5.7. Benzerlik eşiği için deney sonuçları(Ortak ders sayısı alt eşiği 3'tür).....	35
Tablo 5.8. K Komşuluk Değerleri için deney sonuçları.....	36
Tablo 5.9. Benzerlik Eşiği Deney Sonuçları.....	36

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 1.1. Ortak Filtreleme Temel Çalışma Prensibine bir Film Öneri Sistemi Örneği.	3
Şekil 1.2. Örnek Aktivite Öneri Sistemi.....	5
Şekil 1.3. İçerik Tabanlı Filtreleme Film Öneri Sistemi Örneği.....	7
Şekil 1.4. İTF Öneri Motoru Çalışma Prensibi.....	8
Şekil 1.5. Kısıtlama Tabanlı Bilgi Filtreme Örnek Ara Yüzü.....	10
Şekil 1.6. Vaka Tabanlı Bilgi Filtreleme Örnek Ara Yüzü.....	11
Şekil 1.7. Kullanıcı Tabanlı ve Ürün Tabanlı Öneri Sistemlerinin Çalışma Mekanizması.....	13
Şekil 3.1. Ders Öneri Listesi Ekran Görüntüsü.....	23
Şekil 4.1. Ön yüz ve arka yüz katmanları ve çalışma prensibi.....	24
Şekil 5.1. Benzerlik Eşiğinin Değişimine Karşılık, <i>OMH</i> 'nin Değişimi.....	37

KISALTMALAR DİZİNİ

AIS	: Yapay Bağışıklık Sistemi
AJS	: AngularJS
ESTÜOBS	: Eskişehir Teknik Üniversitesi Öğrenci Bilişim Sistemi
ESTÜ	: Eskişehir Teknik Üniversitesi
ESTÜBM	: Eskişehir Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
BAUM	: Bilgisayar Araştırma ve Uygulama Merkezi
BTF	: Bilgi Tabanlı Filtreleme
CRS	: Ders Öneri Sistemi
CSS	: Basamaklı Stil Şablonları
HTML5	: Hiper Metin İşaret Dili versiyon 5
HTOF	: Hafıza Tabanlı Ortak Filtreleme
İTF	: İçerik Tabanlı Filtreleme
JS	: JavaScript
JVM	: Java Sanal Makinesi
kNN	: K En Yakın Komşu
KTOF	: Kullanıcı Tabanlı Ortak Filtreleme
LESS	: Leaner Stil Şablonları
MSSQL	: Microsoft SQL Server
MTOF	: Model Tabanlı Ortak Filtreleme
OF	: Ortak Filtreleme
OFTDÖS	: Ortak Filtreleme Tabanlı Ders Öneri Sistemi
OMH	: Ortalama Mutlak Hata
ÜTOF	: Ürün Tabanlı Ortak Filtreleme
WWW	: World Wide Web Consortium

1. GİRİŞ

Karar vermek birçok insan için zorlu bir süreç oluşturmaktadır. Öyle ki bir fırına gidildiğinde hangi ekmeğin alınacağından, hangi kitabın okunacağına, hangi film izleneceğinden tatil için hangi yöreye gidileceğine kadar hayatımızın hemen her kesiminde bir karar verme durumu söz konusudur. Bu tip kararlar verilirken seçeneklerin fazlalığı süreci daha da zorlaştırmaktadır. Kişilerin kendilerine uygun olan seçimler yapmaları zorlaşmaktadır. Geleneksel yöntemle insanlar, seçim yapacakları konu üzerinde sözlü öneriler alarak ya da kitap, film incelemeleri gibi yapılan yorumlar ve eleştiriler üzerinden kararlarını oluştururlar [1].

Günümüzde internet günlük hayatın büyük bir kısmını kaplamaktadır. İnternet sayesinde birçok işlem hızlı ve mobil bir şekilde çevrimiçi sistemler yardımıyla gerçekleştirilebilmektedir. Bu sistemler hayatı kolaylaştırdığı kadar diğer bir açıdan da zorlaştırmaktadır. Gün geçtikçe artan bu sistemler sonucu internet üzerinde bulunan veri yoğunluğu da artmaktadır. Bu durum kullanıcılarının internet üzerinde ilgi alanlarına uygun verilere erişmelerini zorlaştırmaktadır. Örneğin yabancı dizi sektörünün daha çok ilgi çekmeye başlamasıyla birlikte yapımcılar birçok diziye Türk kullanıcıları için Türkçe dublaj ve altyazı eklemektedir. İzlenebilecek dizi sayısının artmasıyla kullanıcılar beğenecekleri diziyi seçmekte zorlanmaya başlamaktadırlar. Bunun sonucunda olumsuz bir durum ortaya çıkmaktadır.

Seçilen hizmet ya da ürünler kullanıcının beğenisini kazanmalı ve ihtiyacını karşılamalıdır. Kullanıcılar aradıkları ürün veya hizmete hızlı ve basit bir şekilde ulaşabilmelidir. Veri yoğunluğunun fazla olması kullanıcıların ilgili ürün ve hizmetlere ulaşmasını zor hale getirmektedir. Seçeneklerin fazla olması içinden seçilebilecek olanların ayıklanmasını kullanıcılar için zaman alıcı ve meşakkatli bir hale getirmektedir. Bu karar verme zorluğu teknolojik gelişmelerin artması ve yaygınlaşması sebebiyle internet üzerinde de kişilerin hayatına etki etmeye başlamıştır. Her geçen gün internetteki veri miktarı artmaktadır. Bu sebeple internet kullanıcılarının ilgi duydukları içeriklere erişmesi oldukça güçleşmiştir. İnsanlar beğenebilecekleri içeriklere hızlı ve kolayca ulaşmak istemektedirler [2]. Öneri sistemleri, bu problemlere çözüm olması için geliştirilmiştir. Öneri Sistemleri kullanıcılarına, geçmişte yaptıkları seçimlerden yola çıkarak kişisel öneriler sunar, böylece onların ilgi alanlarına giren içeriklere kolayca erişmesini sağlar. Diğer bir deyişle Öneri Sistemleri sözlü öneri gibi geleneksel süreçleri

sayısal ortamda otomatik hale getirir [1]. Bu sistemler kullanıcıların kişiselleştirilmiş öneri almalarını ve internet üzerinde işlem yaparken işlem üzerinde doğru karar vermelerini sağlar. Öneri sistemleri benzer zevkleri olan kullanıcı topluluklarının seçimlerini içlerindeki bir tanesine öneri sunmak için kullanır [3]. Öneri sistemleri, kullanıcılarına beğenebilecekleri ürün ya da hizmetler hakkında öneri ve tahminlerde bulunarak bilgi vermektedir [4, 5]. Bu sistemler kullanıcılarına fazladan bir emek sarf etmelerine gerek kalmadan, onların davranışları ve ayırt edici özellikleri üzerinde çalışarak, ilgilenebilecekleri ürün ve hizmetler ile ilgili verileri kullanarak otomatik olarak öneri veya tahmin üretirler. Böylece kullanıcıların karar verme süreçlerine destek olurlar [6].

Günümüzde öneri sistemleri birçok çevrimiçi sistemlerle bütünleşmiş bir biçimde karşımıza çıkmaktadır. Birçok popüler e-ticaret, sosyal medya, içerik üretici vb. sistemlerde etkin bir biçimde kullanıcılarına zevklerine uygun içerikleri sunmaktadır. Kitap, müzik, oyun, dizi, film, sosyal medya içerikleri, haber, restoran, konaklama, uçak bileti ve benzeri birçok alanda öneri sistemleri aktif bir şekilde kullanılmaktadır. SkyScanner (Uçak Bileti), NetFlix (Dizi ve Film), Spotify (Müzik), Apple Müzik (Müzik), PlayStation Store (Oyun), AirBnb (Konaklama), Amazon (Kitap ve e-ticaret), Twitter (Sosyal Medya), Bundle (Haber) öneri sistemlerini etkin bir biçimde kullanan uygulamalardan bazılarıdır [7, 8, 9].

Öneri sistemlerinin amacı kullanıcılara zevklerine uygun ürünler, hizmetler ya da içerikler sunmaktır. Bu amaç sonucunda öneri sistemleri kullanıldığında birçok fayda elde edilmiş olur. Öneri sistemi kullanıcılarına doğru öneriler verdiğinde kullanıcılar arasında sistemin popülerliği artar. Böylece sistem sahibinin daha fazla kullanıcı/müşterisi olur [10]. Kullanıcılar aradıkları ürünlere daha çabuk ulaşırlar. Daha az çaba sarf etmeleri doğru ürünleri kolayca bulmaları kullanıcıları sisteme daha sadık hale getirir [10]. Öneri sistemleri kullanıcılarını veri yoğunluğunun yükü altında bırakmadığından sistemin kullanımı kolaylaşır. Tüm bu faydalar göz önünde bulunduğu kullanıcı ve sistemin sahibi birçok avantaj elde etmiş olur.

Giriş bölümünün devamı şu şekilde organize edilmiştir: Bölüm 1.1’de öneri sistemlerinde kullanılan algoritmalar anlatılmıştır. Bölüm 1.2’de Ortak Filtreleme algoritma yaklaşımları tartışılmıştır. Son olarak Bölüm 1.3’de tez’in amacı, konusu, kapsamı ve katkılarından bahsedilmiştir.

1.1. Öneri Sistemleri Algoritmaları

Öneri sistemleri kullanıcılarına öneriler ve tahminler üretirken çeşitli algoritmalar kullanılmaktadır. Bunlardan en çok kullanılan dört tanesi aşağıda sunulmaktadır.

1.1.1. Ortak filtreleme

En önemli ve en çok kullanılan öneri sistemleri araçlarından birisi Ortak Filtreleme (OF) yöntemidir [11]. Bu yöntem diğer kullanıcıların seçimlerini kullanarak verileri filtreler. Basitçe geçmişte benzer zevklere sahip olan kullanıcıların gelecekte de aynı zevklere sahip olacağı inancına dayanır (Şekil 1). Temel olarak *OF* şu yöntemi izler: kullanıcılar öğeleri değerlendirir ve aktif kullanıcıya öneride bulunmak için diğer kullanıcıların davranışlarını inceleyerek bir öneri oluşturma yolu bulur. Kullanıcıların ayırıcı özelliklerini eşleştirerek benzer kullanıcılar bulur ve aktif kullanıcıya diğerlerinin beğendiği öğeyi öneri olarak sunar. *OF* kullanıcılarının geçmişteki davranışlarının, tercihlerinin ya da çeşitli aktivitelerinin oluşturduğu veri kümesi üzerinde çalışarak onlar arasında benzerlikler bulur. Bulduğu bu benzerlikleri kullanarak da kullanıcılara önerilerde bulunur. *OF* geçmişte aynı zevklere sahip kişilerin gelecekte de aynı zevklere sahip olacağını varsayar. Böylece geçmişte benzer tercihler yapan kişilerin gelecekte de benzer tercihleri yapacağı çıkarımında bulunur [12].



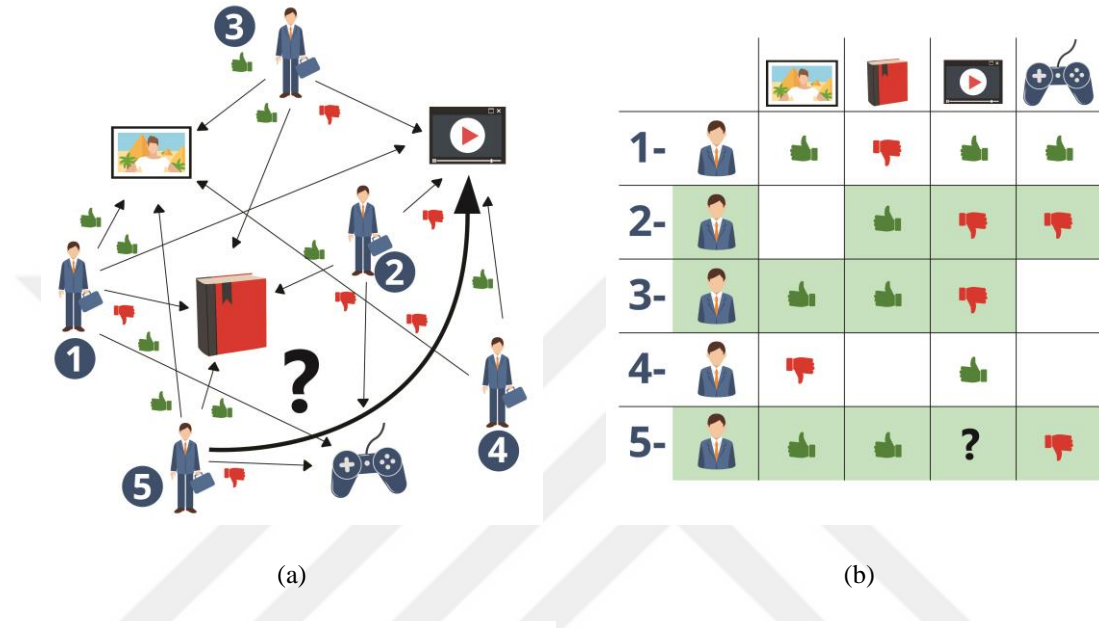
Şekil 1.1. Ortak Filtreleme Temel Çalışma Prensibine bir Film Öneri Sistemi Örneği

OF kullanılan bir öneri sisteminde kullanıcı ve ürün sayıları belirlidir. *OF* içinde kullanıcıların ve ürünlerin bulunduğu bir kullanıcı ürün matrisi oluşturur. Bu matris içerisinde kullanıcılarının daha önce tercih ettikleri ürünler hakkında yaptığı değerlendirmeleri saklar. Gerçek hayatta *OF* veri yoğunluğu yüksek olan öneri sistemlerinde kullanıldığından kullanıcı-ürün matrisi çok sayıda veriden oluşur. Ancak her kullanıcı her ürünü değerlendiremeyeceği için bu kullanıcı-ürün matrisi boşluklu yapıda olur. *OF* kullanan öneri sistemi çeşitli yöntem ve algoritmalarla kullanıcı-ürün matrisini üzerindeki veriler işlenir. Kullanıcı ve/veya ürünler arasında benzerlikler bulunur. Bulunan bu benzerlikler ile de kullanıcılara ürünler hakkında öneriler ve tahminler üretilir.

