KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

İŞBİRLİKÇİ FİLTRELEMEYE DAYALI WEB TABANLI ÖNERİ SİSTEMİ GELİŞTİRİLMESİ

FİDAN KAYA

KOCAELİ 2012

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

İŞBİRLİKÇİ FİLTRELEMEYE DAYALI WEB TABANLI ÖNERİ SİSTEMİ GELİŞTİRİLMESİ

FİDAN KAYA

Prof.Dr. Adnan KAVAK
Danışman, Kocaeli Üniv. .....

Doç.Dr. Fatih ALAGÖZ
Jüri Üyesi, Boğaziçi Üniv. .....

Yrd.Doç.Dr. Ahmet SAYAR
Jüri Üyesi, Kocaeli Üniv. .....

Tezin Savunulduğu Tarih: 21.12.2012

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada İşbirlikçi Filtreleme Yöntemleri kullanılarak web tabanlı bir öneri sistemi geliştirilmesi hedeflenmiştir. Sistem geliştirilirken uygun olan benzerlik yönteminin seçilebilmesi için benzerlik yöntemleri performans ve sisteme uygunluk bakımından karşılaştırılmıştır. Ayrıca öneri üretme aşamasında kullanılacak komşu seçimi yöntemleri de farklı açılardan karşılaştırılmıştır. Sonuç olarak mobil ve web tabanlı olarak çalışacak, İşbirlikçi Filtreleme Yöntemleri'ni temel alan bir öneri sistemi geliştirilmiştir.

Yüksek lisans eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarıma yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygı değer hocam Prof. Dr. Adnan KAVAK'a, desteklerini benden hiçbir zaman esirgemeyen çalışma arkadaşım Gürel YILDIZ'a, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümüne ve tüm iş arkadaşlarıma teşekkürlerimi sunarım.

Ayrıca maddi ve manevi desteklerini tüm hayatı boyunca esirgemeyen annem Lütfiye KAYA'ya, babam Sebahattin KAYA'ya, abim İsmail KAYA'ya ve tezimi düzenleme aşamasında bana yardımcı olan Kadir GÜLAĞIZ'a şükranlarımı sunarım.

Aralık – 2012 Fidan Kaya

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iii
TABLOLAR DİZİNİ iv
SİMGELER DİZİNİ ve KISALTMALAR v
ÖZET vi
ABSTRACT vii
GİRİŞ 1
1. GENEL BİLGİLER 3
1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri 3
1.2. Tez Çalışmasının Katkıları 4
1.3. Literatür Taraması 5
1.4. Tezin Yapısı 9
2. ÖNERİ SİSTEMLERİ 11
2.1. Veri Madenciliği 11
2.2. Öneri Sistemlerinin Yapısı 13
2.3. Öneri Sistemlerinde Kullanılan Yöntemler 14
2.3.1. İçerik tabanlı filtreleme yöntemi 14
2.3.2. İşbirlikçi filtreleme yöntemi 17
2.3.3. Birliktelik kuralları 19
3. İŞBİRLİKÇİ FİLTRELEME 20
3.1. İşbirlikçi Filtreleme Temelleri 20
3.2. İşbirlikçi Filtreleme Yöntem'leri 21
3.3. Kullanıcı ve Nesne Benzerlikleri 24
3.4. Komşu Seçimi 26
3.5. Değerlendirme Ölçütleri 30
4. ÖNERİ MODELİ 34
4.1. İşbirlikçi Filtreleme Algoritması 34
4.2. Öneri Sisteminin Test Edilmesi 37
4.2.1.Benzetim koşulları 38
4.2.2.Benzetim sonuçları 39
5. İŞBİRLİKÇİ FİLTRELEMEYE DAYALI ÖNERİ SİSTEMİ UYGULAMASI: DEKORASİST SİSTEMİ 46
5.1. Mobil Uygulama 46
5.2. Kullanılan Teknolojiler 47
5.3. Sunulan Öneri Modeli 49
6. SONUÇLAR ve ÖNERİLER 57
6.1. Sonuç 57
6.2. Yapılacak Çalışmalar 57
KAYNAKLAR 59
KİŞİSEL YAYINLAR VE ESERLER 75
ÖZGEÇMİŞ 76

ŞEKİLLER DİZİNİ

Şekil 2.1. Veri madenciliği süreci 12
Şekil 2.2. Veri madenciliği yöntemleri..... 13
Şekil 2.3. Öneri sistemlerinin genel yapısı..... 14
Şekil 2.4. İçerik Tabanlı Filtreleme Yöntemi'nin aşamaları..... 15
Şekil 2.5. İçerik Tabanlı Filtreleme..... 16
Şekil 2.6. İşbirlikçi Filtreleme Yöntemi'nin aşamaları 17
Şekil 2.7. İşbirlikçi Filtreleme Örneği..... 18
Şekil 3.1. İşbirlikçi Filtreleme Algoritması için zaman çizelgesi 21
Şekil 3.2. KNN Algoritması..... 28
Şekil 4.1. İşbirlikçi Filtreleme sisteminin genel yapısı 35
Şekil 4.2. Doğrusal Eşleştirme algoritması 35
Şekil 4.3. İşbirlikçi Filtreleme kullanan öneri sistemine ait akış diyagramı 36
Şekil 4.4. Bağıntı Eşiği Yöntemi HKOK değerleri 40
Şekil 4.5. Bağıntı Eşiği MHO değerleri 41
Şekil 4.6. K En Yakın Komşu Yöntemi HKOK değerleri 42
Şekil 4.7. K En Yakın Komşu Yöntemi MHO değerleri..... 42
Şekil 4.8. Benzerlik yöntemleri performans karşılaştırması 43
Şekil 4.9. Tavsiye yöntemleri performans karşılaştırması 43
Şekil 4.10. Komşuluk yöntemleri performans karşılaştırması 44
Şekil 5.1. Sistemin mimarisi..... 46
Şekil 5.2. İşbirlikçi Filtreleme Sistemi'nin genel yapısı 49
Şekil 5.3. Sisteme üye olabilmek isteyen kullanıcıların karşılaşacağı sayfa..... 50
Şekil 5.4. Kullanıcılar tarafından ürünlerin değerlendirildiği sayfa..... 51
Şekil 5.5. Üye giriş sayfası..... 51
Şekil 5.6. Öneri sayfası giriş ekranı..... 52
Şekil 5.7. Öneri işlemi analiz sonuçları..... 52
Şekil 5.8. Öneri sayfası..... 53
Şekil 5.9. Ürünlere ait bilgilerin görüntülendiği sayfa..... 54
Şekil 5.10. Yönetim paneline giriş sayfası..... 54
Şekil 5.11. Firma ekleme sayfası..... 55
Şekil 5.12. Kategori ekleme sayfası 55
Şekil 5.13. Ürün ekleme sayfası..... 56
Şekil 5.14. Sistemde kayıtlı olan ürünlerin tamamının listelendiği sayfa..... 56

TABLOLAR DİZİNİ

Tablo 4.1. Pearson Benzerliği kullanılarak Bağıntı Eşiği Yöntemi'ne ait hata değerleri..... 40
Tablo 4.2. Vektör Benzerliği kullanılarak Bağıntı Eşiği Yöntemi'ne ait hata değerleri..... 40
Tablo 4.3. Pearson Benzerliği kullanılarak K En Yakın Komşu Yöntemi' ne ait hata değerleri..... 41
Tablo 4.4. Vektör Benzerliği kullanılarak K En Yakın Komşu Yöntemi'ne ait hata değerleri..... 42
Tablo 4.5. Puanı hesaplanamayan film sayısı..... 44

SİMGELER DİZİNİ VE KISALTMALAR

Kısaltmalar

ADO	:	ActiveX Data Object (ActiveX Veri Nesnesi)
ASP	:	Active Server Pages (Aktif Sunucu Sayfaları)
ASPX	:	Active Server Pages Entended (Genişletilmiş Aktif Sunucu Sayfaları)
CARS	:	Context Aware Recommendation Systems (İçerik Temelli Öneri Sistemleri)
CF	:	Collaborative Filtering (İşbirlikçi Filtreleme)
CFBIS	:	Collaborative Filtering Based on Influence Sets (Etki Kümesi Temelli İşbirlikçi Filtreleme)
CLR	:	Common Language Runtime (Ortak Programlama Dilleri Çalışma Zamanı)
HKOK	:	Hata Karelerinin Ortalamasının Karekökü
HSPA	:	Hybrid Predictive Algorithm with Smooting (Yumuşatma ile Hibrit Öneri Algoritması)
IMDB	:	Internet Movie Database (İnternet Film Veritabanı)
KNN	:	K Nearest Neighbour (K En Yakın Komşu)
MAE	:	Mean Absolute Error (Mutlak Hatanın Ortalaması)
MHO	:	Mutlak Hatanın Ortalaması
MSIL	:	Microsoft Intermediate Language (Microsoft Ara Dili)
PRC	:	Precision-Recall (Tamlık-Çağrılan)
RMSE	:	Root Mean Squared Error (Hata Karelerinin Ortalamasının Karekökü)
ROC	:	Receiver Operating Characteristic (Alıcı İşlem Karakteristiği)
XML	:	Extensible Markup Language (Genişletilebilir İşaretleme Dili)

İŞBİRLİKÇİ FİLTRELEMEYE DAYALI WEB TABANLI ÖNERİ SİSTEMİ GELİŞTİRİLMESİ

ÖZET

Teknolojinin yaygınlaşması ile birlikte firmalar yaptıkları işleri internet ortamına taşımaktadırlar. Bunun sonucunda da elektronik ortamdaki rekabette gün geçtikçe artmaktadır. Firmalarında bu artan rekabet ortamında üstünlük sağlayabilmeleri için müşterilerine alternatif çözümler sunmaları gerekmektedir. Öneri sistemleri de bu çözümlerden bir tanesidir. Öneri sistemlerinin amacı kullanıcılara internet ortamında yol göstermek ve alternatifler sunmaktır. Bu tip sistemler veritabanında yer alan kullanıcıları ya da veritabanında yer alan ürünlerin içeriklerini ve bunların birbirleri ile olan ilişkilerini dikkate alarak, kullanıcılara tavsiyelerde bulunurlar. Bu tez çalışmasında, İşbirlikçi Filtreleme Yöntemleri'ni kullanan mobilya ve dekorasyon sektörüne yönelik, Dekorasist isimli bir öneri sistemi geliştirilmiştir. Geliştirilmiş olan sistem, dekorasyon sektöründe yer alan ürünleri bir araya getirerek kullanıcılara ürün seçimi sırasında alternatifler sunmayı hedeflemektedir. Kullanıcılara sunulan sistem, mobil ve web uygulama olmak üzere iki kısımdan oluşmaktadır. Tez çalışmasının esas katkısı sistemin web uygulaması bölümünü kapsamaktadır. Web uygulaması bölümünde filtreleme teknikleri ile mobil uygulama bölümünden gelen analiz sonuçları kullanılarak, kullanıcılara öneriler sunulmaktadır. Öneri sunma aşamasında ilk olarak Doğrusal Eşleştirme Algoritması uygulanmıştır. Daha sonra bu algoritma ile filtrelenmiş olan ürünlere ikinci bir filtreleme aşaması olarak İşbirlikçi Filtreleme Algoritması uygulanmıştır. Bu tezde önerdiğimiz öneri sistemine dayalı olarak geliştirilmiş olan yazılım sayesinde mobilya ve dekorasyon firmalarına, ürünlerini müşterilerine ulaştıracak farklı bir yol sunulmuştur. Doğrusal Eşleştirme Algoritması kullanılarak yapılan ön filtreleme işlemi ile klasik İşbirlikçi Filtreleme Yöntemi'nin performansı arttırılmıştır. Yöntem sonucunda elde edilen değerler ile gerçek değerler arasında %0,0035 oranında farklılık olduğu tespit edilmiştir. Bu değer de sistemin gerçeğe yakın sonuçlar ürettiğini göstermektedir. Tez çalışması sonucunda, İşbirlikçi Filtreleme Yöntemi'nin tasarlanan öneri sisteminde uygulanabilirliği gösterilmiştir.

Anahtar Kelimeler: Benzerlik Yöntemleri, İşbirlikçi Filtreleme, Öneri Sistemleri, Veri Madenciliği, Web Tabanlı Öneri Sistemleri

DEVELOPMENT OF WEB BASED RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

ABSTRACT

Companies carry their work to internet environment with the spread of technology. As a result of this, competition based on electronic means is increasing every day. To achieve excellence in this increasingly competitive environment, companies need to provide alternative solutions to their customers. One of the alternative solutions is the recommendation systems. The purpose of recommendation system is to provide guidance and alternative solutions to internet users. By taking into account the relations between users and products stored in database, this type of system makes recommendation to system users. In this thesis study, a recommendation system, referred to as Dekorasist, that uses Collaborative Filtering Methods has been developed, which can be used for furniture and decoration sector. The developed system aims to provide alternatives during product selection by bringing together the products at decoration sector. The system consists of two parts: mobile application and web application. The main contribution of this thesis is the web application part. In the web application part, filtering techniques along with the analysis results coming from mobile application are used to make recommendations to system users. At the first stage of the recommendation, Linear Matching Algorithm has been applied. Later, we applied Collaborative Filtering Algorithm as a second stage filtering to subset of products currently filtered by Linear Matching. With the developed software based on the recommendation system proposed in this thesis, an alternative marketing way is presented to the companies in furniture and decoration sector. The performance of traditional Collaborative Filtering Method is improved by pre-filtering with Linear Matching Algorithm. The error between the actual values and the values obtained using our proposed recommendation system are found to be 0.0035. As a result of this thesis study, applicability of Collaborative Filtering Method to designed recommendation system has been shown.

Keywords: Similarity Methods, Collaborative Filtering, Recommendation Systems, Data Mining, Web Based Recommendation Systems.

GİRİŞ

Teknolojinin hızla gelişmesi ve kullanımının yaygınlaşması, verilerin saklanmasını ve verilere erişilebilmeyi kolay hale getirmiş. Bu durum aynı zamanda farklı sorunları da beraberinde getirmektedir. Teknolojideki gelişim sayesinde bilgilerin neredeyse tamamının elektronik ortama taşınması mümkün hale gelmiştir. Bu yüzden artan verinin içerisinden gerekli ve anlamlı olan verinin, yani bilginin, çıkarılması gerekmektedir. Aynı zamanda firmalar için rekabet gücü, karlılık, hız gibi pek çok etken her zamankinden daha fazla ön plana çıkmıştır. Ancak veriye hızlı bir şekilde ulaşabilmek için verilerin toplanması yeterli değildir. Toplanmış olan veri yığınlarının hızlı bir şekilde yorumlanıp analiz edilmesi gerekmektedir. Bu amaçla geliştirilmiş olan ve yaygın olarak verilerin analiz edilmesinde kullanılan yönteme veri madenciliği denir.

Dünya genelinde veri toplama sistemlerinin yaygın olarak kullanılamaya başlandığı zamanlar 1960'lar olarak bilinmektedir. İlişkisel veri tabanlarının kullanılamaya başlandığı zamanlar ise 1970'li yıllardır. Yukarıdaki verilerden de anlaşıldığı gibi 1960'lardan günümüze kadar veri madenciliğinin aktif olarak kullanılmaktadır. Veri madenciliğinde kullanılan yöntemler analiz işleminin yanı sıra karar verme ve gelecek ile ilgili tahminlerde bulunma gibi alanlarda da kullanılmaktadır.

Elektronik ticaret kavramı, günümüzde gittikçe yaygınlaşan ve internet kullanıcılarının vazgeçemediği bir alan haline gelmiştir. İnternet sayesinde bilgisayar ya da telefon gibi elektronik araçlar la istenilen ürünlere ulaşmak çok kolay bir hale gelmiştir. İnternet üzerinden alışveriş yapmanın kolaylaştığı ve müşteri kapasitesinin de giderek arttığı bu günlerde firmaların farklı rekabet teknikleri geliştirerek üstünlük sağlamaları gerekmektedir. Bu noktada devreye giren öneri sistemleri kullanıcıların geçmiş alışveriş alışkanlıklarını ya da kullanıcıların ürünler hakkındaki değerlendirme verilerini kullanarak hem alışveriş yapan kullanıcının ihtiyaç duyacağı ürünlere kolay bir şekilde ulaşmasını, hem de firmaların satış ve karlılık oranlarının arttırılmasını sağlamaktadır. Bu yüzden öneri sistemlerinin önemi de gün geçtikçe

artmaktadır. Pek çok işletme öneri sistemleri aracılığıyla kullanıcılara önerilerde bulunmakta ya da mevcut öneri sistemlerini geliştirmeye çalışmaktadır.

Bu çalışmada da kullanıcılar arasındaki benzerlikleri dikkate alarak önerilerde bulunan kullanıcı tabanlı, İşbirlikçi Filtreleme Yöntemini temel alan bir öneri sistemi oluşturulmuştur. Uygulamada uygun yöntemin seçilmesini sağlayabilmek için Movilens veri kümesi kullanılmıştır. Movilens veri kümesi Grouplens isimli bir araştırma grubu tarafından oluşturulmuş ve araştırmacıların kullanımına açık bir veri kümesidir. Veri kümesi içerisinde kullanıcıların daha önce filmlere vermiş oldukları puanlar yer almaktadır. Geliştirmiş olduğumuz sistemde mobilyalara verilecek olan puanları temel alarak çalışacağından verilerin benzerliği dikkate alınarak Movilens veri kümesi seçilmiştir. Uygulama .Net ortamında C# dili kullanılarak geliştirilmiştir. Kullanıcılara öneri olarak mobilya ve dekorasyon sektöründe yer alan farklı firmalara ait koltuk, köşe takımı, … gb. pek çok farklı ürün sunulmaktadır. Öneri olarak sunulacak ürünler renk uyumu, kategori (koltuk, çekyat, …) ve ortama uygunluk (en, boy, yükseklik, …) olarak farklı açılardan analiz edilmektedirler. Uygun yöntemin seçilebilmesi için Pearson Benzerlik Yöntemi ve Vektör Benzerlik Yöntemi Movilens veri kümesi üzerinde test edilmiştir. Daha sonrada tavsiye üretme bölümünde kullanılacak uygun yöntemin seçilmesi için K En Yakın Komşu ve Bağıntı Eşiği Yöntem'leri karşılaştırılmıştır.

1. GENEL BİLGİLER

1.1. Tez Çalışmasının Amacı ve Başlatılma Sebepleri

Gelişen teknolojiyle birlikte pek çok bilgiye ve ürüne ulaşmak kolaylaşmıştır. Bu yüzden internetten satış (e-ticaret) kavramı her geçen gün gelişmekte ve yaygınlaşmaktadır. Fakat bazı sektörlerde internetten satış konusunda ciddi sıkıntılar bulunmaktadır. Bu sektörlerden bir tanesi de mobilya ve dekorasyon sektörüdür.

Mobilya ve dekorasyon sektöründe, pek çok farklı çeşitte ürün ve firma bulunmaktadır. Çoğu zaman müşterilerin, ürünlerin tamamına aynı anda erişmeleri ve karar vermeleri mümkün olamamaktadır. Ayrıca ürünlere erişim sağlanmış olsa bile hangi ürünün seçileceği konusunda sıkıntılar yaşanmaktadır.

Bu çalışmada da mobilya sektöründeki sorunlara çözüm bulabilmek ve kullanıcılara milyonlarca ürün arasından seçim yapmak zorunda kaldıklarında, yardımcı olabilmek, ayrıca mobilya ve dekorasyon sektörünün de aktif olarak sanal alışveriş pazarında yer almasına yardımcı olabilmek için bu sektöre yönelik İşbirlikçi Filtreleme tabanlı bir öneri sistemi hazırlanmıştır. Sistemin amacı kullanıcılara yardımcı olmaktır. Her kullanıcının farklı bir bakış açısı olacağından kullanıcılara renk seçimi aşamasında farklı tonlar öneri olarak sunulmaktadır ve kullanıcının seçimleri doğrultusunda önerilerde bulunulmaktadır. Öneri olarak sunulan ürünler, kullanıcılara daha önce incelemediği ürünleri sunarak, kullanıcıların farklı ürünlerle karşılaşmasını da sağlamaktır.

Çalışmada ilk olarak dekorasyonu yapılacak olan ortamın görüntü işleme teknikleri kullanılarak analizi yapılmaktadır. Daha sonra analiz sonuçları ve İşbirlikçi Filtreleme Yöntemi kullanılarak, kullanıcının hem isteklerine hem de dekorasyonunu yapacağı ortama ait uygun ürünler kullanıcıya öneri olarak sunulmaktadır. Böylece kullanıcı sistemde yer alan pek çok farklı firmanın ürününe eşzamanlı olarak ulaşmış olacak, hem de dekorasyonunu yapacağı ortama ve kendi beklentilerine uyumlu ürünlerin filtrelenmiş haline erişecektir.

İşbirlikçi Filtreleme Yöntemleri öneri sistemlerinde en yaygın olarak kullanılan yöntemlerden bir tanesidir. Genellikle kullanıcılar arasındaki benzerlikleri dikkate alarak işlem yapmaktadır. Yani kullanıcıların ilgisini çekebilecek ürünler hakkında bilgi edinmeye çalışırken komşularının geçmişte bu ürünlere vermiş oldukları puanları dikkate alarak işlem yapmaktadır. Çalışmada İşbirlikçi Filtreleme Yönteminin kullanılmasının sebebi, geliştirilecek olan sisteme başlangıç aşamasında en uygun yöntemin bu yöntem olmasıdır. Geliştirilmiş olan sistemde kullanılmış olan algoritmalar farklı veri setleri üzerinde test edilmiştir. Performans ve etkinlik sonuçlarına göre değerlendirilerek en uygun olan yöntemin seçilmesi sağlanmıştır.

1.2. Tez Çalışmasının Katkıları

Bu tezde aşağıdaki katkının sağlanması amaçlanmıştır;

Mobilya ve dekorasyon sektörüne yönelik olarak hazırlanmış ve mobil uygulamaya dayalı olarak geliştirilmiş olan sistem ile web üzerinden kullanıcılara öneri sunulması ve bu sistem üzerinde İşbirlikçi Filtreleme Yöntemi'nin uygulanabilirliğinin gösterilmesi.

Gerçekleştirilmiş olan uygulama sayesinde, İşbirlikçi Filtreleme Yöntemi farklı bir alana uygulanmış olacaktır. Bu yüzden mobilya ve dekorasyon alanında yapılacak olan benzer çalışmalara öncülük etmektedir. Ayrıca kullanılmış olan İşbirlikçi Filtreleme Yöntemi geliştirilmeye açık bir yöntemdir. Yöntem, performans, zaman ve doğruluk oranları gibi farklı yönlerden test edilmiştir. Yöntemin uygunluğu kanıtlanmıştır.

Çalışma sonucunda geliştirilmiş olan uygulama, kullanıcılara en kısa sürede en fazla ilgilerini çekebilecek ürünleri bulmaları konusunda yardımcı olmaktadır ve aynı zamanda ortamdan bağımsız, kolay kullanılabilir, anında cevap alınabilen ve ihtiyaca göre sonuç verebilen bir uygulamadır. Ayrıca uygulama ile dekorasyon firmalarına, ürünlerini müşterilerine ulaştırabilecek yeni bir yol sağlanmış olacaktır. Böylece dekorasyon firmalarının teknoloji ile olan bağlarının artması sağlanacaktır.

1.3. Literatür Taraması

Tez çalışması mobilya ve dekorasyon sektörüne yönelik olarak geliştirilmiş bir öneri sistemini kapsamaktadır. Daha önce yapılan çalışmalar incelendiğinde bu sektöre yönelik olarak İşbirlikçi Filtrelemeyi temel alan bir öneri sisteminin mevcut olmadığı görülmektedir. Firmalar daha çok kullanıcının karşısına çıkarmış oldukları renk paleti aracılığıyla ortam rengini ya da ürünün rengini değiştirerek ortama uyum bakımından kullanıcılara yardımcı olmaya çalışmaktadır. Ancak mevcut sistemlerde algoritma temelli olarak çalışan bir öneri sistemi bulunmamaktadır.

Öneri sistemleri, kullanıcıların daha önceki davranışlarını ve alışkanlıklarını inceleyerek, kullanıcılara uygun ürünler önermeyi hedefleyen uygulamalardır [1]. Web tabanlı pek çok kişisel öneri sistemi bulunmaktadır. Bu sistemler kullanıcılara yol göstermek, sistemin performansını ve satış oranlarını arttırmak gibi pek çok farklı amaçla geliştirilmiş olabilirler. Literatür incelemesi yapıldığında öneri sistemlerinde kullanılan pek çok farklı yöntem bulunmaktadır. Bu yöntemlerden bazıları İçerik Bazlı Filtreleme ve İşbirlikçi Filtreleme Yöntemleri [2,3,4], İlişkilendirme Kuralları [1,5], Markov Modelleri [1,6], Demetleme Yöntemleri [1,7] ve Sıralı Dizinler [1,8] Yöntemleri'dir. Web öneri sistemlerinde ise en yaygın olarak İçerik Bazlı ve İşbirlikçi Filtreleme Yöntemleri'nin tercih edildiği görülmektedir.

İçerik Bazlı Filtreleme Yöntemi'nde, genellikle kullanıcılara tavsiyede bulunurken ürünler arasındaki ilişki kuralları dikkate alınır. İlişki satın alınan ürün bilgisi, ziyaret edilen ürün bilgisi, içerik benzerlikleri ya da benzer müşteriler arasındaki seçimler olabilir. Ürün A ve ürün B arasındaki ilişki ölçüsüne, geçmiş kayıtlara bakılarak ulaşılır. Eğer bu iki ürün aynı zamanda satın alındıysa, bunlar arasında yüksek bir ilişki vardır. Aksi durumda düşük bir ilişki vardır. İçerik Bazlı Filtreleme Yöntemi, "Eğer ürün A ve ürün B arasında yüksek bir ilişki varsa, ürün A'yı satın alanlara ürün B'yi tavsiye edebiliriz" şeklindeki yaklaşımları temel alır [3].

İşbirlikçi Filtreleme Yöntemi'nde ise kullanıcılar ve kullanıcılar arasındaki benzerlikler dikkate alınarak ya da belirli bir içerik üzerindeki kişisel görüşler bir araya getirilerek aynı ilgi alanları olan veya benzer türdeki bilgiye ihtiyaç duyan bireyler eşleştirilir. İşbirlikçi Filtreleme Yöntemi'nin temelinde yatan düşünce, "Eğer

bir kullanıcı geçmişte komşuları ile aynı fikirde ise gelecekte de aynı fikirde olma olasılığının yüksek olduğudur" [9].

İşbirlikçi Filtreleme Yöntemi'de kendi içerisinde iki ayrı bölüme ayrılmaktadır. Bunlar Model Tabanlı İşbirlikçi Filtreleme ve Bellek Tabanlı İşbirlikçi Filtreleme Yöntemleri'dir.