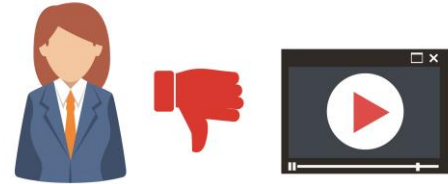
Konuyu daha teknik olarak ele almak gerekirse, n sayıda kullanıcı ve m sayıda da ürün barındıran bir sistem ele alındığında $[n \times m]$ adet kullanıcı ürün matrisi içeren bir *OF* senaryosu karşımıza çıkar. Gerçek hayat örneklerinde bu tarz sistemlerin çok sayıda kullanıcı ve ürünleri olduğu düşünüldüğünde bu matrisin boyutunun oldukça yüksek boyutlu bir matris olduğu söylenebilir [7]. Bu yapıdaki bir sistemde ürünler hakkında değerlendirme sayısı az olursa matris boşluklu yapıda olur. Matrisin boşluklu yapıda olması *OF* için bir sorun teşkil eder. Öneri sistemlerinin ürettiği öneri ya da tahminler aktif kullanıcıya sunulur. Hedef ürün ise üzerinde öneri üretilen üründür. Öneri sistemlerinin çalışma mantıklarına göre aktif kullanıcı ile benzerlik gösteren diğer sistem kullanıcıları bulunur. Çeşitli benzerlik algoritmaları benzerlikleri hesaplamak için kullanılır. Bu benzerlik sonucunda komşuluklar oluşturulur. Oluşturulan komşuluklar *OF* algoritmalarıyla işlenir ve hedef ürün hakkında aktif kullanıcıya bir öneri sunulur.

Ortak Filtreleme yönteminin kullanımına ilişkin bir örnek aktivite öneri sistemine ait bilgiler Şekil 1.2’de sunulmaktadır. Bu aktivite öneri sisteminde beş kullanıcı bulunmaktadır. Bu kullanıcılar dört boş zaman aktivitesi için değerlendirmelerde bulunmuştur. Kullanıcılar değerlendirmelerini beğenme ve beğenmeme olarak ikili seçim şeklinde yapmıştır. 5 numaralı kullanıcının video izleme aktivitesini beğenip beğenmeyeceği hakkında sistemden öneri üretmesi beklenmektedir. Bu bağlamda, aktif kullanıcı olan 5. kullanıcı, hedef ürün ise video izleme aktivitesidir. *OF* algoritması hedef ürün hakkında aktif kullanıcıya öneri veya tahmin üretirken, diğer kullanıcılarla arasında benzerliklerini kullanır. Bu açıdan tablo incelendiğinde 5 numaralı kullanıcı ile en benzer kullanıcıların 2 ve 3 numaralı kullanıcılar olduğu gözlemlenmektedir. 5 numaralı

kullanıcı ile 2 ve 3 numaralı kullanıcılarının diğer aktivitelere ilgilerinin benzer olduğundan, *OF* algoritması video izleme aktivitesinde de 5'in 2 ve 3 ile aynı zevklere sahip olacağı yaklaşımında bulunur. Bu sebeple 2 ve 3'ün video izleme aktivitesine yaptıkları değerlendirme, öneri sisteminin 5'in video izleme aktivitesini beğenip beğenmeyeceğini önermesi için anlamlıdır.



1-					
2-					
3-					
4-					
5-					



(c)

(d)

Şekil 1.2. Örnek Aktivite Öneri Sistemi

OF ve diğer öneri algoritmalarının öneri üretebilmeleri için kullanıcılardan ürünler hakkında yaptıkları değerlendirmeleri toplaması gerekir. Değerlendirme toplama işlemi

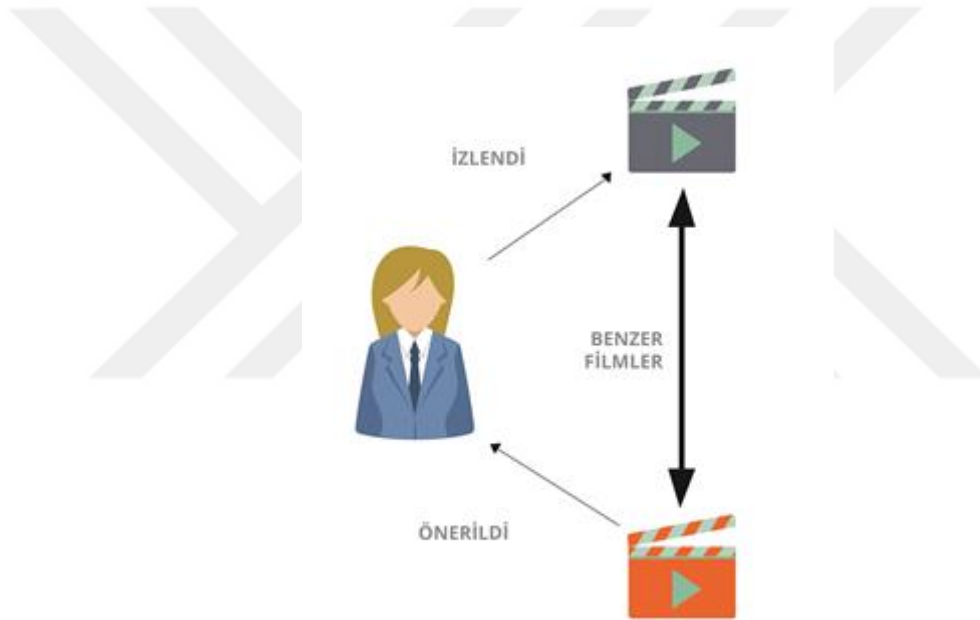
açık ve *örtülü* olmak üzere iki farklı yolla yapılabilir. *Açık* değerlendirmede buradaki örnekte olduğu gibi kullanıcıdan somut bir derecelendirme ölçütünde doğrudan bir değerlendirme yapması istenir. İkincisi ise kullanıcının sitedeki eylemlerini günlüğe kaydeder. Açık veri toplama ile çalışmak kolaydır. Bu değerlendirme ölçütü 0 (beğendi), 1 (beğenmedi) olabileceği gibi çok farklı şekillerde de olabilir. Burada önemli olan sistem sahibinin ürünlerinin nasıl değerlendirmesini istediğidir. Örneğin, kimi sistemlerde 1-5 arasında yıldızlarla değerlendirme yapılırken kimilerinde ise kullanıcılardan bazı görseller ile derecelendirme yapmaları beklenebilir.

Açık veri toplama yönteminde bir kullanıcının sağladığı derecelendirmeler, kullanıcının tercihleri olarak doğrudan yorumlanabilir ve gelecekteki derecelendirmeleri tahmin etmek için verilerden çıkartım yapmayı kolaylaştırır. Ancak, açık verilerle çalışmak ile ilgili dezavantajlarda mevcuttur. Kullanıcılar derecelendirmeye girmek için zaman ayırmak istemeyebilirler ya da kullanıcılar kasti olarak/olmayarak ilgi alanları dışında değerlendirme yapabilirler. Diğer yandan, örtük verilerin, kullanıcı adına herhangi bir çaba göstermeden büyük miktarlarda toplanması kolaydır. Bu yöntemin dezavantajı ise çalışma zorluğudur. Bu yöntemin amacı kullanıcı davranışını kullanıcı değerlendirmelerine dönüştürmektir. Açıktan veri toplama ve örtülü veri toplama melez olarak kullanılabilir.

OF çok kullanılan bir öneri sistemleri algoritması olsa da bazı sorunlara da sahip olduğu bilinmektedir. *OF* öneri sistemleri genellikle üç sorundan mustarıptir: *soğuk başlangıç*, *ölçeklenebilirlik* ve *boşluklu yapı* [13]. *Soğuk başlangıç* sisteme yeni bir kullanıcı ya da ürün eklendiğinde ortaya çıkar. Yeni bir eklemede kullanıcı ya da ürün hakkında öneri üretmek için sistem yeterli veriye sahip olamadığından öneri üretemez veya başarılı bir öneri üretemeyebilir [14, 15]. *OF* yaklaşımını kullanan öneri sistemlerinde genellikle veri yoğunluğu oldukça yüksektir. Bu yoğun veri içerisinde bir kullanıcıya bir ürün hakkında öneri sunmak oldukça zor olabilir ve önerileri hesaplamak için oldukça yüksek bir hesaplama gücü gerekir. Bu durum ölçeklenebilirlik sorunu ortaya çıkarır [16]. *Boşluklu yapı* sorunu öneri sistemindeki kullanıcıları tüm ürünler hakkında değerlendirme yapamayacağı gerçeğine dayanmaktadır. Her kullanıcı her ürün hakkında bir değerlendirme yapamayacağı için kullanıcı ürün matrisi boşluklu yapıda olur. [16]

1.1.2. İçerik tabanlı filtreleme

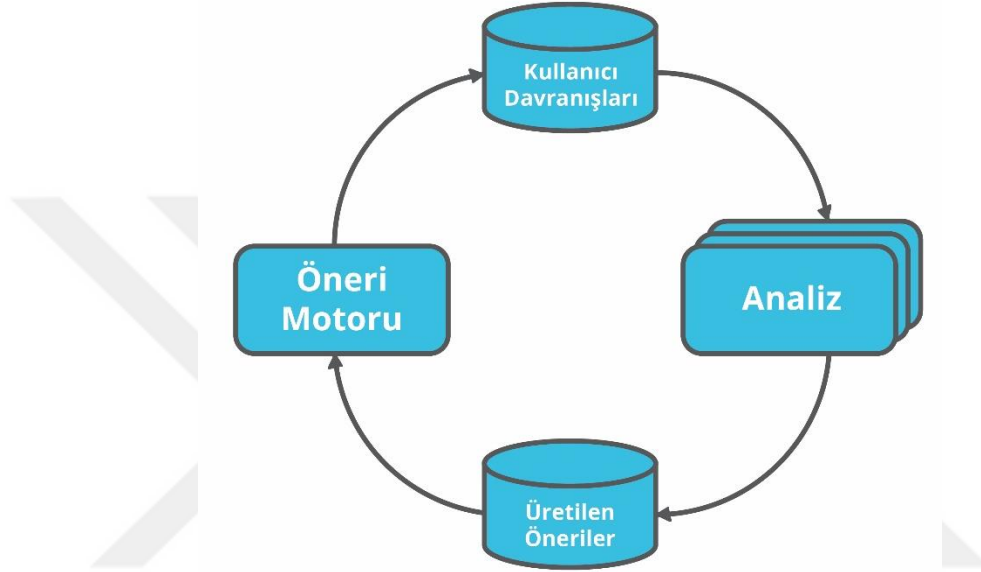
En yaygın kullanılan öneri sistemleri algoritmalarından bir diğeri ise içerik tabanlı filtrelemedir (İTF). İTF de öneriler ürünün açıklamasına ve kullanıcı tercihlerinin profiline bağlıdır [17, 18]. Ürünlerle ilgili bilgiler/özellikler kullanıcı tercih profili oluşturmak için kullanılır. Böylece kullanıcının geçmişte tercih ettiği ürünlerin özellikleri o kullanıcıya yapılacak önerilerde kullanılır (Şekil 1.3). Yani yalnızca kullanıcı-ürün etkileşimlerine dayanan OF yöntemlerden farklı olarak, içerik tabanlı yaklaşımlar kullanıcılar ve öğeler hakkında *özellikler* kullanır. Ayrıca İTF’de kullanıcılar arasında bir benzerlik hesaplaması yapılmaz.



Şekil 1.3. İçerik Tabanlı Filtreleme Film Öneri Sistemi Örneği

Bir film öneri sisteminden örnek vermek gerekirse, bahsi geçen *özellikler* kullanıcı için yaş, cinsiyet, iş, gelir, hobiler vs. olabilir. Bir film için ise kategori, başrol, uzunluk, yönetmen ve diğer karakteristik özellikler olabilir. Bu bilgilerden sonra içerik tabanlı öneri sisteminin yapması gereken adım kullanıcı ürün etkileşimlerini tanımlayan ve bu *özellikler* üzerine tanımlanmış bir model oluşturur. Film öneri sistemi için örnek bir model olarak “Genç kadınların romantik-komedi filmlerini daha çok sever” ya da “Genç erkekler aksiyon filmlerini daha iyi derecelendirme eğilimindedir” sunulabilir. İçerik tabanlı öneri sistemi bir model oluşturduktan sonra öneri üretilebilir.

Bir *İTF* öneri sistemi temel olarak kullanıcının açıktan veya örtülü olarak sunduğu veriler üzerinde çalışır. Bu veriye dayanarak çıkarılan kullanıcı profili, kullanıcıya öneri sunarken kullanılır. Kullanıcı sistemi daha çok kullandıkça sistemde daha çok veri ve davranış bırakır. Böylece öneri sistemi gittikçe daha doğru öneriler sunmaya başlar. Şekil 1.4'te *İTF*'nin çalışma prensibi gösterilmiştir.



Şekil 1.4. *İTF* Öneri Motoru Çalışma Prensibi

İTF'de karmaşık bir hesaplama işlemi mevcut değildir. Öneriler kişiye özel üretildiğinden sadece aktif kullanıcı verileri ile ürün verileri karşılaştırılır. Bu durum ölçeklenebilirliği yükseltmektedir. Daha az hesaplama gücüyle daha yoğun veri üzerinde çalışabilir. Oluşturulan profil modeli bir kullanıcının özel ilgi alanını yakalayabilir ve daha az kullanıcının ilgisini çeken bir ürün aktif kullanıcıya öneri olarak sunulabilir. Soğuk başlangıç problemlerinde nispeten daha az etkilenir. Sisteme eklenen yeni ürünler/kullanıcılar hakkında özellikler sisteme girilebilir ve böylece bu ürünler hakkında kullanıcılara öneri sunulabilir. Ancak daha önce görülmeyen *özelliklere* sahip yeni kullanıcı ve ürünler soğuk başlangıç probleminden etkilenebilir. Tüm bunların yanı sıra *İTF* dezavantajlara da sahiptir. Bir model oluşturmak oldukça zor olabilir. Bir ürünler hakkında bir model çıkartabilmek için o konu hakkında oldukça fazla bilgiye sahip olmak gerekir. Bu sebeple model sahip olunan bilgi kadar iyi olabilir. Ayrıca model kullanıcının

sadece tanımlanan özellikleri için öneriler sunabilir. Başka bir deyişle model kullanıcıların mevcut ilgi alanlarını genişletme yeteneğine sahip değildir.

1.1.3. Bilgi Tabanlı Filtreleme

Bilgi Tabanlı Filtreleme ürün çeşitliliği, kullanıcı tercihleri ve öneri kriterleri hakkındaki açık bilgilere dayanmaktadır [17, 19]. Genellikle *OF* ve *ITF*'nin kullanılmadığı yerlerde kullanılırlar. Özellikle çok fazla alıcısı olmayan nadir alınan lüks ürünlerde *BTF* yaygın olarak kullanılır. Örnek olarak emlak, lüks araçlar, finansal hizmetler ya da pahalı lüks ürünler sayılabilir. Bu tarz ürünlerde etkileşim miktarı az olduğu için kullanıcılardan derecelendirme almak oldukça zordur. Bu sebeple bir soğuk başlangıç problemi ortaya çıkmaktadır. Ayrıca kullanıcının öneri alma süreci içerisinde istediği özelliklerin nitelikleri değişebilir. Diğer taraftan öneri sistemi tarafından aktif kullanıcı için her ne kadar benzer kullanıcılar bulunsa da, aktif kullanıcının aradığı ürün, ürünün birçok niteliğinin çok özel bir kombinasyonu olabilir. Kullanıcının bir yat satın almak istediği senaryoda kullanıcıdan yatın rengi, motor gücü, uzunluğu, tipi, markası, modeli, kamara sayısı gibi birçok özelliği zevkine göre tahmin etmek zor olabilir. Bu tür durumlar, derecelendirmelerin tavsiye amacıyla kullanılmadığı bilgiye dayalı öneri üreten sistemler ile ele alınabilir. *BTF* kullanılarak kullanıcıdan bu özellikleri sisteme girmesi beklenir [17].