Model Tabanlı İşbirlikçi Filtreleme Yöntemi daha çok büyük veri kümelerinde kullanılmaktadır. Bu tip algoritmalar, ilk olarak kullanıcının tercihlerinden bir model oluştururlar ve ardından bu modele uygun olarak öneri üretme işlemi gerçekleştirirler. Geçmiş yıllarda, Model Tabanlı İşbirlikçi Filtreleme Algoritmaları için model oluşturmak için daha çok Bayes Ağları [10], Kümeleme Yöntemleri [10-12] ve Kural Tabanlı Yaklaşımlar kullanılmıştır [13]. Günümüzde ise daha çok Matris Faktörizasyonuna dayalı yaklaşımlar kullanılmaktadır. Model Tabanlı İşbirlikçi Filtreleme Yöntemi alanında yapılan çalışmalara örnek olarak 1998 yılında Breese ve arkadaşları tarafından yapılan çalışma gösterilebilir [10]. Çalışmada bir Demetleme Modeli oluşturularak, aynı tercihleri paylaşan kullanıcılar aynı gruplara yerleştirilmiştir. Belirli bir sınıftaki aktif kullanıcıların oyları bağımsız kabul edilerek Naive Bayes formülüne göre hesaplanmaktadır [1,10]; Nesnelere bağlı olarak kişilerin gruplanmasını içeren algoritmaların, İşbirlikçi Filtreleme Yöntemi'ne uygulanması ise Lyle H. Ungar ve Dean P. Foster tarafından yapılan çalışmada gerçekleştirilmiştir. K Means Kümelemesinin ve Gibbs örnekleme yönteminin pek çok farklı çeşidi kullanılarak kümeleme işlemi gerçekleştirilmiştir. Geliştirilen model ile pek çok farklı özelliğe sahip olan nesnelerin kümelenebileceği gösterilmiştir [12]. 2000 yılında David N. Pennock tarafından ise İşbirlikçi Filtreleme Yöntemleri'ni kullanan, kişiliği tanıma (personality diagnos) adında yeni bir sistem geliştirilmiştir. Geliştirilmiş olan framework sayesinde Pearson Benzerlik Yöntemi'ne bağlı kümeleme işlemleri için yeni bir ölçekleme yöntemi oluşturulmuştur [14]. Miyahara ve Pazzani tarafından ise klasik İşbirlikçi Filtreleme Algoritmalarından farklı olarak hem kullanıcı tabanlı hem de nesne tabanlı İşbirlikçi Filtreleme Algoritması üzerinde Bayesian sınıflandırıcılarını kullanan bir yöntem geliştirilmiştir [15]. 2011 yılında Lian tarafından da kişiselleştirilmiş öneri sistemlerinde, kullanıcılara benzer sayfa önerilerinde bulunmak için geliştirilmiş İşbirlikçi Filtreleme Sistemlerinin yapısı ele alınmıştır. İnceleme işlemini yaparken

de sayfa ağırlıkları ve kullanıcıların sayfalar üzerindeki ortalama ziyaretlerini dikkate almıştır ve kümeleme işlemini buna göre gerçekleştirmiştir [16].

İşbirlikçi Filtreleme Yöntemleri'nin diğer bir çeşidi ise Bellek Tabanlı İşbirlikçi Filtreleme Yöntemleri'dir. Bu algoritmaların diğer bir adı da Kullanıcı Tabanlı İşbirlikçi Filtreleme Algoritmaları'dır. Bu yöntemin ilk çalışmaları 1994 yılında Resnick [17] tarafından yapılmıştır. Yapılan çalışmada kullanıcılara haber önerebilmek için Pearson Bağıntısı'nı kullanan kullanıcı tabanlı bir öneri sistemi geliştirilmiştir. Bellek Tabanlı İşbirlikçi Filtreleme Yöntemi, makine öğrenmesi dilinde, İşbirlikçi Filtreleme Yöntemi'nin tembel versiyonu olarak adlandırılmaktadır. Bu şekilde adlandırılmasının sebebi, işlem yaparken veri tabanının bir kısmı değil de, bütünü üzerinde işlem yapmasıdır.

Kullanıcı Tabanlı İşbirlikçi Filtreleme alanında yapılan çalışmalara örnek olarak Sarvar ve arkadaşları tarafından İşbirlikçi Filtreleme Yöntemleri'nin, öneri kalitesini arttırmak için İçerik Tabanlı Uygulamaları, İşbirlikçi Filtreleme Yöntemi'nin içerisine entegre eden model gösterilebilir [18]. Wittenburg ve arkadaşları tarafından ise web sayfalarında öneri amaçlı kullanılmak üzere GAB isimli bir öneri sistemi tasarlanmıştır. Geliştirilmiş olan yöntem ile sık kullanılan sayfalar belirlenerek kullanıcılara bir etkin sayfa listesi hazırlanmakta ve bu sayfa listesi kullanılarak, kullanıcılara sayfa önerisinde bulunulması sağlanmaktadır [19]. 1999 yılında İşbirlikçi Filtreleme Yöntemleri'ndeki tahminlerin doğruluğunu arttırabilmek için Jonathan J. Herlocker tarafından algoritmik bir Framework geliştirilmiştir. Çalışma sonucunda farklı veri setleri üzerinde Spearman bağıntısının, Pearson bağıntısından daha iyi sonuçlar verdiği gözlemlenmiştir [20]. 2000 yılında GroupLens Araştırma Grubu tarafından e-ticaret sistemleri için kullanılabilecek farklı algoritmaların analizini içeren bir çalışma yapılmıştır. Çalışma iki ayrı veri kümesine uygulanmıştır. Veri kümelerinden bir tanesi Movilens Veri Kümesi diğeri ise bir e-ticaret sitesinden alınmış verilerdir. Çalışma sonucunda farklı alt görevlere sahip parçacıkların bir araya getirilerek daha yeterli bir İşbirlikçi Filtreleme Algoritması'nın gerçekleştirilebileceği gösterilmiştir [21]. Yine 2000 yılında Danyel Fisher ve arkadaşları tarafından da İşbirlikçi Filtreleme Algoritmalarının, Java tabanlı olarak çalıştırılabileceği bir Framework geliştirilmiştir [22]. Goldberg ve arkadaşları tarafından ise 2001 yılında Eigentaste isimli yeni bir algoritma geliştirilmiştir.

Geliştirilen algoritmada oy matrisinin içerisindeki yoğun bir altkümeye Pearson Korelasyon Katsayısını uygulamıştır. Eigentaste Algoritması sayesinde, çevrimiçi hesaplama zamanında anlamlı bir azalma sağlanmıştır [23].

2006 yılında Rong Hu ve Yansheng tarafından geliştirilen Hibrit yöntemde de kullanıcı ve nesne tabanlı İşbirlikçi Filtreleme Yöntemleri birleştirilmiştir. Geliştirilmiş olan Hibrit Algoritma'ya HSPA ismi verilmiştir [24]. 2007 yılında Chen ve Yin, CFBIS isimli yeni bir İşbirlikçi Filtreleme Algoritması önermiştir. Yapılan deneysel çalışmalar sonucunda algoritmanın klasik İşbirlikçi Filtreleme Algoritması'ndan daha iyi sonuçlar ürettiği gözlemlenmiştir. Ayrıca algoritma klasik İşbirlikçi Filtreleme Yöntemi'nin bir eksikliği olan, yetersiz veri sorununu da çözmüştür [25]. Firan ve arkadaşları tarafından yapılan çalışmada web tabanlı uygulamalarda sıklıkla kullanılan etiketlerin kullanıcılar ve kullanıcılara sunulan önerileri, ne şekilde etkilediği üzerinde durulmuş ve Last.fm'e ait veriler kullanılarak etiket temelli İşbirlikçi Filtreleme Yöntemi parça/ şarkı temelli öneri sistemi ile karşılaştırılmıştır [2]. Yine 2007 yılında Chandrashekhar ve Bhasker tarafından kullanıcıların oylarını temel alan Bellek Tabanlı bir İşbirlikçi Filtreleme Algoritması geliştirilmiştir [26]. Buradaki yöntemde aktif kullanıcı için komşuların seçiminde, komşular arasındaki, kompleks ilişkiler kullanılmıştır. Komşu seçiminde kullanılan ölçütler, Entropi ölçüsü kullanılarak ölçülebilecek ölçütlerdir. Kullanılan yöntemin klasik İşbirlikçi Filtreleme Yöntemi'nden daha iyi sonuçlar ürettiği ve her türlü veri kümesi üzerinde uygulanabileceği anlaşılmıştır.

2009 yılında Balturnas ve arkadaşları tarafından filtreleme yöntemleri ile ilgili iki farklı çalışma yapılmıştır. Bunlardan bir tanesi, kullanıcılardan gelecek geribildirimleri kullanan, ön filtreleme yöntemine sahip bir filtreleme sistemidir. Bu sistemin geliştirilmesindeki amaç, CARS olarak adlandırılan içeriğe duyarlı öneri sistemlerinin zamana bağlı olarak geliştirilmesini sağlanmaktır. Bu amaçla, kullanıcı profilleri zamana bağlı olarak alt profillere bölünmüştür ve öneri üretme aşamasında bu alt profillerde dikkate alınmıştır [27]. Diğer çalışmada ise kullanıcı oylarını kullanan, klasik İşbirlikçi Filtreleme Yöntemleri'nin geliştirilmesini sağlamak için kullanıcıların oylama işlemini yaparken, sadece kullanıcı oylarını kullanmanın yeterli olamayacağı, bu yüzden daha önceden kaydedilmiş olan kullanıcıya ait bağlamsal bilgilere ihtiyaç duyulduğu açıklanmıştır. Bu yüzden oylanacak bir ürün için farklı

kategorilerde farklı oylamalar yapılarak daha doğru sonuçlar elde edildiği gösterilmiştir [28]. Rafter tarafından yazılmış olan doktora tezinde, İşbirlikçi Filtreleme Yöntemi'nin geçmişini ve günümüzdeki durumunu kapsayan bir araştırma yapılmış. İşbirlikçi Filtreleme Yöntemi'nin öneri üretme aşamasında, kullanıcıdan etkileşimli olarak bilgi alan bir versiyonu geliştirilmiştir [29].

İşbirlikçi Filtreleme Yöntemi'nde, öneri kalitesini etkileyen en önemli faktörlerden bir tanesi oy değerlerinin yeterli sayıda olup olmamasıdır. Bu problemi çözebilmek için ürünleri tanımlamak amacıyla kullanılan etiketlerin göz önünde bulundurulabileceği düşünülmüştür. Wang ve arkadaşları tarafından yapılan çalışmada benzer şekilde İşbirlikçi Filtreleme Yöntemi'nin gelişmesi sağlamak için etiketler kullanılmıştır [30]. 2011 yılında etiketleri kullanarak önerilerde bulunan İşbirlikçi Filtreleme Sistemleri'nin gelişmesini sağlamak için Hang ve Meifang tarafından CFBTSS isimli yeni bir sistem tasarlanmıştır. Sistemde etiketler ve etiketlerin ait olduğu ürünler arasındaki anlamsal benzerliklerde, filtreleme işlemine dahil edilmiştir ve klasik filtreleme yönteminden çok daha iyi sonuçlar alındığı gözlemlenmiştir [31].

Tez çalışması sırasında geliştirdiğimiz sistemde bellek yani kullanıcı tabanlı bir İşbirlikçi Filtreleme Sistemi'dir. Veri tabanında yer alan kullanıcıların tamamını dikkate alarak hesaplamalar yapılmakta ve öneriler sunulmaktadır. Kullanıcı Tabanlı İşbirlikçi Filtreleme Yöntemi'nin seçilmesinin temel nedenleri sistemdeki verilerin bir model oluşturmaya elverişli olmaması ve içerik olarak çok fazla özellik içeren bir yapıya sahip olmamasıdır. Ayrıca Model Tabanlı İşbirlikçi Filtreleme yöntemi bir modelin oluşturulması ve onun üzerinde işlem yapılmasını gerektiğinden hem sistemde yer alan verilerin güncel olmaması hem de daha fazla zaman kullanımını gerektireceğinden tercih edilmemiştir.

1.4. Tezin Yapısı

Tez çalışması altı bölümden oluşmaktadır. Tezin girişi 1. Bölümde verilmiştir. Bu bölümde yapılan çalışma tanıtılmış, çalışmanın amacı ve öneminden bahsedilmiştir. Giriş bölümünün son kısmında ise konu ile ilgili yapılmış olan çalışmalara dair bir literatür özetine yer verilmiştir. Bölüm 2'de veri madenciliği hakkında kısa bir açıklamaya yer verilmiş ve öneri sistemlerine geçilmiştir. Öneri sistemlerinin yapısı

ve öneri sistemlerinde genel olarak kullanılan yöntemlerden bahsedilmiştir. Bölüm 3'de ise tez çalışmasının temelini oluşturan İşbirlikçi Filtreleme Yöntemi'nden bahsedilmiştir. İşbirlikçi Filtreleme Yöntemi'nin tarihçesi, benzerlik hesaplamalarında ve komşu seçimlerinde kullanılan yöntemlerden kapsamlı olarak bahsedilmiştir. İşbirlikçi Filtreleme Yöntemi'nin avantajlarından da bahsedilerek üçüncü bölüm sonlandırılmıştır. Tez kapsamında geliştirilmiş olan İşbirlikçi Filtreleme Algoritması Bölüm 4'de açıklanmıştır. Aynı zamanda filtreleme algoritmasının uygulanması aşamasında seçilecek olan yöntemlerin karşılaştırılması da bu bölümde verilmiştir. Tezin uygulaması olarak sunulan sistemin açıklaması Bölüm 5'de verilmiştir. Bu bölümde geliştirilmiş olan sistemin mobil uygulama modülünden, kullanılan teknolojilerden ve web uygulama modülünden detaylı olarak bahsedilmiştir. Son bölüm olan Bölüm 6'da ise çalışmanın sonuçları değerlendirilmiş ve ileriye dönük olarak yapılabilecek çalışmalardan bahsedilmiştir.

2. ÖNERİ SİSTEMLERİ

2.1. Veri Madenciliği

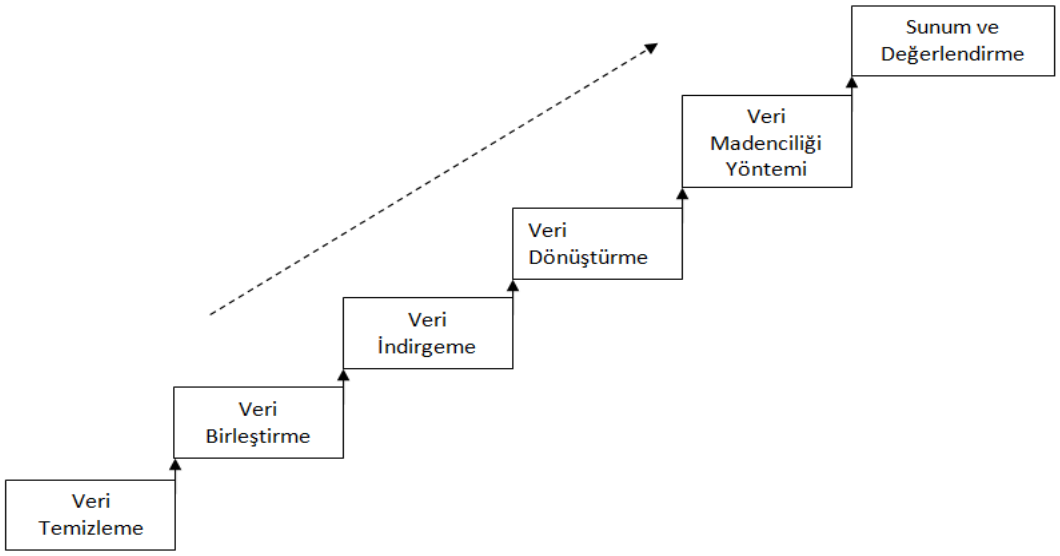
Teknoloji alanındaki gelişmelere her gün bir yenisi eklenmektedir. Bu yeniliklerle birlikte teknolojiyle ilgili ürünler de her geçen gün biraz daha ucuzlamaktadır. Kullanıcılar çok daha fazla özelliğe sahip olan ürünlere çok daha düşük fiyatlarla sahip olabilmektedirler.