BTF'nin en büyük avantajı soğuk başlama probleminin olmayışıdır. Başlangıçta kullanıcıdan ilgilendiği ürün ile ilgili özellik bilgilerini talep eder. Bu özellikler önceden ürünler için de tanımlanmıştır. Böylece kullanıcının tam olarak isteğine karşılık gelen ürünler kullanıcıya öneri olarak sunulur.

Öneri sistemleri ara yüz tipine göre ikiye ayrılabilir;

Kısıtlama Tabanlı Öneri Sistemleri: Bu sistemlerde kullanıcı ürün özellikleri ile ilgili kısıtlamaları belirtir. Örneğin, otomobil önerisi alınan bir sistemde aracın model üst sınırı ya da bugüne kadar yaptığı kilometre üst sınırı gibi kısıtlamaları belirtebilir. Bu yöntemle ait bir örnek Şekil 1.5'da verilmiştir Kısıtlamaya dayalı tavsiye sisteminde kullanıcının kişisel özellikleri (Yaş, eğitim durumu vb.) de öneri üretmek için kullanılabilir. Öneri sonuçlarına bağlı olarak kullanıcı kısıtlamalarda değişiklik yapabilir. Kullanıcı aradığı sonuca ulaşana kadar kısıtlamalar üzerinde gevşetme veya daha kısıtlı bir arama gerçekleştirebilir [20, 21].

LÜKS KONUT SATIŞI İÇİN KISITLAMA TABANLI BİLGİ FİLTRELEME
ÖRNEK ARAYÜZÜ

SATIN ALINMAK İSTENEN KONUTUN ÖZELLİKLERİ:

MİN. ODA SAYISI ▼▼ MAK. ODA SAYISI ▼▼ MİN. BANYO SAYISI ▼▼ MAK. BANYO SAYISI ▼▼

MİN. FİYAT ▼▼ MAK. FİYAT ▼▼ KONUT TİPİ ▼▼ POSTA KODU

BUL

Şekil 1.5. *Kısıtlama Tabanlı Bilgi Filtreme örnek ara yüzü*

Vaka Tabanlı Öneri Sistemleri: Kullanıcı tarafından özel durumların belirtildiği öneri sistemleridir. Örneğin kullanıcı bir emlak öneri sisteminde aradığı bir emlak için tam olarak zaten bildiği bir konutun tam adresini belirtebilir. Bir örneği Şekil 1.6'da sunulmaktadır. Benzerlik ölçütleri bütün ürünler için önceden tanımlanmış özellikler üzerinden yapılır. Benzerlik ölçütleri alana özel bir şekilde özenle tanımlanır. Kullanıcıya döndürülen sonuçlar kullanıcı ile etkileşimli bir şekilde değişerek yeni hedef ürünleri oluştururlar. Yani kısıtlamaya dayalı öneri sisteminde olduğu gibi kullanıcı çeşitli yöntemlerle sorgu özellikleri değiştirilebilir [22, 23, 24, 25].

LÜKS KONUT SATIŞI İÇİN VAKA TABANLI BİLGİ FİLTRELEME ÖRNEK ARAYÜZÜ

SATIN ALINMAK İSTENEN KONUTUN ÖZELLİKLERİ

ODA SAYISI ▼▼ BANYO SAYISI ▼▼ KONUT TİPİ ▼▼

FİYAT ARALIĞI ▼▼ POSTA KODU

BUL

AŞAĞIDA GİRİLMİŞ OLAN ADRESE BENZER BİR KONUT BUL

ÖRNEK MAHALLE ÖRNEK SOKAK NO:11 D:5 ODUNPAZARI ESKİŞEHİR ▼▼

BUL

Şekil 1.6. Vaka Tabanlı Bilgi Filtreleme örnek ara yüzü

1.1.4. Melez Algoritmalar

Öneri sistemleri daha performanslı çalışmak adına artık genelde Ortak Filtreleme, İçerik Tabanlı Filtreleme, Bilgi Tabanlı Filtreleme yaklaşımlarını birleştiren melez yaklaşımları kullanmaktadır. Örneğin, *OF* öneri sistemleri topluluk derecelendirmelerine, *İTF* öneri sistemleri metinsel açıklamalara ve aktif kullanıcının kendi derecelendirmesine, *BTF* öneri sistemleri ise bilgi tabanlı bağlamında kullanıcıyla etkileşime dayanır. Bu farklı sistemler farklı girdi tipleri kullanır ve öneri üretirken farklı yollara başvururlar. Hepsinin farklı güçlü ve zayıf yanları mevcuttur. *BTF* öneri sistemleri verinin az olduğu durumlarda dahi soğuk başlama probleminden etkilenmez. Diğer yandan *OF* verinin yoğun olduğu sistemlerde daha etkili çalışır.

Melez yaklaşımlar ihtiyaca göre farklı şekillerde kullanılabilirler [26]. *İTF* ve *OF* yöntemleri ayrı ayrı çalıştırılarak sonuçlar birleştirilebilir. Diğer bir yol ise *İTF*'nin yetenekleri *OF*'ye eklenerek ya da tam tersi yapılarak melez bir yapı oluşturulabilir. Tüm bunların dışında tüm yaklaşımlar tek bir modelde birleştirilerek de kullanılabilir [27]. Bu

yöntemler *soğuk başlangıç*, *ölçeklenebilirlik* ve *boşluklu yapı* problemlerini çözmek için birlikte kullanılırlar. Yapılan çalışmalar Melez yöntemlerin, saf *İTF* ya da *OF*'den daha performanslı çalıştığını göstermektedir [26, 28]. Netflix platformu melez öneri sistemleri yaklaşımına en güzel örneklerden biridir. Sisteme ilk kayıta beğenilen filmleri seçtirerek *BTF* kullanır. Platformu kullanmaya devam ettikçe kullanıcılar arasında *OF* kullanarak benzerlik hesaplaması yapar ve benzer kişilere ortak zevklerine uygun öneriler sunar. Bunların yanında kullanıcılarının beğendikleri(açıktan veya örtülü) içeriklere benzer içerikleri *İTF* kullanarak kendilerine önerir [28].

1.2. Ortak Filtreleme Algoritma Yaklaşımları

Ortak filtreleme yaklaşımları kullanıcıların benzerlikleri üzerine hesaplamalar yapar ve öneriler oluşturur. *OF* öneri üretmek için iki temel yaklaşımı kullanır. Ayrıca *OF* bu iki yaklaşıma ek olarak bunların karması olan başka bir yaklaşımı kullanır [5]. Bu başlık altında Hafıza Tabanlı yaklaşım, Model Tabanlı yaklaşım ve Karma yaklaşımlar anlatılacaktır.

1.2.1 Hafıza Tabanlı Ortak Filtreleme Algoritması

Hafıza tabanlı ortak filtreleme (HTOF) yaklaşımı literatürde komşuluk tabanlı yaklaşım olarak da bilinmektedir. *HTOF* kullanıcılar veya ürünler arasındaki benzerliği hesaplamak için kullanıcı derecelendirme verilerini kullanır. Bu veriler tüm kullanıcı ürün verilerini kapsamaktadır. *HTOF* aktif kullanıcıya benzer zevklere sahip olan kullanıcıları bulur. Benzerlikleri bulmak için çeşitli istatistiksel teknikler kullanır. Bunlardan en popülerleri Pearson korelasyon katsayısı ve kosinüs benzerliğidir [29, 30]. Pearson korelasyon katsayısı ve kosinüs benzerliği formülleri Denklem 1.1 ve Denklem 1.2'de sunulmuştur.

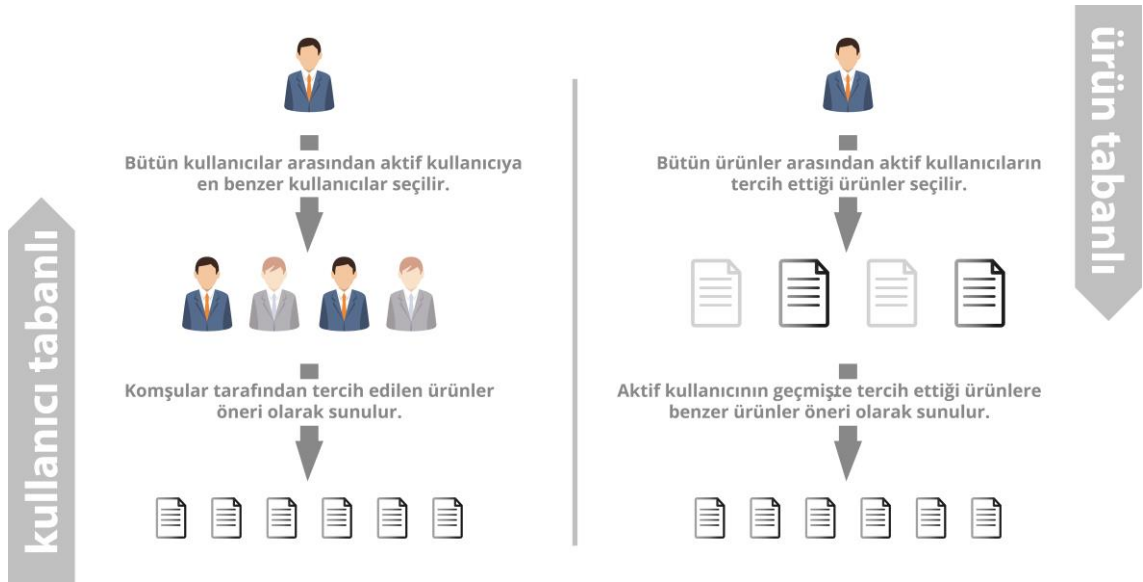
$$Pearson(k, k') = \frac{\sum_{\ddot{u} \in I(k, k')} (R(k, \ddot{u}) - \overline{R(k)}) (R(k', \ddot{u}) - \overline{R(k')})}{\sqrt{\sum_{\ddot{u} \in I(k, k')} (R(k, \ddot{u}) - \overline{R(k)})^2} \sqrt{\sum_{\ddot{u} \in I(k, k')} (R(k', \ddot{u}) - \overline{R(k')})^2}} \quad (1.1)$$

$$Kosinüs(k, k') = \frac{\sum_{\ddot{u} \in I(k, k')} R(k, \ddot{u}) R(k', \ddot{u})}{\sqrt{\sum_{\ddot{u} \in I(k, k')} R(k, \ddot{u})^2} \sqrt{\sum_{\ddot{u} \in I(k, k')} R(k', \ddot{u})^2}} \quad (1.2)$$

Bu denklemlerde k aktif kullanıcıyı, u ise hedef ürünü temsil etmektedir. $R(k,u)$ kullanıcının o ürün hakkında yaptığı puanlamayı temsil eder. $I(k, k')$ 'da k ve k' kullanıcılarının ortak puanladıkları ürünler grubunu temsil etmektedir.

Pearson Korelasyon katsayısı yöntemi iki değer arasında ilişki bulma konusunda oldukça etkili bir yöntemdir. X ve Y değişkenleri arasındaki alanın bir ölçütüdür. Bu denklem kullanılarak iki değişken arasındaki alanın ölçümü için yapılan hesaplamalarda çıkan sonuç +1 ile -1 arasında bir değere sahiptir. Burada ikili arasında hesaplanan korelasyon katsayısının 1 çıkması iki değişkenin pozitif alakalı olduğunu, 0 çıkması aralarında alaka olmadığını ve -1 çıkması negatif alakalı olduklarını gösterir. [31][32]

Benzerlikler bulunduktan sonra aktif kullanıcıya benzerliği en yakın olan kullanıcılar komşular olarak tespit edilir. Bu komşuluk içerisindeki kullanıcıların yaptıkları ürün derecelendirmeleri çeşitli algoritmalar kullanılarak ağırlıklandırılır ve hedef ürün hakkında aktif kullanıcıya bir tavsiye ya da ilk N öneri listesi sunulur. Bu algoritmalar en yakın n komşu, kullanıcı tabanlı ortak filtreleme (KTOF) ya da ürün tabanlı ortak filtrelemedir (ÜTOF). Şekil 8'de Kullanıcı tabanlı ve Ürün tabanlı öneri sisteminin çalışma mekanizmasını gösteren bir görsel sunulmaktadır.



Şekil 1.7. Kullanıcı tabanlı ve Ürün tabanlı öneri sistemlerinin çalışma mekanizması

1.2.1.1. Kullanıcı tabanlı ortak filtreleme

KTOF'da aktif kullanıcıyla benzer zevklere sahip kullanıcıların sağladığı derecelendirmeler aktif kullanıcıya öneri sunulurken kullanılır. Bu nedenle, ana fikir aktif kullanıcıya benzer kullanıcıların tespit edilmesi ve bunlardan komşuların seçilmesidir. Komşular seçildikten sonra komşuların derecelendirmelerinin ağırlıklı ortalaması alınarak aktif kullanıcıya hedef ürün hakkında öneri sunulur [33]. Örneğin, Ali ve Ayşe bir film öneri sisteminde geçmişte filmleri benzer şekilde değerlendirmişlerse, Ayşe'nin izleyip değerlendirdiği Titanic filmi için yaptığı derecelendirme Ali'ye Titanic filmi için öneri üretilirken kullanılabilir. Genellikle aktif kullanıcıya öneri üretilirken, aktif kullanıcıya en yakın k komşu kullanılır [34].

1.2.1.2. Ürün tabanlı ortak filtreleme

ÜTOF yönteminde benzerlikler ürünler üzerinde hesaplanır. Aktif kullanıcıya hedef ürün önermek için öncelikle hedef ürüne en çok benzeyen n adet ürün belirlenir. N adet ürün daha önce aktif kullanıcı tarafından değerlendirilmiş ürünlerden oluşur ve bu ürünler aktif kullanıcıya öneri üretmek için kullanılır. Örneğin, bir film öneri sisteminde aktif kullanıcının daha önceden değerlendirdiği Marslı ve Yıldızlararası gibi bilim kurgu filmlerine yaptığı derecelendirmeler, daha önce değerlendirmedeği Ben Efsaneyim filmi için derecelendirme tahmini üretmede kullanılabilir.

1.2.2. Model tabanlı ortak filtreleme algoritması

Hafıza tabanlı yaklaşımların en büyük avantajı uygulanmalarının kolay olmasıdır. Ancak hafıza tabanlı yaklaşımlar boşluklu yapıda olan matrisler için çok iyi öneri sonuçları üretemez. Örneğin, Ali'ye benzer, Titanic filmini değerlendiren çok fazla kullanıcı olmaması durumunda, Ali'ye Titanic filmi hakkında sağlam bir öneri üretmek oldukça zor olabilir. Bu gibi durumlarda Model Tabanlı Ortak Filtreleme (*MTOF*) yaklaşımı kullanılması daha avantajlı olabilir. *MTOF* algoritması derecelendirme veri seti üzerinde bir model oluşturur. Veri setinden çıkardığı bu model sayesinde veri tabanındaki tüm verileri kullanmak zorunda kalmaz. Bu "model" sistem tarafından öneri üretmek için kullanılır. *MTOF* algoritmaları ile üretilen modeller, veri setinin tamamından çok daha küçüktür; yani, çok büyük veri setleri için bile, "model" verimli bir şekilde kullanılabilir kadar küçüktür. Bu durum sistemin genelini ölçeklenebilir kılar. Böylece ölçeklenebilirlik *HTOF* algoritmasına göre daha yüksektir. Benzer şekilde tüm veri seti kullanılmadan, yalnızca model üzerinden veri tabanı sorgulaması yapıldığı için *MTOF*

algoritmasında hız da daha yüksektir. *MTOF* algoritması, ölçeklenebilirlik, boşluklu yapı ve hız problemlerinin üstesinden gelse de doğru öneri üretme açısından bakıldığında veri tabanındaki tüm veriyi kullanarak öneri üreten *HTOF* algoritmasına göre performansı daha düşüktür. Ayrıca başlangıçta “model” oluşturmak oldukça zor ve sofistike bir süreçtir [35].

Model tabanlı öneri sistemleri genel olarak aynı temel çalışma prensibine sahip olsalar da model oluşturmak ve o modelleri kullanmak için farklı yaklaşımlar kullanırlar. Bunlardan bazıları: Bayes ağları, kümeleme modelleri, tekil değer ayrışımı gibi gizli anlamsal modeller, olasılıksal gizli anlamsal analiz, çoklu çarpımsal faktör, gizli Dirichlet dağılımı ve Markov karar süreci temelli modellerdir [36].

1.2.3. Karma algoritmalar

Hafıza ve model tabanlı yaklaşımların avantaj ve dezavantajları mevcuttur. Karma yaklaşımlar bu iki yöntemin avantajlarını bir araya getirmek için ortaya çıkarılmış algoritmalarlardır. İki yaklaşımda karşılaşılan *soğuk başlangıç*, *ölçeklenebilirlik*, *boşluklu yapı*, *bilgi kaybı*, *hız* ve *performans* gibi sorunların çözülebilmesi için karma yaklaşımlar kullanılmaktadır. Her ne kadar bu sorunların üstesinde gelse de, karma yaklaşımları kullanmak maliyetlidir ve karmaşıklığı arttırmaktadır [37].

OF'nin uygulanması sırasında karşılaşılan bazı problemler ve çözümü için kullanılan karma yaklaşımlar aşağıda sunulmaktadır.

Soğuk başlangıç problemi: Sisteme yeni ürün ya da kullanıcı eklendiğinde soğuk başlangıç problemi ortaya çıkmaktadır [27, 38]. Yeni ürün hakkında yeterli derecelendirme olmadığından ürün hakkında öneri sunmak zordur. Diğer taraftan sisteme yeni eklenen bir kullanıcının da geçmiş tercihleri bilinmeyeceğinden yeni kullanıcıya benzer kullanıcılar tespit etmek oldukça zordur. Bu probleme soğuk başlangıç problemi denmektedir. Bu problem diğer bir *OF* yaklaşımı problemi olan boşluklu yapı problemine de yol açmaktadır. Boşluklu yapı probleminin çözümü için çeşitli boyut indirgeme algoritmalarını temel alan *OF* algoritmaları sunulmuştur. Temel Bileşenler Analizi, Tekil Değer Ayrışımı, Gizli Anlamsal İndeksleme bunlardan bazılarıdır [12, 39].

Ölçeklenebilirlik problemi: Çok büyük veri yoğunluğuna sahip olan öneri sistemlerinde benzerlik ve öneri hesaplamaları oldukça zor hale gelebilir. Bu duruma

ölçeklenebilirlik problemi adı verilir. Çeşitli boyut indirgeme algoritmaları sayesinde bu problemin üstesinden gelmek mümkündür [40].

Aşırı benzerlik problemi: Veri seti içerisinde bulunan bazı veriler birbirine çok benzeyebilir ve hatta aynı ürünler farklı isimle kaydedilebilir. Bu gibi durumlarda *OF* öneri sistemi farklı isimlerdeki ürünleri farklı olarak nitelendireceğinden kullanıcıya birden çok defa aynı ürünü tavsiye edebilir. Bu durum kullanıcı deneyimini olumsuz etkiler ve istenmeyen bir durumdur. Bu problem *OF* algoritması çeşitli algoritmalarla birlikte karma olarak kullanılarak çözümlenir [41].

Bu ve bunun gibi problemlerle karşılaşan birçok öneri sistemi uygulaması durumun üstesinden gelmek için model tabanlı ve hafıza tabanlı ortak filtreleme yaklaşımlarını karma olarak kullanır. Bunlara ek olarak farklı algoritmalar ve yaklaşımlarla birlikte de *MTOF* ve *HTOF* karma olarak kullanılabilir. Karma yaklaşımlar yalnız *OF* yaklaşımının sınırlarının üstesinden gelmek ve tavsiye performansını iyileştirmek için kullanılır.

1.3. Amaç, Konu, Kapsam ve Katkılar

Üniversiteye yerleşen öğrencilerin çoğunluğu şehir dışından gelmektedir. Bu durum ders seçim zamanlarında öğrenciler için büyük bir dezavantaj haline gelmektedir. Öğrenciler genellikle hangi derslerin ilgi alanlarına uygun olduğunu bilemezler. Bunun yanı sıra öğrenciler akademik olarak yeterlilik sağlayabilecekleri dersleri seçmek isterler. Bu gibi durumlarda öğrenciler çoğunlukla geleneksel yöntemlere başvururlar. Almak istedikleri ders ile ilgili bilgileri daha önce almış olan arkadaşlarına sorarak öğrenmeye çalışırlar. Diğer bir durumda da dersin yürütücüsü ile iletişime geçip ders ile ilgili bilgi almaya çalışırlar ya da kendi akademik danışmanlarından dersler hakkında bilgi alıp ders seçim işlemlerini tamamlarlar. Bu gibi geleneksel yöntemlerin yürütülmesi oldukça zordur. Sürekli olarak kişiler arasında iletişim kurma ihtiyacı doğurur. Bu durum kampüsün bulunduğu şehirden farklı şehirlerde yaşayan öğrenciler için gerçekleştirilmesi oldukça zor bir bilgi edinme türüdür. İletişim kurulup gerekli bilgiler alınabilse bile dersler ile ilgili sözlü olarak öneri veren kişiler tavsiye isteyen öğrencinin akademik başarı profilini tam olarak bilemeyecekleri için sağlıklı öneri veremeyebilirler. Bu gibi durumlarda otomatik bir ders öneri sistemine ihtiyaç duyulur.

Bu çalışmada kullanıcı tabanlı ortak filtreleme yöntemi kullanılarak Eskişehir Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü (ESTÜBM) öğrencilerine alabilecekleri derslerle ilgili öneri sunulması hedeflenmiştir. Bu çalışmanın

temel amacı üniversite öğrencilerinin akademik başarılarını yükseltmektir. Çalışma sayesinde öğrenciler ilgi alanlarına uygun derslere yönlendirilerek akademik başarılarının yükseltilmesi hedeflenmiştir.

Bu hedefi gerçekleştirmek için öğrencilerin geçmiş akademik başarılarına dayanan bir akademik profil çıkartımı yapılmıştır. Öğrencilerin akademik profillerine göre yönlendirilecekleri dersler tespit edilmiş ve öğrencilere ders öneri listesi olarak sunulmuştur. Ders öneri sunabilmek adına *ESTÜBM* veri setinden tavsiye sunulacak öğrencinin akademik profiline benzerlik gösteren üst dönem öğrenciler ve mezunlar tespit edilmiştir. Öğrenci benzerlikleri, öğrencilerin ve mezunların transkriptleri üzerinden geçmişte aldıkları derslerin harf notları kullanılarak hesaplanmıştır. Tespit edilen benzer öğrenciler arasından komşular seçilmiştir. Seçilen komşuların geçmişte alıp başarılı oldukları dersler, benzerlik oranlarına göre ağırlandırılarak öneri hesaplamaları yapılmıştır ve tavsiye sunulacak öğrenciye bir ders öneri listesi sunulmuştur. Ders öneri listesinde derse ait bilgiler (Ders adı, kredisi, öğretim elemanı vb.) ile birlikte o dersten tavsiye isteyen öğrencinin hangi harf notunu alacağı tahmini de sunulmuştur.

Diğer yandan spesifik bir ders için tavsiye isteyen bir öğrenci senaryosu üzerinde de çalışılmıştır. Bu senaryoda ise öğrenci seçmek istediği dersi belirttikten sonra ders öneri listesi sunumu senaryosunda olduğu gibi aktif öğrenciye benzer öğrenciler bulunmuştur. Benzerliklerden bir komşuluk oluşturulup o komşuluk içerisinde tavsiye istenen dersi almış öğrenciler arasından benzerlik ağırlıklandırma işlemi yapılarak harf notu tahmini üretilmiştir.

Çalışmada sunulan öneri sistemi uygulaması *ESTÜBM* 'de öğrenim gören tüm öğrencilere uygulanmıştır ve sistemin elde ettiği sonuçlar derslerin gerçek harf notları ile karşılaştırılmıştır. Çalışma sayesinde öğrencilerin doğru yönlendirilmesiyle akademik başarının yükseleceği ortaya konmuştur.

2. İLGİLİ ÇALIŞMALAR

Bu bölümde konu ile ilgili geçmişte yapılan çalışmalar kronolojik sırada sunulmaktadır. Diğer çalışmalarda ders önerileri üretmek için neler yapıldığı, hangi yöntemlerin kullanıldığı anlatılmıştır. Sonuç olarak da bu tezde sunulan çalışmanın literatürde var olan çalışmalardan farkı sunulmaktadır.

Ders öneri sistemlerinin ilk çalışmalarından birisi Farzan ve ark. [42] tarafından yapılmıştır. Bu ders öneri sistemi *BTF* yöntemini kullanmaktadır. Sistem kullanıcılarından birçok bilgi talep etmektedir. Dersleri daha önce almış kişilerden dersin işleyişiyle ve özellikleriyle ilgili bilgiler talep edilmektedir. Ayrıca dersi yürüten öğretim elemanı ile ilgili bilgilerin de talep edildiği bir bilgi formu dersi alan öğrencilere tamamlanmaktadır. Öneri sisteminin çalışması için bu verilere ihtiyaç duyulmaktadır. Yani başlangıçta bir soğuk başlatma problemi mevcuttur. Öğrencileri erişmek istedikleri kariyer hedeflerine yönlendirmeyi amaçlamaktadır. Sistem öğrencilerden, önceden tanımlanmış 22 adet kariyer yolundan istedikleri kadar seçim yapmasını beklemektedir. Seçilen kariyer hedeflerini öğrencilerin akademik danışmanlarına da ileterek öneri sistemine aktif olarak hem danışmanı hem de öğrenciyi dâhil etmektedir. Ayrıca sistem sadece bir bölümün kullanabileceği şekilde üniversitenin kendi öğrenci bilişim sisteminden ayrı olarak tasarlanmıştır.

Diğer bir ders öneri sistemi ise Nishino ve ark. [43] tarafından sunulmuştur. Bu çalışma, e-öğrenme sistemi üzerinden öğrenim gören öğrencilerin öğrenme biçimlerini çıkarımlar. Çıkarım yapılan bu öğrenme biçimlerini kullanarak, öğrenciler için uygun e-öğrenme kurslarını önerme yöntemini ve sistemini tartışır. Bunu yapmak için öğrencilere bir anket yaptırılır. Bu öğrenme tercihi anketi öğrencilere ders çalışma, anlama, sorgulama ve ödev yapma tercihleriyle ilgili sorular yöneltilir. Anket sonucunda öğrencinin tercihi tespit edilir. Tespit edilen bu tercih ışığında veri tabanında bulunan önceki öğrenme verilerine göre öğrencilere bir ders önerilir. Bu çalışma, öğrencinin geçmişteki öğrencilerin öğrenme tercihleri ve kursun uyumluluğuna dayanarak her bir kursa uygunluğunu önermek için çoklu regresyon analizleri yapar. Çoklu regresyon modelini kullanarak, bir öğrenci önceden belli bir e-öğrenme kursuna uygunluğu hakkında bilgilendirilir. Çalışma bir ders listesi önermek yerine tek bir ders üzerinde öğrencinin başarılı olup olamayacağını ya da hangi derste başarılı olacağını tahmin eder.

Chang ve ark. [44] gerçekleştirdikleri ders öneri sistemi çalışmasında öğrencilere ders önerisi sunmak için birden fazla faktörü bir araya getirerek öneri üretmişlerdir. Öğrencilerin aldığı dersler ve aldıkları puanları içeren verileri toplamışlardır. Ardından Öğretim elemanlarının öğrenciler arasında popüler olup olmadığına bakarak ders öğretmenleri hakkında bilgi toplamışlardır. Daha sonra bu üç veriyi önerdikleri sınıflandırma modeline girdi olarak uygulamışlardır. Böylece oluşturdukları model kullanıcıların hangi derslerde yüksek puan alabileceklerini tahmin etmiştir. Dolayısıyla bu öneri sisteminde Model Tabanlı Ortak Filtreleme algoritması kullanılmıştır. Bu çalışma, öğrencileri belirli kategorilerde kümeleyerek ders önerileri sunmaktadır. Kümeleme işlemini gerçekleştirmek için Yapay Bağışıklık Sistemi (AIS) kullanmışlardır.

Taha [45] yaptığı çalışmada uzaktan öğretim yöntemiyle öğrenim gören öğrenciler için bir ders öneri sistemi (CRS) tasarlamıştır. Yaptığı bu çalışmaya *CRS* ismini vermiştir. *CRS* bir XML tabanlı ortak filtreleme öneri sistemidir. Sistem bir uzaktan eğitim öğrencisine, öğrenciyle benzer ilgi alanlarına sahip ve akademik performansları örtüşen öğrencilerin başarılı oldukları dersleri önermeyi hedeflemiştir. *CRS*, her akademik bölüm için bir dizi ders niteliği tanımlamıştır. Ders nitelikleri, öğrencilerin dersten başarılı olması için sahip olması gereken özellikleridir. Taha'nın çalışması [45], aktif öğrencinin komşuluk sınıfında bulunan diğer öğrenciler tarafından başarılı olunan dersleri önerir. Yani, *CRS*, öneri listesi oluştururken hem aktif öğrencinin geçmişteki tercihlerini hem de bulunduğu komşuluğun derecelendirmelerini hesaba katar.