Elbette bu yenilikler beraberinde pek çok sorunu da getirmektedir. Artan teknolojiyle birlikte hemen hemen her veri sayısal ortamda kayıt altına alınmaktadır. Gelişen teknoloji bu verileri saklamak için yeterli olmakla birlikte önemli olan, bu verilerin ne işe yaracağı, verilerin gerçek anlamda bilgiye dönüştürülebilecek yapıda olup olmaması, bu bilgiler firmalar ve müşteriler için faydalı hale getirilip getirilemeyeceği hususlarıdır [32].

Bu sorunlara çözüm bulabilmek, veriyi anlamlı ve kullanılabilir hale getirmek için veri üzerinde pek çok işlem uygulanabilir. Bu amaçla geliştirilmiş pek çok matematiksel ve istatistiksel yöntem bulunmaktadır. Ancak klasik veri tabanları ve klasik sayısal yöntemlerle işlem yapıldığında veri sayısındaki artış yüzünden, veri tabanları yetersiz hale gelecektir. Aynı zamanda aşırı yüklenmiş olan bu veri tabanları üzerinde işlem yapmak zorlaşacaktır. Bu sorunlara çözüm bulabilmek için veriyi saklamamızı sağlayan, veri tabanlarından daha kullanışlı veri ambarları ve yararlı bilgiyi elde etmemizi sağlayan veri madenciliği yöntemleri bulunmaktadır.

Veri madenciliğinin asıl amacı var olan veriyi kullanarak veriler arasındaki ilişkileri ortaya koymak ve gerektiğinde geleceğe yönelik tahminlerde bulunabilmektir. Veri madenciliğinde, milyonlarca hatta milyarlarca veri ve çok fazla değişken ile ilgilenilmesi gerekmektedir [32]. Veri madenciliği, verinin bütününü kullanması bakımından diğer istatistiksel yöntemlerden ayrılır. Bu yöntemle, geleneksel yollarla elde edilmiş küçük verilerle çalışma yerine daha kolay değerlendirme yapabilecek, yeni bağımsız veriler tercih edilebilmektedir [33].

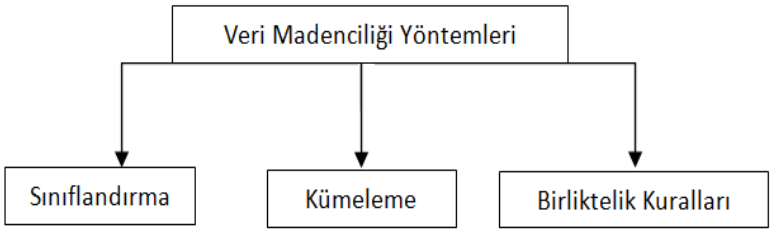
Disiplinler arası nitelik taşıyan veri madenciliğini en yaygın kullanan bilim dalları; Veritabanı Sistemleri, İstatistik, Matematik, Makine Öğrenmesi, Görselleme ve Bilişim Bilimleridir [34].



Şekil 2.1. Veri madenciliği süreci [31]

Veri madenciliğini bir süreç olarak değerlendirdiğimizde; Süreç altı adımdan oluşmaktadır. Bunlar; veri temizleme, veri bütünleştirme, veri indirgeme, veri dönüştürme, veri madenciliği algoritmasının uygulanması ve değerlendirme aşamalarıdır [35]. Sürecin genel görünümü Şekil 2.1'de gösterilmiştir. Sürecin ilk adımı olan veri temizleme işleminde gereksiz ve tutarsız verilerin çıkarılması sağlanmaktadır. Veri bütünleştirme işleminde; pek çok farklı kaynaktan veri birleştirilmektedir. Veri indirgeme işleminde; yapılacak olan analiz ile ilgili olan veriler seçilmektedir. Daha sonra verilerin uygulanacak yönteme uyumlu hale getirilebilmesi için veri dönüştürme işlemi gerçekleştirilmektedir. Veriler uygun hale getirildikten sonra seçilmiş olan veri madenciliği algoritması uygulanır ve çıkan sonuçlar değerlendirilerek elde edilmiş olan yararlı bilginin kullanıcıya sunumu gerçekleştirilir.

Verilerin analizinde genellikle kullanılan üç yöntem bulunmaktadır. Bunlar Şekil 2.2'de gösterildiği gibi Sınıflandırma, Kümeleme ve Birliktelik Kurallarıdır.



Şekil 2.2. Veri madenciliği yöntemleri

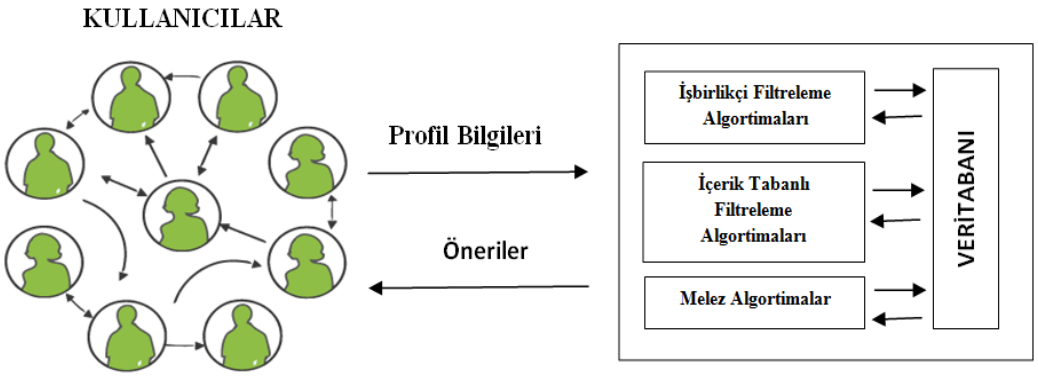
Sınıflandırma günlük yaşamda sıklıkla kullanılan bir yöntemdir. Sınıflandırma işlemi sırasında nesneler bölünerek ayrıştırılır. Pratik karar verme problemlerinin her biri bir sınıflandırma problemi olarak düşünülebilir [36]. Sınıflandırma işleminde ilk olarak veritabanının belirli bir bölümü eğitim verisi olarak kullanılır ve buradan kurallar elde edilir daha sonra bu kurallar kullanılarak tahminlerde bulunulur. Sınıflandırma işleminin pek çok farklı yöntemi bulunmaktadır. Yaygın olarak kullanılan yöntemler; Karar Ağaçları ile Sınıflandırma, Bellek Tabanlı Sınıflandırma ve Regresyon Ağaçları kullanılarak yapılan sınıflandırma yöntemleridir.

Kümeleme işlemi nesneleri birbirlerine göre benzerliklerine bakarak gruplandırma yapmaktadır. Kümeleme işleminde benzer olan nesneler bir kümeye benzer olmayanlar ise benzerlik durumlarına göre başka bir kümeye yerleştirilirler.

Birliktelik kuralları ise hangi olayların birlikte gerçekleşebileceğini tahmin edebilmek için kullanılırlar. Birliktelik kuralları özellikle pazarlama alanında yaygın olarak kullanılırlar. Bu alanda, birliktelik kurallarından faydalanılarak müşterilerin alışveriş alışkanlıkları belirlenmeye çalışılır [32].

2.2. Öneri Sistemlerinin Yapısı

Öneri sistemleri kullanıcıların görüş ya da önerilerini dikkate alarak onların nesneler üzerindeki fikirlerini tahmin etmeye çalışır. Buradaki nesneler kitap, film, müzik, mobilya, online olarak satışa sunulan bir ürün ya da benzerleri olabilir. Bu anlamda öneri sitemleri gelişen teknoloji sonucunda oluşan bilgi havuzu içerisinde kullanıcıların ihtiyaçlarını karşılayacak bilgilere hızlı bir şekilde ulaşmalarını sağlayan bir köprü olarak da düşünülebilir.



Şekil 2.3. Öneri sistemlerinin genel yapısı

Öneri sistemlerinin uygulanması aşamasında pek çok farklı ölçüt göz önünde bulundurulabilir. Bunlardan bazıları; kullanıcılar arasındaki benzerlikler, kullanıcıların ürünler hakkındaki yorumları, geçmişte ziyaret etmiş oldukları web siteleri, beğendikleri ürünlere ait etiketler gibi pek çok özellik olabilir.

Genel olarak öneri sistemlerinin sınıflandırılması sırasında hangi tip verileri kullandıkları göz önünde bulundurulur. Bu veriler nesnelerin içeriklerine ait veriler ise İçerik Tabanlı Filtreleme Sistemleri olarak, sosyal veriler (kullanıcıların ürünler hakkındaki hoşlanma-hoşlanmama durumlarını gösteren veriler) ise İşbirlikçi Filtreleme Sistemleri olarak adlandırılırlar. Bu sistemler haricinde her iki sistemi eş zamanlı olarak kullanan Melez Öneri Sistemleri de bulunmaktadır. Bir sonraki bölümde öneri sistemlerine ait yöntemler detaylı olarak anlatılacaktır.

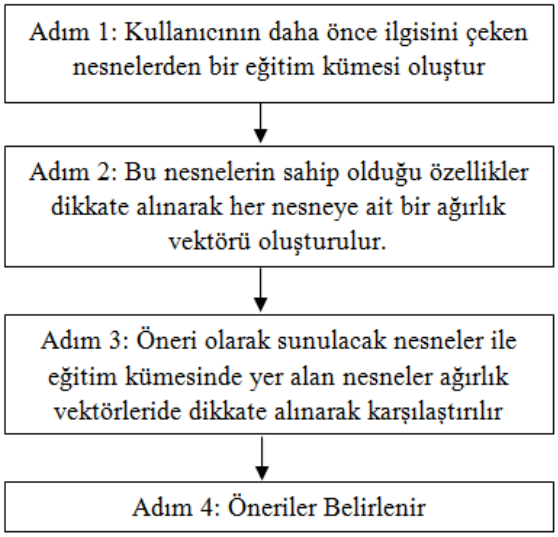
2.3. Öneri Sistemlerinde Kullanılan Yöntemler

Öneri sistemlerinde genel anlamda kullanılan iki temel yöntem bulunmaktadır. Bunlar İçerik Tabanlı Filtreleme ve İşbirlikçi Filtreleme Yöntem'leridir. Bunlara ek olarak Birliktelik Kuralları'ndan da faydalanılmaktadır.

2.3.1. İçerik tabanlı filtreleme yöntemi

İçerik Tabanlı Filtreleme yöntemleri genel olarak dokümanların filtrelenmesi işlemlerinde kullanılırlar. Bu tip sistemler bilginin içeriği hakkında veriler toplarlar ve öneri aşamasında bu verilerden yararlanırlar. Buradaki veriler o içeriğe ait kelimeler ya da cümleler olabilir. Dokümanların içerisinde yer alan her kelimenin belirli bir kullanım sıklığı ya da önemi bulunmaktadır ve böylece her kelimenin

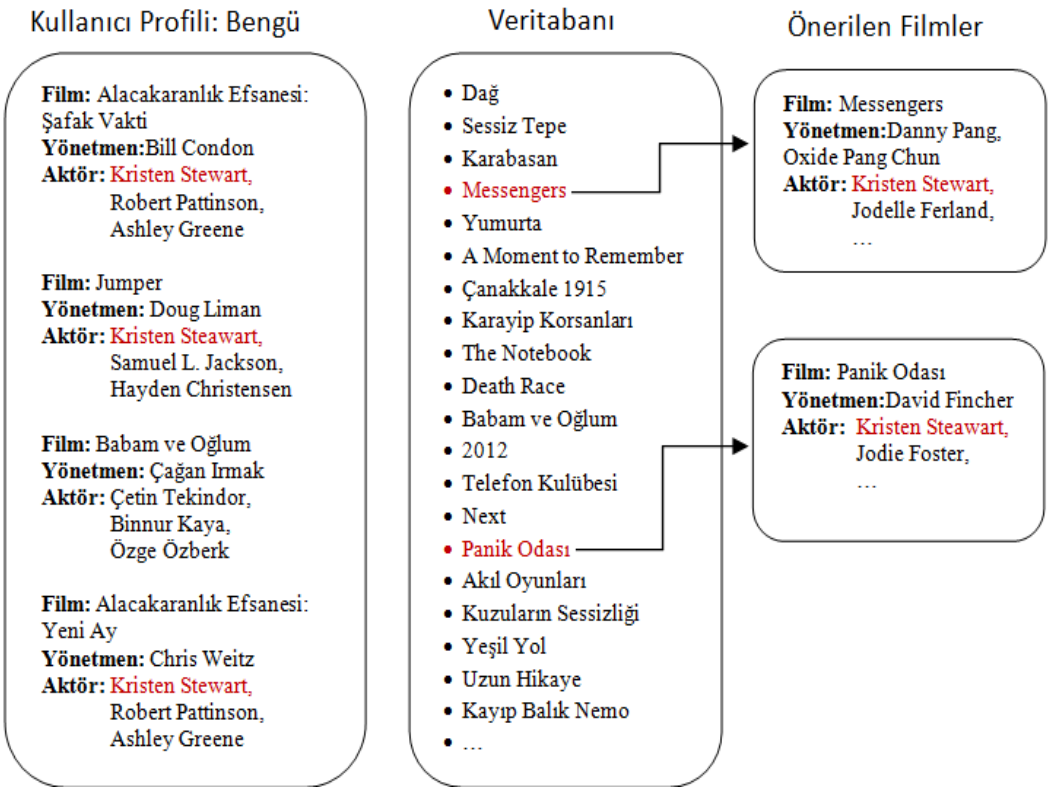
ağırlık olarak adlandırılan bir vektör açıklaması oluşmaktadır. İçerikler arasındaki benzerlikler hesaplanırken ölçüm olarak bu ağırlık vektörleri dikkate alınmaktadır. Kullanılacak yönteme göre en yüksek ya da en düşük ağırlık değerine sahip ilk n kelime dikkate alınarak dokümanların temsil edilmesi sağlanır [37]. Şekil 2.4'de İçerik Tabanlı Filtreleme Yöntemi'nin aşamaları gösterilmiştir.