Pardos ve Weijie [46] daha önce mevcutta olan bir ders öneri sistemini farklı yaklaşımlarla geliştirmek üzere yeni bir öneri sistemi yaklaşımı sunmuşlardır. Bu öneri sistemi bir Model Tabanlı Ortak Filtreleme algoritmasının Tekrarlayan Sinir Ağları ile karma kullanımına bir örnek teşkil etmektedir. Buna göre mevcutta olan öneri sistemlerine iki adet metodu test ederek sonunda bunların karması olan üçüncü bir metot ortaya çıkarıp kullanmışlardır. Öğrencilerin belirlediği favori dersin benzerliğine dayanan önerileri çeşitlendirmek için kullanılan ders vektörü temsillerini öğrenmek için dokuz yıllık tarihi ders kayıt sırasına uygulanan geçmeli gram modelinde yeni bir değişiklik sunmuşlardır. Öneri sistemlerinde ders kataloglarındaki açıklama metinlerini de kullanarak sistemlerini iyileştirmişlerdir.

Diğer çalışmaların aksine bu tezde sunulan çalışma, öğrencilerden girdi olarak hiçbir bilgi almadan otomatikman bir ders öneri listesi sunar. Sistem Eskişehir Teknik

Üniversitesinin (ESTÜ) öğrenci bilişim sistemine entegre olarak çalışır. Sistemin çalışmasında herhangi bir birim ya da bölüm kısıtlaması yoktur. Çalışma üniversitenin tüm bölümlerinde öğrenim gören öğrenciler için öneri oluşturabilir. Bu sistem örgün öğrenci verileri üzerinde çalışır. Öneri sunulacak derslerle veya dersin öğretim elemanlarıyla ilgili özelliklerin önceden girilmiş olmasını gerektirmez. Öğrenciler arasındaki benzerliği aktif öğrenci ile aynı bölümde öğrenim görmüş olan üst sınıf ve mezun öğrencilerin transkriptleri üzerinde çalışarak bulur. Bu çalışma aktif öğrenciye bir ders öneri listesi sunarken öneri sunulan derslerden hangi harf notunu alabileceğini de tahmin eder. Bunlara ek olarak aktif öğrencinin seçmiş olduğu bir dersten hangi harf notunu alabileceğini de tahmin eder.



3. ORTAK FİLTRELEME TABANLI DERS ÖNERİ SİSTEMİ

İnternetin yaygınlaşması ile birlikte içerik üreten çevrim içi sistemlerin hayattaki yeri giderek artmıştır. İnternetteki içerik sayısı giderek yoğunlaştığından kullanıcıların istedikleri veriye ulaşmaları da gittikçe çaba sarf edilmesi gereken bir iş haline gelmeye başlamıştır. Aslında gelişen her teknoloji gibi internetin gelişmesi ve yaygınlaşması da insanların bilgiye erişiminin kolaylaşması için yapılmış bir mühendislik çözümüdür. Ancak veri yoğunluğu veriye erişimi gün geçtikçe zorlaştırmıştır. İşte tam da bu sebepten dolayı öneri sistemleri ortaya çıkmıştır. Öneri sistemleri kullanıcıların kişisel özelliklerini ortaya çıkararak onlara beğenebilecekleri ya da ilgilenebilecekleri ürünleri veya içerikleri önermeyi hedefler.

Akademik alanda da öğrencilerin bir öneri sistemine ihtiyaç duyduğu aşikârdır. Öğrenciler kayıtlı oldukları bölümde ya da diğer bölümlerden seçebilecekleri derslere karar verirken birçok ders arasından seçim yapmak durumunda kalarak kararsızlık yaşarlar. Öğrenciler ders seçimi yaparken başarılı olabilecekleri dersleri seçmek isterler. Ancak hangi dersin öğrencinin profiline uyduğuna karar vermek zor bir süreçtir. Geleneksel yöntemlerle aşılmaya çalışılan bu probleme, Ortak Filtreleme Tabanlı Ders Öneri Sistemi (OFTDÖS) ile çözüm önerisi sunulmuştur.

Bu çalışma öneri sistemleri çalışmalarını akademik ortama taşımaktadır. Bu çalışmada sunulan öneri sistemleri uygulaması, öğrenci olan kullanıcılarına, başarılı olabilecekleri dersleri öneri listesi olarak sunmaktadır. Buna ek olarak kullanıcılarına önerilen bu derslerden harf notu olarak kaç puan alacaklarını tahmin etmektedir. Bu öneri ve tahminleri üretmek için çeşitli öneri sistemleri algoritmaları ve yöntemlerini kullanmaktadır.

Çalışmada önerilerin üretilmesi ve harf notlarının tahmin edilmesi için gerçekleştirilen kullanıcı benzerlikleri, komşulukların tespit edilmesi gibi hesaplamaların tümü transkript verileri üzerinden hesaplanmaktadır. Transkript verileri içerisinde öğrencilerin alıp tamamladıkları dersler ve öğrencilerin o derslerden aldıkları puanlar ile ilgili bilgiler bulunmaktadır.

Geliştirilen sistem öğrencilere ders öneri listesi sunmak için aktif öğrencinin bulunduğu birimin ders kataloğunda bulunan seçmeli ders bilgilerini kullanır. Kullanıcılara önerilecek dersler katalogta bulunan seçmeli derslerden o birimdeki öğrencinin alabileceği dersler arasından seçilir.

Temel olarak ders önerisi listesi üretme algoritması aşağıda sunulan işlemleri sıralı olarak takip etmektedir;

1. Aktif kullanıcının kayıtlı olduğu birimde öğrenim gören, bitirmiş olduğu yarıyıl sayısı aktif öğrenciden büyük olan öğrencilerin ve mezunların transkriptlerindeki tüm ders bilgileri alınır.
2. Aktif öğrenci ile birimdeki diğer öğrencilerinin geçmişte ortak olarak aldığı dersler bulunur.
3. Benzerlik hesaplamaları yapılacak öğrenci listesi oluşturulur. Bu listeye deneysel olarak tespit edilmiş olan ortak ders eşiğini geçen öğrenciler ve mezunlar eklenir. Ortak ders sayısı, ortak ders eşliğinden az olan öğrenci ve mezunlar ise benzerlik hesaba katılmaz, bu yüzden listeye eklenmez.
4. Geçmişte alınmış ortak dersler üzerinden öğrencilerin benzerlik puanları Pearson Korelasyon katsayısı bulunarak hesaplanır.
5. Hesaplanan bu değerler öğrenciler ile ilişkilendirilecek bir listeye eklenir.
6. Listedeki öğrencilerden benzerlik puanı deneysel olarak tespit edilmiş olan benzerlik eşliğinden büyük olan kullanıcılar komşuluk oluşturulmak üzere filtrelenir.
7. Filtrelenmiş bu listeden de hedef dersi almamış öğrenciler filtrelenir. Hedef derse özel bir komşuluk ortaya çıkarılır.
8. Komşuluk içerisinde bulunan öğrencilerin hedef dersten aldığı notlar hesaplanan benzerlik oranlarıyla ağırlıklı ortalaması alınarak hedef ders için harf notu puanı tahmini üretilir.
9. Bu döngü aktif öğrencinin kataloğunda bulunan ve henüz almadığı seçmeli dersler için tekrar işlenir.
10. Başarılı olarak tahmin edilen dersler harf notu sıralaması yapılarak kullanıcıya öneri listesi halinde kullanıcıya sunulur.

Çalışma kullanıcılara diğer bir senaryo daha sunmaktadır. İkinci senaryoda, kullanıcının kendi belirttiği spesifik bir desten hangi harf notunu alacağına dair harf notu tahmini sunulur. Senaryonun algoritması öneri listesi algoritmasına benzer şekilde çalışmaktadır. Tespit edilen komşuluk içerisinde bulunan öğrenciler arasından hedef dersi alanlardan dersten aldıkları harf notu ile benzerlik puanı ağırlandırılarak ders için harf notu tahmini üretilir ve bu harf notu tahmini kullanıcıya sunulur.

Önerdiğimiz öneri sistemi Kullanıcı Tabanlı Ortak Filtreleme algoritmasını kullanmaktadır. Kullanıcılarının transkript verilerinden yola çıkarak benzerliğini hesaplar onlara ders öneri listesi sunar. Önerilen sistem Eskişehir Teknik Üniversitesi Öğrenci Bilişim Sistemi(ESTÜOBS) üzerinde çalışır.

ESTÜOBS öğrenci bilişim sistemi öğrencilerin dersleri, danışmanları, transkriptleri vb. gibi bilgilerini görebildikleri, kayıt yenileme dönemlerinde ders seçim işlemlerini gerçekleştirdikleri, danışmanları ve derslerini aldığı öğretim elemanları ile mesajlaşabildiği bir öğrenci bilişim sistemidir. Aynı zamanda akademik personeline de kullanabildiği bir sistemdir. Akademisyenler de yürüttükleri derslerle ilgili bilgilere erişebilir, sınav notları, yoklama, sınav yüzdeleri gibi bilgileri girebilirler ve akademik danışmanlık yaptığı öğrenciler ile iletişime geçebilmektedir. Aynı zamanda akademik danışmanlık işlemlerini kayıt yenileme dönemlerinde bu sistem üzerinden sunabilmektedirler.

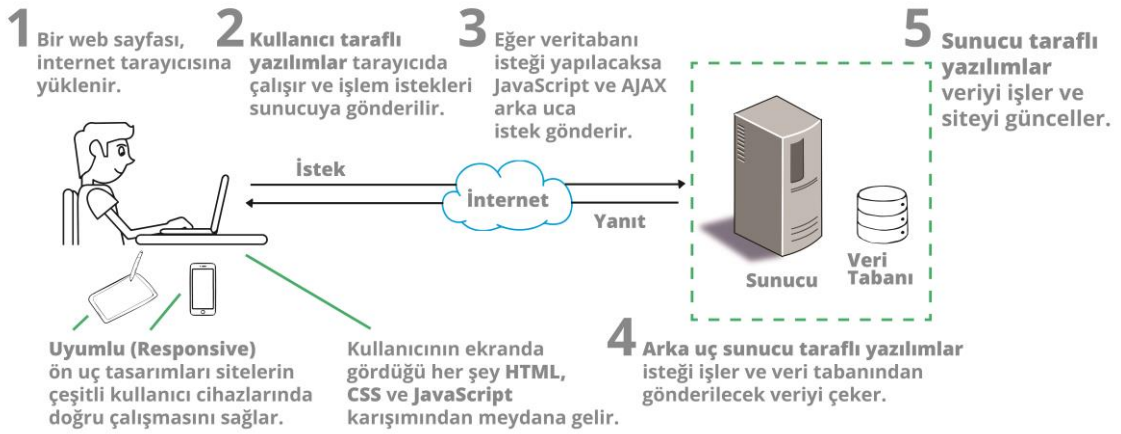
Bu çalışmada sunulan OFTDÖS ESTÜOBS ile bütünleşmiş çalışabilen bir modül olarak tasarlanmıştır. Öğrenciler sisteme giriş yaptıktan sonra “Öğrenci İşlemleri” aşağı açılır listesini açar(farenin imleci ilgili sekmenin üstüne geldiğinde liste otomatik olarak aşağıya doğru açılır). Açılan sekmeden “Ders Önerisi” alt menüsünü seçer. Sayfaya girildiğinde öneri üretim mekanizması tetiklenmiş olur ve sistem öneri üretmek üzere çalışmaya başlar. Çalışma sonucunda ekranda ders önerilerini gösterir liste öğrenciye tablo halinde sunulur. Sistemin ekran görüntüsü Şekil 3.1’de sunulmaktadır.

Ders Kodu	Ders Adı	Harf Notu
BIM492	Design Patterns	AA
KÜL199	Kültürel Etkinlikler	AA
ALM256	Almanca II	AA
BIM453	Introduction to Machine Learning	AB
BIM490	Introduction to Information Retrieval	AB
BIM309	Artificial Intelligence	BA
BIM470	Neural Networks	BB

Şekil 3.1. Ders Öneri Listesi Ekran Görüntüsü

4. KULLANILAN TEKNOLOJİLER

Bu tezde sunulan çalışmanın yazılımı geliştirilirken son teknolojilerin kullanılmasına özen gösterilmiştir. Uygulama bir web projesi olarak geliştirilmiştir. Aynı zamanda *ANASİS* uygulamasının içerisinde yer almıştır. Web uygulamalarında geliştirilen yazılımlar iki kategoriden oluşur. Kullanıcının uygulama ile iletişim kurduğu önyüz ve arka yüzde gerçekleşen tüm süreçler. Yani web uygulama yazılım mimarisi olarak ön uç ve arka uç olmak üzere iki kategoride incelenebilir. Ön yüz ve arka yüz katmanları ve çalışma mantığı Şekil 4.1’de sunulmaktadır. Bu çalışmada kullanılan ön uç teknolojileri Hiper Metin İşaret Dili versiyon 5 (HTML5), Basamaklı Stil Şablonları (CSS), AngularJS (AJS)’dir. Arka uçta ise Java programlama dilinin üzerinde Spring Web MVC çatısı kullanılmıştır. Bunların yanında veri tabanı sistemi olarak Microsoft SQL Server (MSSQL) teknolojisi kullanılmıştır.



Şekil 4.1. Ön yüz ve arka yüz katmanları ve çalışma prensibi

4.1. Ön Uç Teknolojileri

Ön uç teknolojileri birçok ön uç yazılım dili ve kütüphanelerinden meydana gelir. Farklı uygulamalar farklı dil ve kütüphanelerden oluşsa da, tüm web tarayıcıları tarafından derlenebilen sınırlı sayıda dil ve kütüphane vardır. Genel olarak tüm web tarayıcıları tarafından anlaşılan üç temel teknoloji *HTML5*, *CSS* ve *JavaScript(JS)*'tir. Bu üç temel teknoloji web sayfalarının temel çerçevesini oluştururlar. Diğer çoğu ön uç teknolojisi, bu üç ön uç teknolojisi temel alınarak ortaya çıkmıştır. Örneğin, bir web sayfasını otomobil gibi düşünebiliriz. Otomobilin şasisi *HTML* teknolojisini temsil eder. *CCS* ise otomobilin albenisini ortaya çıkaran rengi, jantları, tamponu ya da cam tavanıdır.

JS ise otomobilin hız sabitleme, multimedya oynatıcısı ya da otomatik vitesi gibi işlevsel teknolojileri olarak düşünülebilir.

Bu çalışmada *HTML* teknolojisi olarak *HTML5*, *CSS* teknolojisi olarak Leaner Stil Şablonları (*LESS*) ve *JS* teknolojisi olarak da *AJS* kullanılmıştır.

4.1.1. HTML5

HTML, belgelerin internet tarayıcılarında görüntülenebilmesini sağlamak için geliştirilmiş biçimlendirme dilidir. İnternet tarayıcıları web sunucularından aldıkları *HTML* dosyalarını çoklu medya sayfalarına dönüştürüp kullanıcılara sunarlar.

HTML sayfalarının içeriği *HTML* etiketlerinden oluşur. Bu etiketler sayesinde internet tarayıcıları çeşitli içerikleri gösterebilirler. *HTML* etiketleri sayesinde web sayfalarında görseller, başlıklar, paragraflar, sıralı-sırasız listeler ve formlar gösterilebilir. *HTML* etiketleri büyüktür küçüktür işaretleri kullanarak kapsamları arasında içeriklerini alırlar. Örneğin bir paragraf yazı göstermek için `<p>` ve `</p>` arasına metinsel ifade girilir. Bazı özel etiketlerse kapsamsız olarak tanımlanır. Kapsamı olmayan etiketler içerisinde gösterecekleri içerikleri etiketleri özellikleri ile alırlar. Bunlara en iyi örnek web sayfasında görsel göstermeye yarayan `` etiketidir. Bu tür etiketler *html* uzantılı dosyalarda yazılmış olsa da internet tarayıcıları bu etiketleri göstermez. Bu etiketleri yorumlayarak ekran içerikleri gösterirler [47].

HTML5 ilk olarak 2008 yılında tanıtılmış olsa da 2014 yılında kullanıma sunulmuştur. *HTML*'in 5. sürümü ile birlikte tüm tarayıcılarda çalışır hale gelmiştir. Bu *HTML5*'in en önemli özelliğidir. Diğer gelen yeni özellikler ise video ve ses dosyaları gibi çoklu medya kullanımına olanak sağlamaktadır. *HTML5* sayesinde kullanıcının izni dâhilinde kullanıcının konumuna erişilebilir. Düşük işlem gücü gerektirdiğinden mobil platformlarda web sayfaları rahatça çalıştırılabilir [48].

Bu çalışmada *HTML5*, mobil platformlarda çalıştırılabilme, düşük işlem gücü gereksinimi ve tüm internet tarayıcılarının desteğini alabilmek için kullanılmıştır.

4.1.2. Basamaklı Stil Şablonları ve LESS

CSS, *HTML* gibi işaretleme dilleriyle yazılmış dokümanların görsel olarak sunumunu tanımlayan bir stil sayfası dilidir. *CSS* dünya çapında “World Wide Web Consortium (WWW)” olarak bilinen internet teknolojisinin ana teknolojilerinden biridir. *CSS*, web sayfalarının içerik ve yerleşim düzenlemelerinin yapılabilmesi için

tasarlanmıştır. Bu teknoloji sayesinde oluşturulan *HTML* gibi işaretleme diliyle oluşturulmuş dokümanlara ve içeriklerine renkler, yerleşim düzenlemeleri gibi düzenler, yazı fontları ve çeşitli hareketli elemanlar, *CSS* sayesinde tanımlanabilir. Bu sayede web sayfalarının içeriklerin erişilebilirlikleri artar, görsel tasarımlarında esneklik ve kontrol sağlanmış olur. Basamaklı Stil Şablonlarında birden fazla kuralın bir etikete uygulanması mümkündür. Örneğin, bir *HTML* sayfasında yazılmış bir paragraf etiketine birden fazla *CSS* kuralı uygulanabiliyor olabilir. Basamaklı kelimesi, bu gibi durumlarda hangi stil kuralının öncelikli olarak uygulanacağını belirlenmesi basamaklı bir öncelik sırasından gelmekte olduğu için kullanılmaktadır. *CSS*, *WWW* tarafından geliştirilmekte ve hakları bu grup tarafından sahiplenilmektedir [49].

Genelde büyük projelerde *CSS* sınıfları tek başlarına kullanılmazlar. Bunun yerine bir stil şablonu ön işleyicisi ile birlikte kullanılırlar. Bunun avantajları dinamik bir *CSS* yapısı oluşturmaları, kod karmaşası ve tekrarının önlenmesidir.

Bu çalışmada *CSS* teknolojileri *LESS* ile birlikte kullanılmıştır. *LESS*, *CSS* sayfalarına derlenebilen, sunucu ya da istemci tarafında çalıştırılabilen, *JS* tabanlı dinamik bir *CSS* ön işleyicisidir [50]. *LESS* projesi açık kaynak kodludur. *LESS*'in bu projede kullanılmasının ana sebebi, sunucu üzerindeki yükün azaltmasına yardımcı olacak şekilde kullanıcı tarafında gerçek zamanlı derlenmesidir.

4.1.3. JavaScript ve AngularJS

JS çoğunlukla kullanıcı tarafı teknolojiler üzerinde çalışan betik bir programlama dilidir. *HTML* ve *CSS* ile birlikte *WWW*'nun temel teknolojilerini oluştururlar. *JS* kullanıcılarla etkileşimli web sayfaları oluşturmanın önemli bir parçasıdır [51]. *HTML* 'in içerisinde tanımlanırlar ve tanımlandıkları *HTML* sayfalarında çalışırlar. Kullanıcının bilgisayarındaki erişimleri oldukça kısıtlıdır. Direkt olarak bilgisayarın hafızasına ya da işlemcisine erişim izni yoktur. Bunun yerine etkileşimini web tarayıcısı üzerinde çalışarak gerçekleştirir. Oldukça hafiftir [52]. Mobil cihazlar da dâhil olmak üzere hemen hemen bütün cihazlarda kullanılırlar.

Bu çalışmada bir *JS* çerçevesi olan *AJS* kullanılmıştır. *AJS* Google tarafından geliştirilen bir yapısal çerçevedir. *HTML* sözdiziminin genişletilmesine olanak sağlar. Bağımlılık enjeksiyonu ve iki yönlü veri bağlama özellikleri sayesinde uygulamalardaki kod karmaşıklığı ve kalabalıklığının önüne geçer [53].

Bu çalışmada yapılan uygulamanın kullanıcı tarafında hızlı çalışması ve dinamik olabilmesi için *AJS* teknolojisi kullanılmıştır.

4.2. Arka Uç Teknolojileri

Arka uç teknolojileri, bir web uygulamasında ön uca veri sağlayan teknolojilerdir. Sunucular, veri tabanları, uygulama programı ara yüzleri ve programlama dillerinin birleşiminden oluşur. Sunucu işlem gücü yüksek üzerinde veri tabanı ve arka uç yazılımını çalıştıran bir bilgisayardır. Yazılım ise bu ikisi arasında bağlantıyı kuran teknolojidir. Bu çalışmada sunucu tarafı yazılım dili olarak Java ve onun bir çerçevesi olan Spring Web MVC, veri tabanı olarak da *MSSQL* kullanılmıştır.

4.2.1. Java ve Spring Web MVC

Java açık kaynak kodlu bir sunucu tarafı nesne yönelimli bir programlama dilidir. Java'yı diğer dillere göre üstün kılan birçok yönü mevcuttur. Java programlama dili platform bağımsız, yüksek başarımlı, güvenilir ve yüksek performanslıdır [54]. Java'nın platform bağımsızlığı ya da mimari yapıdan bağımsızlığı üzerinde koştuğu Java Sanal Makinasından (JVM) kaynaklanmaktadır. Her platform için en az bir adet *JVM* sunulmuştur. Java kodları derlendikten sonra ortaya çıkardığı *bytecode* denen dosyaları *JVM* üzerinde çalıştırdığı için her platformda o platforma özgü kod yazmaya gerek kalmadan çalışabilmektedir [55]. Bu Java'nın çıkış noktası olan bir kere yaz her yerde çalıştır mantığını desteklemektedir. Java standart sürümünden ayrı olarak bir kurumsal sürümüne de sahiptir. Java kurumsal sürüm büyük ölçekli projelerde kullanılmaktadır. Ölçeklenebilir olması, güvenilirliği ve güvenlik yapısı çok katmanlı uygulamalarda başarılı olmasını sağlamaktadır [56].

Spring Web MVC çerçevesi Java için geliştirilmiş bir yazılım çerçevesidir. Faydalı ekleri sayesinde Java web uygulamalarını daha kullanışlı hale getirmektedir. Obje oluşturma, web servisleri sağlama, güvenliği kontrol altına alma, nesnelerin rollerinin açıkça ayrılması gibi teknik manada birçok avantaja sahiptir [57].

4.2.2. Microsoft SQL Server

Bilgisayar sistemleri verileri üzerinde çalışmalarını için tasarlanırlar. Kullanıcıları ile ilgili ve hizmet sundukları alanla ilgili veriler üzerinde işlemlerini gerçekleştirirler. Bu sebeple sahip oldukları verileri saklama ihtiyacı duyarlar. Veriler veri tabanlarında saklanırlar. Verilerin istendiğinde bir mantık çerçevesinde ulaşılabilmesi için veriler

arasında ilişkiler bulunur. Veriler ve ilişkilerin saklandığı veri tabanı yönetim sistemlerine ilişkisel veri tabanı yönetim sistemi (VTYS) denir.

Piyasa da kullanılan birçok ilişkisel VTYS mevcuttur. *MSSQL* de bu sistemlerden biridir. *MSSQL* Microsoft firması tarafından geliştirilen bir VTYS'dir. *MSSQL* genelde büyük ölçekli projelerde bilgisayar programları tarafından kullanılmak üzere geliştirilmiştir. *MSSQL* aynı sunucuda çalışılan programlar tarafından çalıştırılabileceği gibi uzak sunucularda çalıştırılan programlar tarafından da çalıştırılabilir [58]. Eskişehir Teknik Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi (*BAUM*) tarafından *MSSQL* kullanıldığı için bu çalışmada da *MSSQL* kullanılmıştır.



5. DENEYSEL ÇALIŞMALAR

Bu bölümde, önerilen *OF* tabanlı ders öneri sisteminin performansını ölçülemek için gerçek veriler üzerinde yapılan deneylere ait ayrıntılı bilgiler sunulmuştur. Bölün devamı şu şekildedir: Bölüm 5.1’de yapılan deneylerde kullanılan veri kümesi ile ilgili bilgiler sunulmuştur. Bölüm 5.2’de sistemin ürettiği önerilerin değerlendirme ölçütleri, Bölüm 5.3’te ise deney çalışmalarında belirlenen yöntem sunulmuştur. Son olarak deneysel çalışmaları sonuçları ve analizi Bölüm 5.4’te sunulmuştur.

5.1. Veri Kümesi

Önerilen ortak filtreleme tabanlı ders öneri sisteminin öneri üretme başarısının analizi için *ESTÜBM*’nin 2000 – 2013 yılları arasındaki öğrenci ve mezunların bilgileri veri setini oluşturmuştur. Bu bilgi öğrenci ve mezunların transkript bilgilerini içermektedir. Transkript bilgileri hali hazırda aktif olarak öğrenci olan ve mezun olmuş öğrencilerin geçmişte almış oldukları derslere ilişkin verileri saklamaktadır. *ESTÜBM*’de öğrenim gören bir öğrenciye ait örnek transkript verisi Tablo 5.1’de sunulmaktadır.

Tablo 5.1. *Eskişehir Teknik Üniversitesi Bilgisayar Mühendisliği Örnek Öğrenci Transkript Verisi*

Yıl	Dönem	Ders Kodu	Ders Adı	ECTS	Not	Kredi*Not
2010	Güz	BİM213	Data Str. and Alg.	5.5	AA	22.00
2010	Güz	BİL200	Computer Prog.	6.0	AA	26.00
2010	Bahar	BİM312	Database Man. Sys.	6.0	AB	22.20
2011	Güz	EEM336	Microprocessors	7.0	BA	23.10

Öneri sisteminde öneri üretme sürecinde kullanılan derecelendirme verisi transkript verisindeki harf notuna karşılık gelmektedir. Yani aslında sistem kullanıcılardan derecelendirmeler açık bir şekilde değil de örtülü olarak elde edilmektedir. Öğrencilerin ve mezunların aldıkları harf notu verisi veri setinde harf olarak saklanmaktadır. Harf notları 22 adet değere sahiptir. Ancak bunlardan 11 tanesi (AA, AB, BA, BB, BC, CB, CC, CD, DC, DD, FF) derecelendirme için kullanılabilir. Diğer 11 harf notu ise devamsızlık, çekilme, eksik vb. gibi durumlarda kullanılmaktadır. Bu özel durumlar veri seti dışında bırakılmıştır. Derecelendirme olarak kullanılacak harf notları ise rakamsal derecelendirmelere çevrilmiştir. Derecelendirme de kullanılan harf notlarına karşılık gelen değerler Tablo 5.2’de sunulmaktadır.

Tablo 5.2. Harf notları ve karşılık gelen derecelendirme puanları

Harf Notu	Derecelendirme Puanı
AA	10
AB	9
BA	8
BB	7
BC	6
CB	5
CC	4
CD	3
DC	2
DD	1
FF	0

Transkript verileri öğrenciler dersleri tamamladıkça oluşmaktadır. Yani her dönem sonunda öğrencilerin derslerden aldıkları harf notları belli olduktan sonra aldıkları derslere ait bilgiler veri tabanındaki transkript tablosuna aktarılmaktadır. Bundan dolayı bu çalışmada sunulan öneri sisteminin transkript verilerinden oluşan veri seti boşluklu yapıda değildir. Deneylerde *ESTÜBM* veri setinin tamamı test veri seti olarak kullanılmıştır. Bu veri setinin içerdiği bilgiler ve deney sonucundan önerilen toplam ders sayıları Tablo 5.3’de sunulmaktadır.

Tablo 5.3. Veri Seti ile İlgili Bilgiler

Öğrenci/Mezun Sayısı	Seçmeli Ders Sayısı	Ders Önerisi Sayısı
338	30	9615

5.2. Değerlendirme Ölçütleri

Yapılan çalışmaların ne kadar doğru sonuçlar ürettiklerini tespit edebilmek için çeşitli değerlendirme ölçütleri kullanılmaktadır. Bu çalışmada, yapılan sistemin ürettiği önerilerin doğruluğunun ölçülmesi için ortalama mutlak hata (OMH) metriği kullanılmıştır. *OMH* yöntemi iki sürekli değişken arasındaki farkı ölçer. X ve Y ’nin aynı ürünün farklı derecelendirme değişkenleri olduğu ve Y derecelendirmesinin X derecelendirmesi için tahmin edilen bir derecelendirme olduğu varsayılınsın. Spesifik bir ürün olan i için gerçekte olan X_i derecelendirmesine Y_i tahmini yapılırsa ve bir koordinat düzleminde bu iki noktanın gösterildiği düşünülürse, *OMH* bu iki nokta ile $X=Y$ çizgisi arasındaki ortalama dikey mesafedir. *OMH* formülü Denklem 5.1’de sunulmaktadır.

$$\text{Ortalama Mutlak Hata} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n} \quad (5.1)$$

OMH tüm mutlak hataların ortalamasıdır. Yani sistemin ürettiği tüm tahminlerin gerçek değerden olan mutlak hatalarının üretilen tahmin sayısına bölümüyle bulunur. Denklem 5.1’de y_i tahmin edilen derecelendirmeyi, x_i ise gerçekte var olan derecelendirmeyi temsil etmektedir. $|e_i|$ ise tahmin edilen değer gerçekte var olandan mutlak farkını temsil eder. *OMH* değeri ne kadar küçük ise üretilen önerilerin doğruluğu o kadar yüksektir. Deneysel gerçekleştirilirken çeşitli parametreler değiştirilerek sonuçlar üzerindeki etkisi izlenmiştir. *OMH* değerlendirme ölçütü kullanılarak üretilen ders öneri listelerinin ve tahmin edilen harf notlarının başarısı gözlemlenmiştir.