Şekil 2.4. İçerik Tabanlı Filtreleme Yöntemi'nin aşamaları

Internet ortamında kullanıcılar sürekli olarak bilgi araştırırlar, sayfaları görüntülerler ya da bir şeyler hakkında yorumlar yazarlar. Böylece sistemin kullanıcıların ilgi duyacağı, hoşlanacağı şeyler hakkında bir fikri oluşur [38, 39]. Öneri sistemleri değerlendirme işlemi sırasında bu bilgilerin tamamını kullanırken, İçerik Bazlı Öneri sistemleri önermiş olduğu nesnelerin kullanıcının ne kadar ilgisini çekeceği ile ilgilenmezler [40, 41]. İçerik Bazlı Öneri Sistemi kullanılarak yapılmış olan 10 öneri olduğunu varsayarsak buradaki belki 4 önerinin kullanıcıların ilgilerini çekebileceğini söyleyebiliriz [42].

İçerik Tabanlı sistemlerinin çalışma mantığına örnek olarak "İçerik Tabanlı Filtreleme ile ilgili son 10 yılda yapılmış akademik çalışmaları bul" sorgusu verilebilir. Buradaki sorgu sonucunda cevap olarak döndürülecek sayfaların içeriği göz önünde bulundurularak önerilerde bulunulacaktır. İçerik Tabanlı Filtreleme sistemlerini kullanan tanınmış web siteleri arasında IMDB (http://www.imdb.com) ve Pandora (http://www.pandora.com) da yer almaktadır [29].



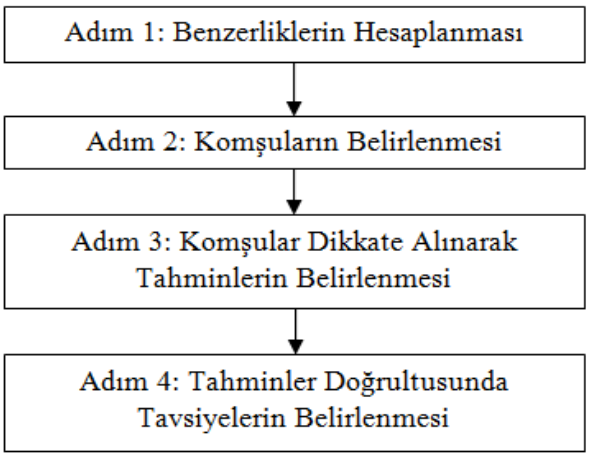
Şekil 2.5. İçerik Tabanlı Filtreleme [29]

IMDB internet ortamındaki film veritabanı olarak bilinmektedir. IMDB öneri sistemini kullanırken Şekil 2.5'de de gösterildiği gibi kullanıcıların daha önce beğenmiş olduğu filmlerin içeriklerini (yönetmen, aktör, anahtar kelimeler, tema …) dikkate almaktadır. Pandora ise Music Genome projesi [43, 44] üzerine geliştirilmiş bir İçerik Tabanlı Filtreleme yapısını kullanmaktadır. Pandora'nın temel amacı kullanıcılara müzik önerisinde bulunmak olduğundan, belirli bir kullanıcı bir şarkıyı ya da müzisyeni dinlediğinde öneri sistemi benzer müzikleri öneri olarak sunmaktadır. Kullanıcılar sistemden istedikleri müzikleri oylayabilirler ve böylece bu bilgide kullanıcıya gelecekteki önerilerinde yardımcı olmak için kullanılabilir. Fakat sistemin genel amacı kullanıcı tercihlerinden önce müziklerin içeriklerini dikkate alarak önerilerde bulunmaktır. Sistemde yer alan parçalar hemen hemen 400 özellik kullanılarak temsil edilmektedir. Bu özellikler belirlenirken bir grup müzisyen tarafından her parça için ayrı ayrı analiz işlemi gerçekleştirilmektedir. Parçalar arasındaki benzerlikler hesaplanırken bu özelliklere ait vektörler kullanılarak bir uzaklık ölçümü çıkarılmaktadır [29].

İçerik Tabanlı Filtreleme Yöntemi'nin zor olan yanı çok fazla veri ile işlem yapılması gerekmesidir. Nesnelere ait açıklamaların, özelliklere indirgenmesi ve bunların nesneleri temsil etmesi gerekmektedir. Bununda yanında nesneleri temsil eden özelliklerin doğru olabilmesi için değersiz özellikler olmaması gerekmektedir. Bu yüzden bu maliyetli işlemlere gerek olmadan geliştirilebilecek yeni bir öneri sistemi oluşturulmuştur. Bu sistem İşbirlikçi Filtreleme Sistemi olarak adlandırılmaktadır.

2.3.2. İşbirlikçi filtreleme yöntemi

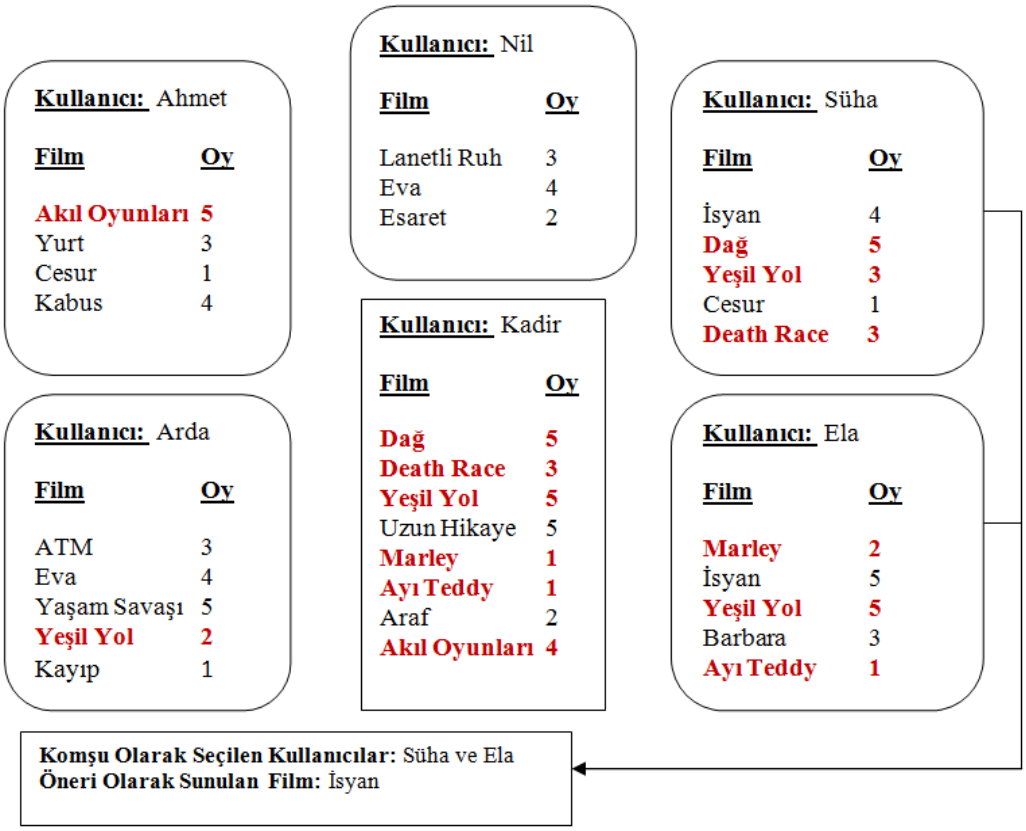
İşbirlikçi Filtreleme Sistemleri öneride bulunurken nesnelerin içerikleri ile ilgilenmezler. Bu yüzden bu sistemler "content-free" sistemler olarak da adlandırılmaktadırlar [29]. İşbirlikçi Filtreleme Sistemleri öneride bulunurken çoğu zaman kullanıcıların bir ürün hakkındaki geçmişteki fikirlerini ortaya koyan özellikler ile ilgilenirler. Bu yüzden İşbirlikçi Filtreleme Sistemleri öneri sürecinde "word of mouth" yani sözlü öneri sistemleri olarak adlandırılırlar [45]. Şekil 2.6'da İşbirlikçi Filtreleme Yöntemi'nin adımları gösterilmiştir.



Şekil 2.6. İşbirlikçi Filtreleme Yöntemi'nin aşamaları

İşbirlikçi Filtreleme Yöntemi'nde, anlamlı öneriler yapabilmenin yolu, kullanıcılar arasında doğru bağlantıları kurabilmeyi gerektirmektedir. Daha önceki deneyimler ve tercihler kullanılarak aynı fikirde olan kullanıcılar komşu olarak belirlenir. Örnek olarak bir nesneyi seven ya da o nesneyi sevmeyen kullanıcılar bir grup altında toplanarak komşu olarak belirlenebilir. Grup oluşturulabilmesi için kullanıcıdan

alınacak olan geribildirimler sayısal bir oy, “thumps up”, “thumps down”, “like”, “dislike” ya da kullanıcının favorilerine eklediği, işaretlediği bir ürün olabilir [29]. İşbirlikçi Filtreleme Sisteminin başarılı önerilerde bulunabilmesi için bu geribildirimlerin doğru bir şekilde yorumlanabilmesi çok önemlidir.



Şekil 2.7. İşbirlikçi Filtreleme Örneği [29]

İşbirlikçi Filtreleme Sistemlerinin çalışma mantığını gösteren bir örnek Şekil 2.7’de verilmiştir. Kadir isimli kullanıcıya film önerisinde bulunabilmek için onunla benzer filmleri beğenmiş olan kullanıcılar (Süha ve Ela) komşu olarak seçilmiştir. Daha sonra komşularının beğendiği ama Kadir’in henüz listesinde olmayan “İsyan” filmi Kadir isimli kullanıcıya öneri olarak sunulmuştur. İşbirlikçi Filtreleme tekniklerini kullanan en sık ziyaret edilen web sitelerinden ikisi Amazon ve Last Fm’e ait web siteleridir. Amazon.com’da çalıştırılan öneri sistemine bakıldığında başlangıç olarak tipik bir öneri sistemi olduğu görülmektedir. Kitap almak isteyen bir kullanıcı olduğu düşünülürse almak istediği kitap ile en benzer özete sahip olan kitaplar listelenmektedir. Bu işlemin yanında o kitabı satın alan kullanıcılarda incelenmektedir ve bu kitabı satın alan kullanıcıların kitap ile birlikte hangi kitapları satın aldığı da dikkate alınmaktadır. Bunlarında yanında daha doğru önerilerde

bulunabilmek adına sistem kullanıcılardan okumuş oldukları kitapları puanlamalarını istemektedir. Böylece kullanıcının hangi tarz kitaplardan hoşlandığı bilgisi de dikkate alınarak daha doğru öneriler üretilmektedir [46]. Last.fm ise sahip olduğu öneri sistemi yardımıyla kullanıcılara müzik önerisinde bulunmaktadır. Last.fm in Last.fm dışında AudioScrobbler isimli bir müzik öneri sistemi de bulunmaktadır. Bu öneri sistemlerinde de kullanıcıların radyodan ya da kendi müzik listelerinden dinlemiş oldukları müzikler karşılaştırılarak benzer müzik zevkine sahip kullanıcılar tespit edilmektedir. Kullanıcının komşuları da dikkate alınarak müzik önerisinde bulunulmaktadır [2]. Amazon ve Last.fm dışında İşbirlikçi Filtreleme Yöntem'lerini kullanan web sitelerine örnek olarak; Digg, eBay, iTunes, Netflix, StumbleUpon verilebilir.

2.3.3. Birliktelik kuralları

Öneri sistemlerinde İşbirlikçi Filtreleme ve İçerik Bazlı Filtreleme Yöntem'leri dışında kullanılan farklı bir yöntem daha bulunmaktadır. Bu yöntemde Birliktelik Kuralları adı verilen yöntemdir. Birliktelik Kuralları, genellikle sepet analizleri, online alışverişler, market raflarının yerleştirilmeleri gibi alanlarda kullanılırlar. Genel mantığı ise bir kullanıcı alışveriş yaparken bir ürün aldığında diğer kullanıcıların bu ürünün yanında başka hangi ürünleri de aldığı sorusuna göre analiz yapmaktır. Birliktelik Kuralları Yöntemi'nde en yaygın olarak kullanılan algoritma Apriori Algoritmasıdır.