5.3. Metodoloji

Bu tez çalışmasında tüm *ESTÜBM* transkript veri setinin tamamı test verisi olarak kullanılmıştır. Veri setindeki öğrenciler ve mezunların tamamının transkriptleri kullanılmıştır. Transkript verileri üzerinde var olan derslere ait harf notları için tahminde bulunulmuştur. Başlangıç olarak herhangi bir öğrenci aktif öğrenci olarak seçilmiştir. Aktif öğrencinin transkriptinde bulunan seçmeli dersler harf notu tahmini üretmek üzere hedef ders olarak seçilmiştir. Seçilen hedef dersin transkriptte var olan harf notu silinerek deney tablosuna kaydedilmiştir. Deney tablosunda öğrenci numarası, ders kodu, dersin gerçek harf notuna karşılık gelen derecelendirme ve sistemin tahmin ettiği derecelendirme kaydedilmiştir. Transkriptten çıkartılan hedef dersin harf notu tahmin edilmiştir. Tahmin işlemini gerçekleştirmek için aktif öğrenciye benzer diğer öğrenciler veri setinden bulunmuştur. Benzerlik bulma işlemi için öncelikle veri setinden hedef dersi almış olan öğrenciler filtrelenmiştir. Aktif öğrenci ile diğer öğrenciler arasındaki benzerlik Pearson korelasyon katsayısı yöntemi kullanılarak tespit edilmiştir. Benzerlik tespit etmek için öncelikle aktif öğrenci ile teker teker diğer öğrencilerin aldıkları ortak dersler tespit edilmiştir. Daha sonra ortak derslerden aldıkları harf notları karşılık gelen derecelendirmelere dönüştürülerek Pearson korelasyonları hesaplanmıştır. Böylece ortaya aktif öğrenciler ve diğer öğrenciler arasında bulunan benzerlik değerleri çıkmıştır.

5.3.1. Komşuluk seçme ve öneri üretme yöntemi

Aktif öğrenci ve diğer öğrenciler arasında bulunan benzerlik değerleri kullanılarak iki farklı yöntemle komşuluklar belirlenmiştir. İki yöntem de ayrı ayrı deneylere tabi

tutulurak çıkan sonuçlar incelenmiştir. Diğerine göre daha başarılı olan yöntem komşuluk belirleme yöntemi olarak kabul edilmiştir. Bu iki yöntemden ilki K en yakın komşu (kNN), ikincisi ise benzerlik eşiği yöntemidir.

5.3.1.1. K en yakın komşu yöntemi

Bu yöntem OF algoritmalarıyla birlikte kullanılan en çok kullanılan yöntemlerden birisidir. Bu yöntemde komşuluklar tespit edilirken aktif kullanıcı ile diğer kullanıcılar arasında hesaplanan benzerlik değerleri kullanılır. Hesaplanan bu benzerlik değerlerine göre aktif kullanıcıya en çok benzeyen K adet kullanıcı komşu olarak seçilir.

Bu çalışmada yapılan deneylerde bu yöntem test edilirken K değeri 2 ile 200 arasında değiştirilerek sonuçlar üzerindeki etkisi gözlemlenmiştir.

5.3.1.2. Benzerlik eşiği yöntemi

Bu yöntem de kNN yöntemi gibi popüler OF komşuluk seçme yöntemlerinden biridir. Aktif kullanıcı ile diğer kullanıcılar arasında hesaplanan benzerlik değerleri belirlenir. Bu değerler belirlendikten sonra benzerlik değeri için bir eşik değeri tanımlanır. Tanımlanan bu eşik değerinin üstünde benzerlik değerine sahip olan kullanıcılar aktif kullanıcının komşuları arasına eklenir. Diğer yöntemin aksine bu yöntemde bir komşu sayısı kısıtlaması yoktur. İlgili eşik değerini geçen her kullanıcıya aktif kullanıcının komşusu muamelesi yapılır.

5.3.1.3. Öneri üretimi

Aktif öğrenciye harf notu tahmini üretilebilmesi için aktif kullanıcının komşularının verileri kullanılmıştır. Aktif kullanıcı ile diğer kullanıcılar arasında tespit edilen komşuluk içerisinde bulunan öğrencilerin hedef dersten aldıkları derecelendirme ile benzerlikleri ağırlıklandırılarak bir harf notu tahmini üretilmiştir.

Aktif öğrencinin transkriptinde bulunan tüm dersler için bu yöntem tekrar edilmiştir. Böylece aktif kullanıcının gerçek aldığı ve bir harf notuna sahip tüm dersleri için harf notu tahminleri üretilmiştir. Tüm dersler için tahmin üretme işlemi bittikten sonra yazar tarafından başarılı olarak nitelendirilen dersler öğrenciye öneri listesi olarak sunulmuştur. Bu çalışmaya göre başarılı olan harf notları AA, AB, BA ve BB'dir.

Tüm bu yöntemler test edilirken en yakın k öğrenci değeri, ortak ders eşiği ve benzerlik eşiği değerleri değiştirilmiştir. En yakın n öğrenci faktörünün değeri test edilirken, n sayısı belirli bir gösterge çizelgesi çerçevesinde 2 ile 200 arasında

değiştirilmiştir. Bu sırada ortak ders ve benzerlik eşikleri sabit tutulmuştur. Benzer şekilde benzerlik eşiği faktörü test edilirken, Pearson Korelasyon katsayısı ile hesaplanan benzerlik değeri -0,9 (negatif Korelasyon) ile 0,99 (pozitif Korelasyon) arasında değiştirilmiş ve sonuçlar incelenmiştir.

Öğrenciler/mezunlar için ders öneri listeleri ve harf notu tahminleri üretilmiştir.

5.4. Deneysel Sonuçları ve Analizi

Sunulan ders önerilerinin doğruluğunun gözlemlenmesi için kullanılan faktörlerin aldığı değerler üzerinde değişiklik yapılarak sonuçlar izlenmiştir. Bu deneylerde, benzerlik eşiği, ortak ders eşiği, en *kNN* eşiği değerleri test edilmiştir. Bu faktörlerin değerleri değiştirilerek gerçekte var olan harf notları sistem tarafından tahmin edilmiştir. *OMH* ölçütü kullanılarak yapılan tahminlerin doğruluğu gözlemlenmiştir. Bu ölçüt göz önünde bulundurularak yukarıda belirtilen faktörlerin optimum değerleri tespit edilmiştir. Ayrıca deneylerde öneri sunulabilen ders sayısı, öğrenci sayısı ve hata oranları da gözlemlenmiştir.

5.4.1. Ortak ders eşiği deneyleri

Bu deneyde aktif öğrenci ile diğer öğrencilerin geçmişte aldıkları ortak derslerin sayısının alt eşiği belirlenmek üzere deneyler yapılmıştır. Deneyler yapılırken ortak ders sayısı alt eşiği değeri değiştirilmiş, diğer faktörler ise sabit tutulmuştur. Tablo 5.4 ve Tablo 5.5'te ortak ders sayısı alt eşiği değeri 5 olarak sabitlenmiştir, *kNN* sabiti ve benzerlik eşiği değeri değiştirilmiştir. Tablo 5.6 ve Tablo 5.7'de ise ortak ders sayısı alt eşiği 3 olarak sabitlenmiş, *kNN* sabiti ve benzerlik eşiği değeri değiştirilmiştir. Deney sonuçlarında öneri sunulabilen ders sayısı, öğrenci sayısı ve hata oranları gözlemlenmiştir.

Tablo 5.4. *K* komşuluk değerleri için deney sonuçları(Ortak ders sayısı eşiği 5)

K	Ders Önerisi Sayısı	Kapsama	OMH
5	9604	329/338	1,5017
10	9604	329/338	1,5024
15	9604	329/338	1,5259
20	9604	329/338	1,5565
25	9604	329/338	1,5903
50	9604	329/338	1,7073
100	9604	329/338	1,8599
200	9604	329/338	2,0130

Tablo 5.5. *Benzerlik Eşiği* için deney sonuçları(Ortak ders sayısı eşiği 5)

Benzerlik Eşiği	Ders Önerisi Sayısı	Kapsama	OMH
-0,9	9561	324/338	2,1232
0	9559	324/338	2,0495
0,1	9556	324/338	2,0436
0,5	9399	324/338	1,7704
0,8	6438	309/338	1,3494
0,85	5100	288/338	1,3002
0,9	3020	236/338	1,2676
0,95	1207	150/338	1,2037
0,99	213	42/338	1,9454

Tablo 5.6. *K* komşuluk değerleri için deney sonuçları(Ortak ders sayısı eşiği 3)

K	Ders Önerisi Sayısı	Kapsama	OMH
5	9606	330/338	1,5149
10	9606	330/338	1,5141
15	9606	330/338	1,5348
20	9606	330/338	1,5662
25	9606	330/338	1,5995
50	9606	330/338	1,7135
100	9606	330/338	1,8607
200	9606	330/338	2,0241

Tablo 5.7. Benzerlik eşiği için deney sonuçları(Ortak ders sayısı eşiği 3)

Benzerlik Eşiği	Ders Önerisi Sayısı	Kapsama	OMH
-0,9	9563	325/338	2,1142
0	9561	325/338	2,0551
0,1	9558	325/338	2,0473
0,5	9401	325/338	1,7763
0,8	6598	321/338	1,3826
0,85	5390	317/338	1,3594
0,9	3573	312/338	1,4145
0,95	1910	267/338	1,5343
0,99	636	128/338	1,7662

Ortak ders sayısının bulunabilmesi için yapılan deneylerin sonuçlarına göre öneri sunulabilecek derslerin ve öğrencilerin sayısını artırmak için ortak ders sayısının alt eşiğinin olabildiğince küçük tutulması gerektiği sonucuna varılmıştır. Bu sebeple deneylere ortak ders sayısının 1 değerinde sabitlenerek devam edilmesine karar verilmiştir. Ortak ders sayısı 1 olarak sabitlendikten sonra, komşulukların oluşturulması için *kNN* ve benzerlik eşiği yöntemlerinden hangisinin kullanılması gerektiğine karar verilmesi gerekmektedir. Bu kararın verilebilmesi için deneylere devam edilmiştir. Buna göre ilerleyen deneylerde ortak ders sayısı 1 de sabitlenerek *kNN* ve benzerlik eşiği yöntemlerinin sonuçlara etkisi incelenmiştir.

5.4.2. Komşuluk belirleme yöntemi deneyleri

Komşulukların belirlenebilmesi için iki adet yöntemden birisinin seçilmesi gerekmektedir. Bu kararı verebilmek için ortak ders sayısı alt eşiği 1 değerinde sabitlendikten sonra iki yöntem test edilmiştir. Deney sonuçları Tablo 5.8 ve Tablo 5.9’da sunulmaktadır.

Tablo 5.8. *K Komşuluk Değerleri için deney sonuçları*

K	Ders Önerisi Sayısı	Kapsama	OMH
2	9602	331/338	1,66
3	9615	331/338	1,51
5	9628	331/338	1,55
10	9641	331/338	1,54
15	9654	331/338	1,56
20	9667	331/338	1,59
25	9680	331/338	1,73
50	9693	331/338	1,84
100	9706	331/338	1,99
200	9719	331/338	2,13

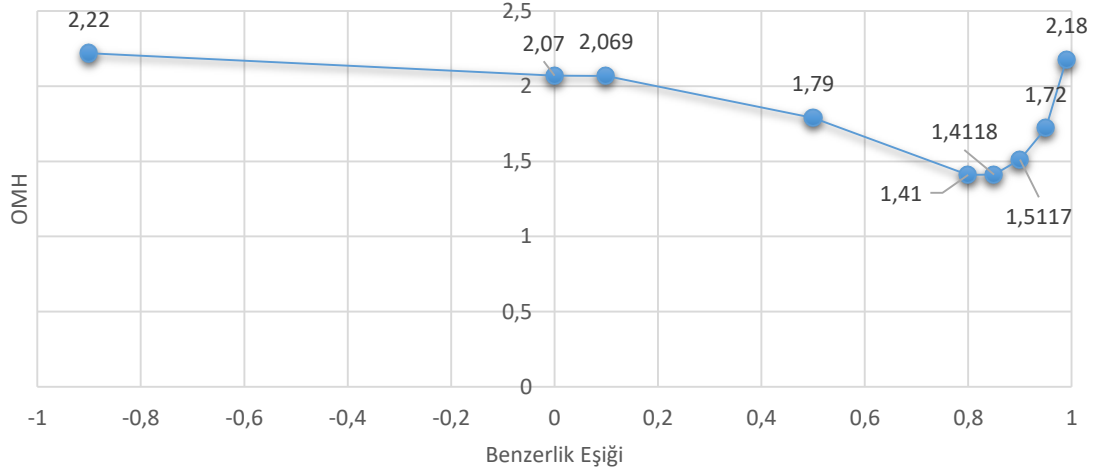
Tablo 5.8’deki sonuçlara göre komşuluk değerine karşılık gelen OMH değerleri incelenmiştir. Komşuluk değeri arttıkça *OMH* değerinin de arttığı gözlemlenmiştir. Başlangıç noktasında komşuluk sayısı arttıkça 3 adet komşuya kadar bir iyileşme gözlemlenirken, komşuluk değeri 5 ve üzerine çıktıkça ortaya çıkan *OMH* değeri de artmıştır. Öneri sunulan öğrenci sayısında bir değişim gözlenmemiştir. En başarılı önerinin 1,51 *OMH* değeri ile sunulabildiği gözlenmiştir. Komşuluk değerine karşılık gelen önerilen ders sayısı tablo sonuçlarında incelenmiştir. Komşuluk değeri artmasıyla *OMH*’nin de giderek artmasına rağmen öneri sunulan ders sayısında artış gözlenmiştir. Benzerlik eşiği yönteminin değerlendirilmesi için deneylere benzerlik eşiği ile devam edilmiştir.

Benzerlik eşiği deneyinde Pearson Korelasyon katsayısı ile hesaplanan benzerlik değerleri bir eşik olarak tanımlanmış ve değişken olarak uygulanmıştır. Bu değişkenin aldığı değere göre aktif kullanıcı ile diğer kullanıcılar arasındaki komşuluklar belirlenmiş ve deneylere devam edilmiştir. Daha önceki deneyde olduğu gibi ortak ders eşiği 1 seçilmiştir. Deneylerde önerilen ders sayısı, öneri üretilen öğrenci sayısı ve *OMH* değerleri incelenmiştir. Tablo 5.9’de deney sonuçları sunulmaktadır.

Tablo 5.9. *Benzerlik Eşiği Deney Sonuçları*

Benzerlik Eşiği	Ders Önerisi Sayısı	Kapsama	OMH
-0,9	9574	325/338	2,2200
0	9572	325/338	2,0700
0,1	9569	325/338	2,0690
0,5	9421	325/338	1,7900
0,8	6603	325/338	1,4100
0,85	5391	322/338	1,4118
0,9	3765	318/338	1,5117
0,95	2201	294/338	1,7200
0,99	1064	210/338	2,1800

Deney sonuçlarının daha iyi anlaşılabilmesi için şekil 5.1’de ise benzerlik eşiğinin değişimine karşılık, *OMH*’nin değişimi gözlemlenmiştir. Benzerlik eşiği arttırıldıkça, ders önerisi sayısında ciddi bir düşüş söz konusudur. Tablo 5.9’ dan da anlaşılacağı üzere benzerlik eşiği arttırıldıkça, ders önerilebilen öğrenci sayısı da azalmaktadır.



Şekil 5.1 Benzerlik Eşiğinin Değişimine Karşılık, *OMH*’nin Değişimi

Deney sonuçları incelendiğinde *kNN* değerinin en başarılı *OMH* değerinin 1.51 olduğu gözlenmiştir. Bu *OMH* değeri *kNN* değeri 3 olduğunda alınmıştır ve 9615 adet ders önerilebilmiştir. Diğer yöntem olan benzerlik eşiği yöntemin deney sonuçları incelendiğinde ise en başarılı önerinin 1.41 *OMH* değeriyle sunulduğu gözlemlenmiştir. Bu *OMH* değeri benzerlik eşiği değeri 0,8 olduğunda alınmıştır ve toplamda 6605 adet ders önerilebilmiştir.

Bu deney sonuçları ışığında ders öneri sisteminde benzerlik eşiği yönteminin kullanılmasına karar verilmiştir. Benzerlik eşik değerinin 0,8 değerinde, ortak ders eşiğinin ise 1 değerinde sabitlenmesi uygun görülmüştür. Yapılan deneyler optimum *OMH* sonucunun 1,41 olduğunu göstermektedir. Yani bir öğrenciye ders önerisi sunulduğunda derslerin tahmin edilen harf notları maksimum 1 harf notu hata sapmasına sahip olacaktır. Örneğin sistem bir öğrenciye ders önerisi sunduğunda, aktif öğrencinin o dersten BB notunu alacağını tahmin etmişse öğrenci gerçek hayatta o dersten en fazla 1 harf notu uzakta bir not almış demektir. Dolayısıyla sistem BB önerdiğinde öğrenci ya gerçekten BB notu almıştır ya da BA ve BC notlarından birini almıştır. Sonuçlara bu çalışmanın başarılı olduğunu göstermektedir.

6. SONUÇLAR VE DEĞERLENDİRME

Günümüzde öneri sistemleri hayatımızın çoğu alanında karşımıza çıkmaktadır. Bu sistemler gün geçtikçe gelişmektedir. Gelişen bu sistemler hayatımızın her alanına dokunmaya başladığı için artık bir ihtiyaç haline gelmektedir. Bu ihtiyaç e-ticaret, sosyal medya gibi alanlarda sıkça rastlansa da artık akademik camiada da ihtiyaç haline gelmiştir. Öğrenciler ders seçim dönemlerinde başarılı olabilecekleri dersleri seçmek isterler. Bu seçimi yapabilmeleri içinde bir akademik yönlendirmeye ihtiyaç duyarlar.

Akademik yönlendirme hem danışmanlar hem de derslerini doğru seçmeyi planlayan öğrenciler arasında her zaman büyük bir problem olmuştur. Öğrenciler hangi alanda başarılı olacaklarını her zaman kestiremeyebilirler. Dolayısıyla kendi başarı alanlarında ders seçemedikleri için de akademik olarak başarısız olabilirler. Diğer yandan akademik danışmanlık yapan akademisyenler ise öğrencilerin akademik profilini doğru tespit edemeyebilirler. Bu sebeple öğrencileri akademik profillerinin gerektirdiklerine göre yönlendiremeyebilirler. Bu durum öğrencileri akademik başarısızlığa sürükler.

Bu tezde önerilen yöntem öğrencilerin akademik başarılarına göre onlara ders önerileri sunar. Aktif öğrenci ile aynı bölümde öğrenim gören diğer öğrencilerin daha önce almış oldukları puanlar üzerinden harf notu tahmini yapar. Aktif öğrencinin akademik profiline benzer öğrenciler bulur ve bulunan bu benzer öğrencilerin aldıkları dersler arasından aktif öğrenciye ders öneri listesi sunulur. Bunun yanı sıra öğrenci almak istediği bir dersten hangi harf notunu alabileceğinin tahminini de bu yöntem aracılığıyla öğrenebilir. Ampirik analizler, önerilen yöntemle, ders seçmek isteyen öğrencilere başarılı bir şekilde ders öneri listesi sunulabileceğini ve harf notu tahmini yapılabileceğini göstermiştir.

Gelecek çalışmada olarak öğrenci verilerinin yanı sıra derslerle ilgili haftalık ders programı, teorik ve uygulamalı ders saatleri, ders ile ilgili verilen ödevler ve yapılan sınavların tamamı üzerinden elde edilen veriler üzerinde çalışılması planlanmaktadır.

KAYNAKÇA

- [1] Resnick P., & Varian H. (1997): ‘Recommender Systems’, *Communications of the ACM*, vol. 40,no. 3, March 1997, pp. 56–58.
- [2] Vaughan-Nichols, S. J. (1997). *Switching to a faster Internet*. *Computer*, 30(1), 31-32.
- [3] Resnick, Paul & Iacovou, Neophytos & Suchak, Mitesh & Bergstrom, Peter & Riedl, John. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. 10.1145/192844.192905.
- [4] Mahmood, T. and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, New York, USA: ACM*, 73-82.
- [5] Burke, R. (2007). Hybrid Web recommender systems. *The adaptive web. Springer Berlin Heidelberg*. 4321(1), 377-408.
- [6] Resnick, P. and Varian, H. R. (1997). Recommender Systems. *Communications of the ACM*, 40(3), 56-58.
- [7] Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1), 76-80.
- [8] Jacobson, K., Murali, V., Newett, E., Whitman, B., & Yon, R. (2016, September). Music personalization at spotify. *In Proceedings of the 10th ACM Conference on Recommender Systems (pp. 373-373)*. *ACM*.
- [9] Bennett, J., & Lanning, S. (2007, August). The netflix prize. *In Proceedings of KDD cup and workshop (Vol. 2007, p. 35)*.
- [10] Kazienko, P., and Musiał, K. (2006). Recommendation Framework for Online Social Networks, Last M., *Advances in Web Intelligence and Data Mining. Springer Berlin Heidelberg*. 23(1), 111-120
- [11] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999 (pp. 230-237)*. *Association for Computing Machinery, Inc*. <https://doi.org/10.1145/312624.312682>
- [12] Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2), 133-151.

- [13] Lee, S., Yang, J., & Park, S. Y. (2004, October). Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. *In International Conference on Discovery Science (pp. 396-402)*. Springer, Berlin, Heidelberg.
- [14] Rubens, Neil; Elahi, Mehdi; Sugiyama, Masashi; Kaplan, Dain (2016). "Active Learning in Recommender Systems". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook (2 ed.)*. Springer US. doi:10.1007/978-1-4899-7637-6_24. ISBN 978-1-4899-7637-6.
- [15] Elahi, Mehdi; Ricci, Francesco; Rubens, Neil (2016). "A survey of active learning in collaborative filtering recommender systems". *Computer Science Review*. 20: 29–50. doi:10.1016/j.cosrev.2016.05.002.
- [16] Grčar, Miha & Mladenić, Dunja & Fortuna, Blaž & Grobelnik, Marko. (2006). Data Sparsity Issues in the Collaborative Filtering Framework. 58-76. 10.1007/11891321_4.
- [17] Aggarwal, Charu C. (2016). *Recommender Systems: The Textbook*. Springer. ISBN 9783319296579.
- [18] Peter Brusilovsky (2007). The Adaptive Web. p. 325. ISBN 978-3-540-72078-2.
- [19] Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69(Supplement 32), 175-186.
- [20] Felfernig, A., & Burke, R. (2008, August). Constraint-based recommender systems: technologies and research issues. *In Proceedings of the 10th international conference on Electronic commerce (p. 3)*. ACM.
- [21] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker (2011). Developing constraint-based recommenders. *Recommender Systems Handbook*, Springer, pp. 187–216.
- [22] D. Bridge, M. Goker, L. McGinty, and B. Smyth (2005). Case-based recommender systems. *The Knowledge Engineering Review*, 20(3), pp. 315–320.
- [23] Burke, R. (1999, July). Integrating knowledge-based and collaborative-filtering recommender systems. *In Proceedings of the Workshop on AI and Electronic Commerce (pp. 69-72)*.
- [24] Lorenzi, F., & Ricci, F. (2003, August). Case-based recommender systems: A unifying view. In IJCAI Workshop on Intelligent Techniques for Web Personalization (pp. 89-113). Springer, Berlin, Heidelberg.

- [25] Smyth, B. (2007). Case-based recommendation. In *The adaptive web* (pp. 342-376). Springer, Berlin, Heidelberg.
- [26] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331-370.
- [27] Adomavicius, G.; Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". *IEEE Transactions on Knowledge and Data Engineering*. 17 (6): 734–749.
- [28] Gomez-Uribe, Carlos A.; Hunt, Neil (28 December 2015). "The Netflix Recommender System". *ACM Transactions on Management Information Systems*. 6 (4): 1–19.
- [29] Sarwar, B., Karypis, G., Konstan, J. A. and Riedl, J. T. (2000). Analysis of recommendation algorithms for e-commerce. *Proceedings of the 2nd ACM Conference on Electronic Commerce, New York, USA: ACM*, 158-167.
- [30] Choi, K. and Suh, Y. (2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems*, 37(1), 146-153.
- [31] Karl Pearson (20 June 1895) "Notes on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, 58 : 240–242.
- [32] Stigler, Stephen M. (1989). "Francis Galton's Account of the Invention of Correlation". *Statistical Science*. 4 (2): 73–79.
- [33] Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. T. (1999). An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, USA: ACM*, 230-237.
- [34] Karypis, G. (2001). Evaluation of item-based top-n recommendation algorithms. *Proceedings of the 10th International Conference*.
- [35] Breese, J. S., Heckerman, D. and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.*, 43-52.
- [36] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

- [37] Ghazanfar, M. A., Prügel-Bennett, A. and Szedmak, S. (2012). Kernel-mapping recommender system algorithms. *Information Sciences*, 208, 81-104.
- [38] Yu, K., Schwaighofer, A., Tresp, V., Xu, X. and Kriegel, H. P. (2004). Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1), 56-69.
- [39] Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. *Proceedings of the 15th International Conference on Machine Learning, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.*, 46-54.
- [40] Sarwar, B., Karypis, G., Konstan, J. A. and Riedl, J. T. (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. *Proceedings of the 5th International Conference on Computer and Information Science*, 27-28.
- [41] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391-407
- [42] Farzan, R., & Brusilovsky, P. (2011). Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1), 276–284
- [43] Nishino, Kazunori & Iribe, Yurie & Mizuno, Shinji & Aoki, Kumiko & Fukumura, Yoshimi. (2012). The development of a course recommendation system for e-learning students. *International Journal of Knowledge and Web Intelligence*. 3. 19-32. [10.1504/IJKWI.2012.048161](https://doi.org/10.1504/IJKWI.2012.048161).
- [44] Chang, P., Lin, C., & Chen, M. (2016). A Hybrid Course Recommendation System by Integrating Collaborative Filtering and Artificial Immune Systems. *Algorithms*, 9, 47.
- [45] Taha, K. (2018). CRS: A Course Recommender System. In *Student Engagement and Participation: Concepts, Methodologies, Tools, and Applications* (pp. 553-570). *IGI Global*.
- [46] Pardos, Zachary & Jiang, Weijie. (2019). Combating the Filter Bubble: Designing for Serendipity in a University Course Recommendation System..
- [47] Berners-Lee, T., & Connolly, D. (1995). Hypertext markup language-2.0.
- [48] Hoy, M. B. (2011). HTML5: a new standard for the Web. *Medical reference services quarterly*, 30(1), 50-55.

- [49] Badros, G. J., Borning, A., Marriott, K., & Stuckey, P. (1999, November). Constraint cascading style sheets for the web. *In Proceedings of the 12th annual ACM symposium on User interface software and technology* (pp. 73-82). ACM.
- [50] Kennedy, A., & de León, I. (2011). CSS Style Guide. In *Pro CSS for High Traffic Websites* (pp. 37-66). Apress, Berkeley, CA.
- [51] Flanagan, D. (2006). JavaScript: the definitive guide. "O'Reilly Media, Inc."
- [52] Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83.
- [53] Jain, N., Bhansali, A., & Mehta, D. (2015). AngularJS: A modern MVC framework in JavaScript. *Journal of Global Research in Computer Science*, 5(12), 17-23.
- [54] Arnold, K., Gosling, J., & Holmes, D. (2005). The Java programming language. Addison Wesley Professional.
- [55] Weaver, A., Luo, J., & Zhang, X. (1999, November). Monitoring and control using the Internet and Java. In *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 99CH37029) (Vol. 3, pp. 1152-1158)*. IEEE.
- [56] Kassem, N., & Team, E. (2000). Designing enterprise applications: Java 2 platform. Addison-Wesley Longman Publishing Co., Inc..
- [57] Gupta, P., & Govil, M. C. (2010). Spring Web MVC Framework for rapid open source J2EE application development: a case study. *International Journal of Engineering Science and Technology*, 2(6), 1684-1689.
- [58] Delaney, K. (2000). Inside Microsoft SQL Server 2000. Microsoft Press.

ÖZGEÇMİŞ

Adı Soyadı : İsmail Burak YAVRU
Yabancı Dil : İngilizce
Doğum Yeri ve Yılı : Bursa / 1990
E-Posta : ismailburakyavru@gmail.com

Eğitim:

- Anadolu Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
2013

Meslek:

- Java Yazılım Uzmanı, İstanbul, 2013
- Kıdemli Yazılım Mühendisi, Eskişehir, 2014
- Öğretim Görevlisi, Eskişehir, 2017
- Öğretim Görevlisi, Bursa, 2018