3. İŞBİRLİKÇİ FİLTRELEME

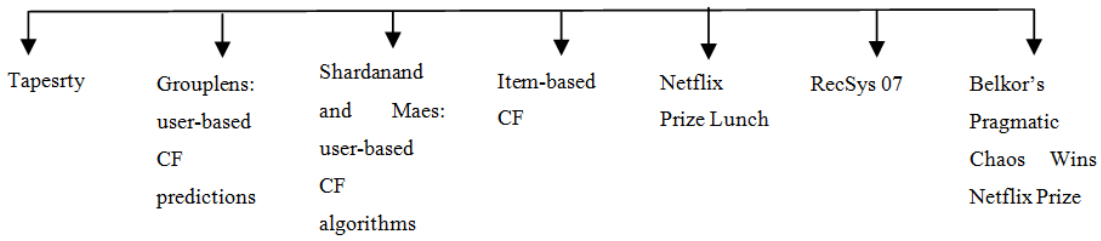
3.1. İşbirlikçi Filtreleme Temelleri

İşbirlikçi Filtreleme Yöntemi'nin temelleri 1992 yılına dayanmaktadır. İlk kez 1992 yılında e-mailleri filtreleme amaçlı olarak geliştirilen Tapestry Projesi'nde kullanılmıştır. Burada geliştirilmiş olan sistemin amacı kullanıcılar tarafından etiketlenmiş olan mailleri belirli kriterlere göre ayrıştırmaktır [29, 47]. İlk kullanıcı tabanlı İşbirlikçi Filtreleme Yöntemi ise Gruplens Araştırma projelerinde 1994 yılında kullanılmıştır. Bu sistem Usenet haberlerini kullanarak kullanıcılara kişisel önerilerde bulunmaktadır [17]. Grouplens tarafından yapılmış olan bu çalışmalar sayesinde araştırmacılar İşbirlikçi Filtreleme Yöntemleri üzerine daha çok yoğunlaşmışlardır. 1995 yılında Shardanand ve Maes tarafından kendilerinin oluşturmuş oldukları Ringo sistemi üzerinde İşbirlikçi Filtreleme Yöntemleri'ni kullanarak çalışmalar yapmışlardır. Grouplens araştırmacıları tarafından yapılan çalışmalarda kullanıcılar arasındaki benzerliklerin hesaplanabilmesi için yaygın olarak kullanılan bir yöntem olan Pearson Benzerlik Bağıntısı'nı kullanmışlardır. Yine 2001 yılında Grouplens araştırmacıları öneri sistemlerinin gelişmesi için çalışmalar yapmışlardır ve Nesne Temelli İşbirlikçi Filtreleme Yöntemi üzerine çalışmışlardır.

2006 yılında ise Netflix tarafından ödüllü bir yarışma düzenlenmiştir ve yarışmada kendi sisteminin İşbirlikçi Filtreleme yapısını geliştirmeyi hedeflemiştir. Yarışma sonucunda kazanacak yarışmacıya bir milyon dolarlık ödül vaat edilmiştir. Yarışma 3 yıl devam etmiştir ve 2009 yılının temmuz ayında sonlanmıştır. Yarışmayı Belkor's Pragmatic Chaos kazanmıştır. Bilindiği gibi Netflix dünyanın en büyük online film kiralama servisidir. Aynı zamanda kullanıcılarına önerilerde bulunabilmek için bir öneri motoruna sahiptir. Firmanın amacı öneri kalitesini %10 arttırmakken implementasyonu gerçekleştirilen algoritmalar içerisinde en fazla %6-7'lik bir artış

sağlanabilmiştir. Belkor's Pragmatic Chaos 2 farklı modelleme yaklaşımı üzerinde durmuştur. Bunlardan ilki kullanıcıların verdiği oyları kullanan "Neighbourhood Modeli" diğeri ise Gizlilik Faktörü olarak adlandırılan filmin korku ya da komedi seviyesini belirleyen ve bu seviyelere göre işlem yapan modeldir [29].

Bunların yanında 2007 yılında düzenlenmeye başlanan ACM Konferansları sayesinde öneri sistemlerinin günümüzde de önemli bir yere sahip olmaları sağlanmıştır. Her geçen gün öneri sistemlerini kullanan web tabanlı pek çok uygulama geliştirilmektedir. Böylece bu tip konferanslar sayesinde filtreleme alanındaki yeni gelişmeler takip edilebilmektedir. İşbirlikçi Filtreleme Yöntemi'nin kırılma noktalarını içeren bir zaman çizelgesi Şekil 3.1 de gösterilmiştir [29].



Şekil 3.1. İşbirlikçi Filtreleme Algoritması için zaman çizelgesi [29]

3.2. İşbirlikçi Filtreleme Yöntem'leri

İşbirlikçi Filtreleme Yöntem'leri kullanıcıların geçmişteki davranışlarına göre öneride bulunma işlemini gerçekleştirirken iki farklı yöntem kullanırlar. Bu yöntemler Model Tabanlı İşbirlikçi Filtreleme ve Bellek Tabanlı İşbirlikçi Filtreleme Yöntem'leridir. Literatürde Bellek Tabanlı İşbirlikçi Filtreleme Yöntem'leri daha yaygın olarak kullanılmaktadır. Model Tabanlı İşbirlikçi Filtreleme Algoritmaları'nın, Bellek Tabanlı Algoritmalar'a göre avantajları bulunmasına rağmen Bellek Tabanlı Algoritmalar kadar doğru sonuçlar üretemediğinden Bellek Tabanlı Algoritmalar günümüzde de Model Tabanlı Algoritmalar'a göre daha yaygın olarak kullanılmaktadır. Bellek Tabanlı Algoritmalar Komşu Tabanlı Algoritmalar olarak da bilinmektedir. Bunlar dışında her iki yöntemi de birlikte uygulayan hibrit sistemlerde bulunmaktadır [29].

Model Tabanlı İşbirlikçi Filtreleme algoritmaları olasılıksal yaklaşımlar kullanarak önerilerde bulunmaktadırlar. Genellikle makine öğrenme yöntemlerini kullanmaktadırlar. Model Tabanlı Algoritmalar'da yaygın olarak kullanılan makine öğrenmesi yöntemleri Bayes Ağları, Kümeleme Yöntem'leri ve Kural Tabanlı Yaklaşımlar'dır. Bunların yanında günümüzde Matris Faktorizasyonu Yöntem'leri de yaygın olarak kullanılmaktadır. Model Tabanlı Yaklaşımlar, Bellek Tabanlı Yaklaşımlar'a göre çok daha hızlı çalışırlar ve bazı durumlarda kullanılan yöntemlere de bağlı olarak Bellek Tabanlı Algoritmalar'a göre çok daha doğru sonuçlar üretebilmektedirler. Hafıza Tabanlı İşbirlikçi Filtreleme Yöntemi'nin yavaş bir yöntem olmasının sebebi, veritabanında yer alan tüm kullanıcı oyları üzerinde ayrı ayrı işlem yapmasıdır. Model Tabanlı İşbirlikçi Filtreleme Yöntemi'nde ise bunun aksine kullanıcı veritabanını bir model oluşturmak ya da öğrenmek için kullanır ve daha sonra tahmin yapmak için bu modelden faydalanır [10, 41].

Model Tabanlı yaklaşımlarda kullanılan veriler offline verilerdir yani güncel veriler değildir. Bellek Tabanlı yaklaşımlarda ise kullanılan veriler günümüze yani yakın zamana ait up-to-date verilerdir. Bellek Tabanlı İşbirlikçi Filtreleme Yöntemi'nde yapılan işlemler çevrimiçi olarak yapılır. Kullanıcılara ait komşular güncel veritabanı içerisinde sürekli aktif kullanıcılar arasından belirlenir. Bellek Tabanlı yaklaşımında kullanıcıların ürünler üzerinde verdiği oylar kullanılarak komşuları belirlenir ve kullanıcılara komşuların oylarını da dikkate alınarak tavsiyelerde bulunulur [48]. Komşuların belirlenmesi aşamasında genellikle istatistiksel benzerlik yöntemleri kullanılır.

Model Tabanlı ve Bellek Tabanlı yaklaşımlara ek olarak En Yakın Komşu Tabanlı Algoritmalar da İşbirlikçi Filtreleme Yöntemin'de kullanılabilmektedir. Bu algoritmalar Bellek Tabanlı yaklaşımlar ile çok benzer olmalarına rağmen tam olarak Bellek Tabanlı yaklaşımları karşılamamaktadır. En yakın Komşu Tabanlı yaklaşımlarda kendi aralarında Kullanıcı Tabanlı İşbirlikçi Filtreleme ve Nesne Tabanlı İşbirlikçi Filtreleme olarak ikiye ayrılmaktadır. Kullanıcı tabanlı olan yöntem yalnızca Bellek Tabanlı İşbirlikçi Filtreleme Yöntemi'ne uygulanabilirken nesne tabanlı olan yöntem hem bellek tabanlı hem de model tabanlı yaklaşımlara uygulanabilmektedir. Kullanıcı Tabanlı (bellek tabanlı) olan İşbirlikçi Filtreleme Yöntemleri en yaygın olarak kullanılan İşbirlikçi Filtreleme Yöntem'leridir.

Kullanıcı Tabanlı İşbirlikçi Filtreleme Yöntemi'nin temel aldığı düşünce benzer kullanıcıların benzer nesnelerden hoşlanacağıdır. Kullanıcı Tabanlı İşbirlikçi Filtreleme Algoritması'nda benzerliklerin hesaplanmasında farklı yöntemler kullanılabilir. Bu yöntemlerden yaygın olarak kullanılanları Pearson Benzerlik Bağıntısı ve Vektör Benzerlik Bağıntısı'dır. Literatüre baktığımızda 2001 yılına kadar ilk olarak kullanılan İşbirlikçi Filtreleme Algoritmaları'nın Kullanıcı Tabanlı Filtreleme Algoritmaları olduğunu görüyoruz [47, 49].

Kullanıcı Tabanlı Algoritmalar çok sık olarak kullanılmalarının yanında uygulama aşamasında bazı zorluklara sahiptirler. Bunlar seyreklik ve ölçeklenebilirliktir [50]. Seyreklik olarak adlandırılan kavram, verilen oy sayısının az olması olarak ifade edilebilir. Milyonlarca filmin ve bu filmlere ait oyların saklandığı bir veritabanı olduğu düşünülürse insanların oyladığı film sayısının düşük olması durumu seyreklik olarak ifade edilir. Bu durumda çok doğru olmayan sonuçlar üretilebilir. Ölçeklenebilirlik olarak ifade edilen kavram ise kullanıcı sayısının sürekli artması ve dolayısıyla yapılacak olan işlemler yüzünden sistemin yavaşlaması durumudur.

Nesne Tabanlı İşbirlikçi Filtreleme Algoritmaları ilk olarak Sarwar ve arkadaşları tarafından 2001 yılında ortaya çıkarılmış bir yöntemdir [51]. Nesne tabanlı yaklaşımlar, kullanıcı tabanlı yaklaşımlara ait olan ölçeklenebilirlik sorununa alternatif olarak çıkarılmışlardır. Bu sorunun temelinde kullanıcıların hızlı bir şekilde artışı yatmaktadır. Bu yüzden araştırmacılar nesnelerin artışının kullanıcıların artışından daha az olduğunu düşünmüşlerdir ve kullanıcılar arasındaki benzerlikleri hesaplamak yerine nesneler arasındaki benzerlikleri hesaplamaya karar vermişlerdir. Nesne temelli yaklaşımının kullandığı ana düşünce ise; kullanıcının geçmişte ilgi duyduğu nesnelere benzer nesnelere gelecekte de ilgi duyacağıdır. Tabi ki bu işlem yapılırken kullanıcılarda işleme katılır. Oyları temel alan bir öneri verileceği düşünülürse kullanıcının profilindeki nesneler ile öneri olarak sunulacak nesneler karşılaştırılır ve en benzer k tanesi seçilir. Benzer nesne olarak seçilecek ürünlerin aynı zamanda komşu olarak seçilen k adet kullanıcının da benzer ürün listesinde yer alması gerekmektedir. Son olarak kullanıcı ile komşularının benzer ürün listesinde yer aldığı belirlenen k adet ürüne ait asıl kullanıcı (öneri sunulacak kullanıcı) tarafından verilen oyların ortalaması alınarak öneri olarak sunulacak ürüne ait puan değerleri hesaplanır [29].

Tez çalışmasında geliştirilmiş olan sistemde Kullanıcı Tabanlı (Bellek Tabanlı) İşbirlikçi Filtreleme Yöntemi kullanılarak öneriler geliştirilmiştir.

3.3. Kullanıcı ve Nesne Benzerlikleri

İşbirlikçi Filtreleme Yöntemi'nde kullanıcılar ya da nesneler arasındaki benzerlikleri hesaplarken kullanılabilecek 4 farklı yöntem bulunmaktadır. Bunlar Pearson Korelasyon Katsayısı (Pearson's Correlation Coefficient) ya da Pearson Benzerliği, Sperman Korelasyon Katsayısı (Sperman's Correlation Coefficient) ya da Benzerliği, Kosinüs Benzerliği (Cosine Similarity) ya da Vektör Benzerliği (Vector Similarity) ve Ayarlanabilir Vektör Benzerliği (Adjusted Cosine Similarity) Yöntem'leridir.

Pearson Benzerliği, İşbirlikçi Filtreleme Yöntemi'nde en sık olarak kullanılan benzerlik bağıntısıdır. İlk olarak Grouplens Projeleri'nde kullanılmıştır. Pearson Benzerliği, iki kullanıcının arasındaki benzerliğini hesaplamak için nesnelere vermiş oldukları puanları karşılaştırır. Benzerlik değeri -1 ile +1 aralığında değişkenlik gösterir. Benzerlik değerinin yüksekliği de iki kullanıcının birbirlerine yakın olduğunu ifade eder. Eğer benzerlik değeri sıfır ise bu iki kullanıcı ya da nesnenin arasında hiçbir benzerlik olmadığı anlaşılır. Pearson Benzerliği'ne ait bağıntı Denklem (3.1)'de gösterildiği gibidir.

$$\rho_{PB}(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{3.1}$$

x ve y kullanıcıları arasındaki benzerliğin hesaplandığı düşünülürse x_i; x kullanıcısının i nesnesine vermiş olduğu puanı, y_i; y kullanıcının i nesnesine vermiş olduğu puanı, n değeri x ve y kullanıcıların ortak oylamış oldukları nesnelerin sayısını ve sırasıyla $\bar{x}$, $\bar{y}$ değerleri ise x ve y kullanıcılarının oylamış oldukları nesnelerin ortalama puan değerini gösterir.

Sperman Korelasyon Katsayısı ise doğrudan nesnelere verilen oylar ile ilgilenmez. Kullanıcılar tarafından verilen oyları sınıflandırarak işlem yapmaya çalışır. Sperman Korelasyon Katsayısı ek işlemler gerektirdiğinden literatürde de yaygın olarak kullanılan bir yöntem değildir. Yöntemin uygulanabilmesi için ilk olarak oyların sınıflandırılması gerekmektedir. Hesaplamalarda oylar üzerinden değil

sınıflandırmalar üzerinden yapılmaktadır. Sperman Korelasyon Katsayısı'da -1 ile +1 aralığında değer almaktadır. -1 değeri nesneler ya da kullanıcıların tamamen farklı olduğunu +1 değeri kullanıcı ya da nesnelerin tamamen benzer olduğunu, 0 değeri ise aralarında bir ilişki olmadığını gösterir. Sperman Korelasyon Katsayısı'na ait bağıntı Denklem (3.2)'de gösterildiği gibidir.

$$\rho_{SKK}(x,y) = \frac{\sum_{i\epsilon I}(k_{x,i}-\bar{k}_x)(k_{y,i}-\bar{k}_y)}{\sqrt{\sum_{i\epsilon I}(k_{x,i}-\bar{k}_x)^2 \sum_{i\epsilon I}(k_{y,i}-\bar{k}_y)^2}} \quad (3.2)$$

x ve y kullanıcıları arasındaki bağıntının hesaplanmış olduğu düşünülürse I değeri ortak oylanmış nesnelerin sayısını, $k_{x,i}$ ve $k_{y,i}$ değerleri kullanıcılar tarafından nesnelere verilmiş olan oyların sıralama değerini, $\bar{k}_x$ ve $\bar{k}_y$ ise kullanıcıların oylamış olduğu nesnelerin sıralama değerlerinin ortalamasını ifade etmektedir [29].

Kosinüs Benzerliği ya da diğer ifade şekliyle Vektör Benzerliği genellikle iki nesne arasındaki benzerliğin hesaplanması gerektiği durumlarda kullanılan yöntemdir. Bu yöntemde her nesne bir vektör olarak düşünülür ve aralarındaki açının kosinüs değeri ise bu nesnelerin birbirlerine olan yakınlığını ifade eder. Pearson Benzerliği ve Sperman Benzerliğinde de olduğu gibi -1, +1 aralığında değerler alır. Vektör benzerliği kullanılarak a ve b kullanıcıları arasındaki benzerlik hesaplanırsa; **x** ve **y** değerlerinin x ve y kullanıcıları arasındaki oy vektörü olduğu düşünülürse;

$$\rho_{VB}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} = \sum_{i=1}^{n} \frac{x_i}{\sqrt{\sum_{i\epsilon I_X} x_i^2}} \frac{y_i}{\sqrt{\sum_{i\epsilon I_y} y_i^2}} \quad (3.3)$$

x_i ve y_i değerleri kullanıcılar tarafından i nesnesine verilmiş olan oyları göstermektedir. Vektör Benzerliği daha çok Nesne Tabanlı Filtreleme Yöntemleri'nde yaygın olarak kullanılmaktadır. Literatürde yapılan çalışmalar incelendiğinde Nesne Tabanlı Filtreleme Yöntemleri'nde daha iyi sonuçlar verdiği gözlemlenmektedir [29].

Ayarlanabilir Vektör Benzerliği Yöntemi ise Nesne Tabanlı İşbirlikçi Filtreleme Yöntemi'ni iyileştirmek için geliştirilmiş bir yöntemdir. Bu yöntem standart Vektör Benzerliği Yöntemi'nin biraz daha değiştirilmiş halidir. İki nesne arasındaki benzerliği hesaplarken her nesnenin ortalama oy değerini gerçek oy değerinden

çıkartmayı temel alır. Bu durum Nesne Tabanlı İşbirlikçi Filtreleme Yöntemi'nin bireysel oyları dikkate almamasından kaynaklanmaktadır. Kullanıcılar arasında çok yüksek ya da çok düşük oy verme eğiliminde olan kullanıcılar olabileceğinden oy değerlerinin yumuşatılması için böyle bir yöntem geliştirilmiştir.

Tez kapsamında yürütülen çalışmada ise Pearson ve Vektör Benzerlikleri kullanılarak hesaplamalar yapılmıştır. Sperman Korelasyon Katsayısı Yöntemi ek işlem gerektirdiğinden, Ayarlanabilir Vektör Benzerliği Yöntemi ise Kullanıcı Tabanlı Filtreleme Yöntemi'ne uygun olmadığından tercih edilmemiştir.

3.4. Komşu Seçimi

Üçüncü bölümde anlatılan yöntemlerle benzerlik hesaplamaları yapıldıktan sonra komşuların belirlenmesi gerekmektedir. Komşu seçiminde de iki temel yöntem bulunmaktadır. Bu yöntemler K En Yakın Komşu (K Nearest Neighbour) ve Bağıntı Eşiği (Correlation Threshold) Yöntem'leridir.

Bağıntı Eşiği Yöntemi'nde bir sınır değeri belirlenir ve sadece bu sınır değerinden daha yüksek benzerliğe sahip kullanıcılar dikkate alınarak hesaplamalar yapılır [45]. Bağıntı Eşiği Yöntemi, benzerlikleri düşük olan kullanıcıları eleyerek gürültü değerini en aza indirgemeyi hedeflemektedir. Ancak eşik değerinin çok yüksek seçilmesi durumunda yeterli sayıda kullanıcı işleme alınamayacağından doğru sonuçlar elde edilemeyecektir.

Tablo 3.1'de Pearson Benzerliği kullanılarak 1 numaralı kullanıcı ile farklı kullanıcılar arasındaki benzerlikler hesaplandığında elde edilmiş olan sonuçlar gösterilmiştir.

Tablo 3.1. Komşular arasındaki benzerlik değerleri

Kullanıcı Numarası	Komşu Numarası	ρ_{PB}
1	69	0,1809526204
1	28	0,2513203349
1	100	0,3689677368
1	2	0,4978353977
1	7	0,5125724168
1	9	0,5125724168
1	12	0,5302448466
1	13	0,5320035202
1	16	0,5825364545
1	6	0,6046314473
1	10	0,6341484785
1	54	0,6348190602
1	79	0,7270357488
1	36	0,8170808045
1	92	0,9251514265
1	3	1,0041364594
…	…	…

Bağıntı Eşiği Yöntemi kullanılarak 1 numaralı kullanıcıya ait komşuların seçileceğini düşünürsek, örnek olarak eşik değeri 0,5555555555 olarak belirlendiğinde bu değerin üstünde kalan kullanıcılar komşu olarak seçilecektir. Yani 16, 6, 10, 54, 79, 36, 92 ve 3 numaralı kullanıcılar komşu olarak belirlenecek ve yapılacak işlemlerde bu kullanıcılara ait veriler kullanılacaktır. Yukarıda eşik değeri örnek olarak 0,5555555555 şeklinde belirlenmiştir. Bağıntı Eşiği Yöntemi'nin kullanılacağı alana göre uygun eşik değerinin belirlenmesi gerekmektedir. Hata hesaplama yöntemleri kullanılarak hangi eşik değerinin daha iyi sonuçlar verdiğine karar verilebilir. En düşük hata değerine sahip benzerlik değeri eşik değeri olarak belirlenecektir.

K-NN Yöntemi'nde ise kullanıcılar ya da gözlemler arasındaki uzaklıkların hesaplanması gerekmektedir. Uzaklıklar hesaplandıktan sonra en küçük uzaklığa sahip k adet komşu ya da gözlem seçilir. Uzaklıkların hesaplanması aşamasında herhangi bir uzaklık hesaplama yöntemi kullanılabilir. En yaygın olarak kullanılan uzaklık hesaplama yöntemi Öklid Uzaklığıdır. x ve y noktaları için Öklid Uzaklığının hesaplanışı Denklem (3.4)'de gösterildiği gibidir.

$$d(x,y) = \sqrt{\sum_{s=1}^{k} \left(x_{is} - y_{js}\right)^2} \tag{3.4}$$

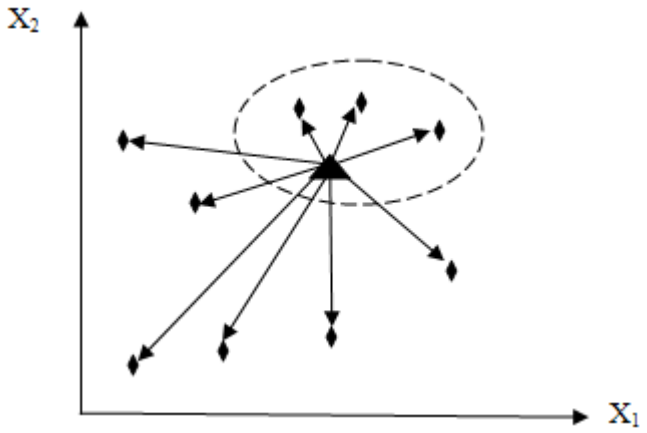
K –NN algoritmasının adımları aşağıda gösterilmiştir [32];

Adım 1: Kaç adet komşu seçileceği (K değeri) belirlenir.

Adım 2: Komşu sayısı belirlenecek nokta ile diğer tüm noktalar arasındaki uzaklıklar hesaplanır.

Adım 3: Uzaklık değerleri en küçük olan K adet nokta komşu olarak belirlenir.

K-NN Yöntemi, İşbirlikçi Filtreleme Algoritması'yla birlikte kullanıldığında benzerlik hesaplamalarında Öklid Uzaklığı yerine bölüm 3.3.'de bahsedilen benzerlik hesaplama yöntemleri kullanılır. 2002 yılından sonra Herlocker tarafından yapılan çalışmalarda İşbirlikçi Filtreleme Yöntem'lerinde komşu sayısının 20 ile 50 aralığında olmasının gerçek dünyada yer alan pek çok uygulama için yeterli olacağı gösterilmiştir. Ancak komşu sayısının 20'nin altına düşürülmesi durumunda hatalı sonuçlar üretildiği gösterilmiştir [20]. Şekil 3.2'de KNN algoritması için en yakın komşu seçimi gösterilmiştir.



Şekil 3.2. KNN Algoritması [32, 52]

Aşağıda verilmiş olan Tablo 3.2'yi kullanarak (6,4) gözlem değerinin hangi sınıfa ait olduğunu KNN algoritmasını kullanarak hesaplarsak;

Tablo 3.2. Gözlem ve sınıf değerleri

A1	A2	SINIF
2	8	Sınıf1
3	5	Sınıf1
4	9	Sınıf1
8	2	Sınıf2
5	5	Sınıf1
3	5	Sınıf1
10	6	Sınıf2
9	6	Sınıf2
7	6	Sınıf2

Komşu sayısını 3 olarak belirlediğimizi varsayarak işlemleri yaparsak; ilk olarak Öklid Uzaklık formülü kullanılarak uzaklıkların hesaplanması gerekmektedir.

(2,8) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(2-6)^2+(8-4)^2} = 5{,}65$

(3,5) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(3-6)^2+(5-4)^2} = 3{,}16$

(4,9) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(4-6)^2+(9-4)^2} = 5{,}38$

(8,2) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(8-6)^2+(2-4)^2} = 2{,}82$

(5,5) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(5-6)^2+(5-4)^2} = 1{,}41$

(3,5) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(3-6)^2+(5-4)^2} = 3{,}16$

(10,6) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(10-6)^2+(6-4)^2} = 4{,}47$

(9,6) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(9-6)^2+(6-4)^2} = 3{,}60$

(7,6) noktası ile (6,4) noktası arasındaki uzaklık: $\sqrt{(7-6)^2+(6-4)^2} = 2{,}23$

Uzaklık değerlerinden en küçük olan üç tanesi dikkate alınacağından (5,5), (7,6) ve (8,2) noktalarının dahil oldukları sınıflara bakılır.

Tablo 3.3. Komşu olarak seçilmiş gözlem değerleri

A1	A2	SINIF
8	2	Sınıf2
5	5	Sınıf1
7	6	Sınıf2

Sınıf2'lerin sayısı Sınıf1'lerin sayısından fazla olduğundan (6,4) noktası Sınıf2'ye dahil edilir [32].

3.5. Değerlendirme Ölçütleri

Öneri sistemlerinin doğruluğunun değerlendirilmesi iki farklı bölümden oluşmaktadır. Bunlardan ilki tahminin doğruluğunun belirlenmesi diğeri ise önerinin doğruluğunun belirlenmesidir.

Tahminin doğruluğunun belirlenmesi sırasında istatistiksel doğruluk ölçütleri ya da karar destek doğruluk ölçütleri kullanılmaktadır. Bu amaçla en yaygın olarak kullanılan istatistiksel doğruluk ölçütleri Mutlak Hatanın Ortalaması (MHO ya da MAE) ve Hata Karelerinin Ortalamasının Karekökü (HKOK ya da RMSE) değerleridir. İstatistiksel doğruluk ölçütleri gerçek değerler ile sistemin üretmiş olduğu değerler arasındaki yakınlığı ölçer [41].

MHO tahminlerin doğruluğunun belirlenmesinde en yaygın olarak kullanılan yöntemdir [10, 14, 18, 20]. MHO İşbirlikçi Filtreleme Yöntemi için önerilen diğer değerlendirme ölçütlerine göre doğruluğu daha kesin sonuçlar üretmektedir ve hesaplanması çok daha kolay bir yöntemdir. MHO yöntemi ilk olarak N adet ürün için mutlak hatayı bulur ve ardından bu hata değerinin ortalamasını alır. MHO Yöntem'inin hesaplanışı (3.5)'de gösterildiği gibidir.

$$MHO(x,\hat{x}) = \sum_{i=1}^{n} \left| \frac{x_i - \hat{x}_i}{n} \right| \tag{3.5}$$

$$HKOK(x, \hat{x}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(x_i - \hat{x}_i\right)^2} \tag{3.6}$$

Hem (3.5) hem de (3.6) numaralı formüllerde yer alan x değeri gerçek değeri, $\hat{x}$ ise hesaplamalar sonucu elde edilen, tahmini değeri temsil etmektedir. n adet için ürün için mutlak hata hesaplanmaktadır.

HKOK Yöntemi'nin hesaplama şekli MHO Yöntemi ile benzerdir. HKOK Yöntemi, MHO ile karşılaştırıldığında HKOK Yöntemi'nin büyük hatalara karşı daha duyarlı olduğu görülmektedir. HKOK Yöntemi'nin hesaplanışı (3.6)'da gösterilmiştir.

Bu yöntemlere ek olarak oylar ve tahmin edilen oy arasındaki bağıntıda hesaplanabilir. Gerçek oy bir vektör ve tahmin edilen oy değeri ayrı bir vektör olarak düşünülürse, bağıntı değeri bu iki veri vektörü arasındaki uyuşmanın istatistiksel ölçümünü göstermektedir. Bağıntı değeri -1 ile +1 arasında değişen değerler almaktadır. -1 değeri vektörler arasında negatif bir bağlantı olduğunu, +1 değeri ise vektörler arasında pozitif bir bağlantı olduğunu göstermektedir. Bağıntı değerinin hesaplanışı (3.7)'da gösterilmiştir.

$$\text{Bağıntı}(x, \hat{x}) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(\hat{x}_i-\bar{\hat{x}})}{n*\sigma_x*\sigma_{\hat{x}}} \tag{3.7}$$

Burada x_i ve $\hat{x}_i$ değerleri sırasıyla tahmin edilen oy ve gerçek oy değerlerini göstermektedir. $\bar{x}$ ve $\bar{\hat{x}}$ değerleri ise tahmin edilen ve gerçek oylara ait ortalamaları göstermektedir. σ_x ve $\sigma_{\hat{x}}$ ise gerçek ve tahmin edilen oylara ait standart sapma değerlerini göstermektedir. Standart sapma değerlerinin nasıl hesaplanacağı (3.8)'de gösterilmiştir [29].

$$\sigma_x=\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2} \tag{3.8}$$

HKOK ve MHO Yöntem'leri gerçek değer ile tahmin edilen değer arasındaki hatayı hesapladıkları için sonuç olarak elde edilen değer ne kadar düşük ise hata değeri de o kadar küçük olacaktır. Bağıntı Yöntemi'nde ise bağıntı değeri ne kadar büyük ise tahminin doğruluğu da o kadar iyi olacaktır [41].

Tahminlerin doğruluğunun belirlenmesinde kullanılan diğer yöntem karar destek ölçütleriydi. Bu yöntemler üretilen tahminlerin kullanıcıya eleman seçimi sırasında, ne kadar faydalı olduğuyla ilgilenmektedir. En yaygın olarak kullanılan karar destek

ölçütleri Alıcı İşlem Karakteristiği (ROC) ve Tamlık-Çağrılan (PRC) duyarlılık ölçütleridir [41].

Bir öneri senaryosu olduğu düşünülürse, ROC Duyarlılık Ölçüsü, nesneler ve gürültü verileri arasındaki uzaklığın en iyi değerde olmasını sağlamaya çalışmaktadır. Yani öneri sisteminin, kullanıcıyı, yüksek oy alanlara yakınlaştırabilmesi düşük oy alanlardan uzaklaştırabilmesinin ölçüsüdür. İşlemsel olarak ise 0-1 aralığında çizilen ROC eğrisinin altında kalan alan ile gösterilir. ROC eğrisi alanı duyarlılık ve belirlilik olmak üzere iki parçaya böler. Duyarlılık, rastgele olarak seçilen bir elemanın kabul edilme olasılığı, Belirlilik ise, kötü bir elemanın reddedilme olasılığıdır. Eğri altında kalan alanın 1 değerini alması tüm elemanların mükemmel olduğunu göstereceğinden en mükemmel durum olarak değerlendirilir. ROC duyarlılık ölçütünün kullanılabilmesi için kullanıcı tarafından hangi elemanların iyi hangilerinin kötü olarak değerlendirileceğinin belirlenmesi gerekmektedir [29, 41].

PRC ise öneri sisteminin kullanıcıya sunduğu bilgilerin ne kadarının kullanıcının ilgi alanına girdiğinin ölçüsüdür. İşlemsel olarak ROC Duyarlılık Ölçütüne benzer bir gösterime sahiptir. Tamlık-çağrılan eğrisi olarak adlandırılan alan ile gösterilir. Tamlık seçilmiş olan ürünlerden yüzde kaçının kullanıcının ilgi alanına girdiğini gösterirken, çağrılan ise ilgi alanına giren ürünlerden yüzde kaçının seçildiğini göstermektedir [41].

Önerilerin doğruluğunun belirlenmesi, tahminin doğruluğunun belirlenmesinden daha karmaşık bir süreçtir. Literatürde önerinin doğruluğunun belirlenmesi için pek çok yöntem yer almaktadır. Bu yöntemlerden en yaygın olarak kullanılanları tamlık, çağrılan ve F1 ölçütleridir. Aynı zamanda bu ölçütler daha önceki araştırmacılar tarafından öneri sistemlerinin geliştirilmesi içinde kullanılmıştır.

Tamlık değeri, hem test kümesinde hem tavsiye listesinde yer alan elemanların sadece tavsiye listesindeki elemanlara olan oranını göstermektedir. Çağrılan ölçütü ise; hem test kümesi hem tavsiye kümesinde yer alan elemanların sadece test kümesinde yer alan elemanlara olan oranıdır. Öneri listesindeki elemanların sayısının fazla olması çağrılan ölçütünün başarılı olma olasılığını arttırırken iyi öneriler sunma olasılığını da azaltmaktadır. F ölçütü olarak adlandırılan ölçüt ise bu iki değerlendirme ölçütünün kombinasyonu gibidir. Ancak kullanıcılarının sistemde yer

alan bütün ürünleri görmemiş ya da oylamamış olma ihtimali yüksek olduğundan bu ölçütlerin öneri sistemlerinin değerlendirilmeleri ve geliştirilmeleri bakımından biraz daha az pratik olmalarına sebep olmuştur. Tamlık, Çağrılan ve F ölçütlerinin matematiksel gösterimleri (3.9), (3.10) ve (3.11) da gösterilmiştir.

$$\text{Tamlık} = \frac{|\{\text{Test Kümesi}\} \cap \{\text{Tavsiye Kümesi}\}|}{\text{Tavsiye Kümesi}} \tag{3.9}$$

$$\text{Çağrılan} = \frac{|\{\text{Test Kümesi}\} \cap \{\text{Tavsiye Kümesi}\}|}{\text{Test Kümesi}} \tag{3.10}$$

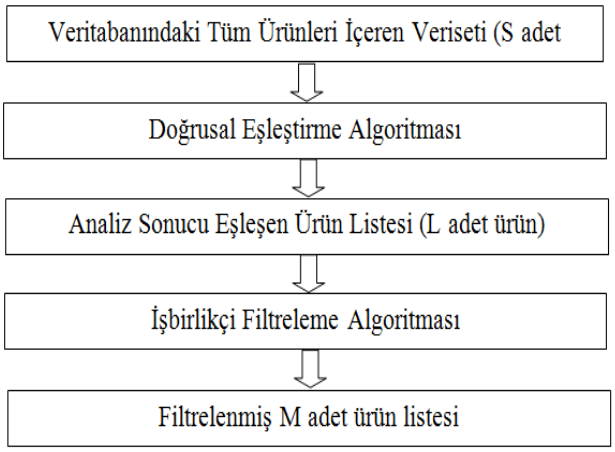
$$\text{F1} = \frac{2 \text{ x Tamlık x Çağrılan}}{\text{Tamlık+ Çağrılan}} \tag{3.11}$$

4. ÖNERİ MODELİ

4.1. İşbirlikçi Filtreleme Algoritması

Tez çalışmasının uygulama bölümü olarak "Dekorasist" isimli, mobil ve web tabanlı bir öneri sistemi geliştirilmiştir. Geliştirilen sistemin mobil uygulama bölümünde yüklenmiş olan resmin analizi yapılmaktadır ve analiz sonuçları web uygulama bölümüne gönderilmektedir. Web uygulama bölümünde geliştirilmiş olan öneri sistemi ile mobil uygulamadan, genel veriler kullanılarak kullanıcıya öneriler sunulmaktadır. Web uygulama bölümünde hem öneri algoritmaları çalıştırılmaktadır hem de firmalara ürünlerini yükleyebilecekleri bir panel sunulmaktadır.

Öneri sistemi iki seviyeli olarak çalışmaktadır. Kullanılan modelin genel bir görünümü Şekil 4.1'de gösterilmiştir. Sisteme üye olan kullanıcılar, giriş yaparak sistemde yer alan ürünlere 1-5 aralığında puan verebilmekte ve sistemden öneri alabilmektedirler. Ayrıca geliştirilmiş olan yönetim paneli aracılığıyla firmalar ürünlerini sisteme yükleyebilmektedirler. Kullanıcı tarafından yüklenmiş olan ortam resmine ait öneri sunulabilmesi için ilk olarak "color quantization" metotları kullanılarak, analizi yapılmış olan resme en yakın tonlara sahip L adet ürün listelenmektedir. Burada Doğrusal Eşleştirme (Lineer Matching) Algoritması kullanılmaktadır. Doğrusal Eşleştirme Algoritması veri tabanında yer alan bütün ürünleri analizi yapılmış ortama ait tonlarla karşılaştırmaktadır ve minimum farka sahip olan L adet ürün listelenmektedir. Daha sonra İşbirlikçi Filtreleme Algoritması, Doğrusal Eşleştirme Yöntemi uygulanarak filtrelenmiş L adet ürüne uygulanarak, her ürüne ait puan değeri hesaplanmaktadır. Son olarak puanı hesaplanan ürünlerden en yüksek puan değerine sahip M adet ürün kullanıcıya öneri olarak sunulmaktadır. İşbirlikçi Filtreleme Algoritması'na ait akış diyagramı Şekil 4.1'de detaylı olarak gösterilmiştir.



Şekil 4.1. İşbirlikçi Filtreleme sisteminin genel yapısı

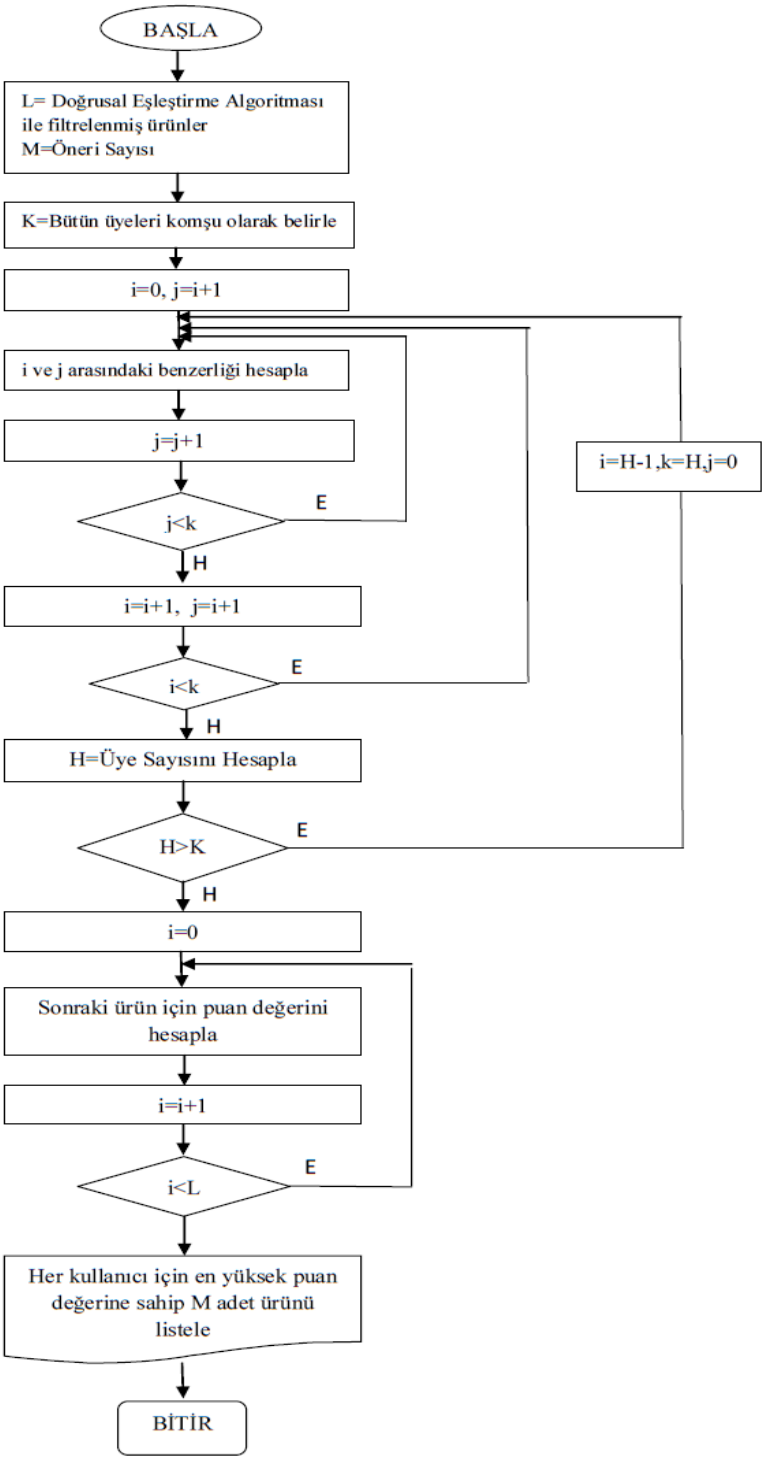
Doğrusal Eşleştirme Algoritması'na ait örnek bir kod parçacığı aşağıda gösterilmiştir. En yüksek benzerliğe sahip ürünleri elde etmek için Şekil 4.2'de yer alan algoritma uygulanır.

```
for w=1:S;
  if D_{w,a}<D_b
        p_w = |c_b − c_{w,a}|
  end
end
p_opt = min{p_w}
```

Şekil 4.2. Doğrusal Eşleştirme Algoritması

Algoritmada yer alan c_b değeri mobil kısımdan gelen ya da web uygulaması kısmında renk nicemleme metotları uygulanarak elde edilen analiz sonucunu göstermektedir. D_b değeri ise dekorasyonu yapılacak olan ortama ait boyut (en, boy, yükseklik) değerlerini göstermektedir. $D_{w,a}$ ise veritabanında yer alan ürünlere ait boyut değerlerini temsil etmektedir. $C_{w,a}$ değeri de veritabanında yer alan ürünlere ait renk analizi sonucunu göstermektedir. Doğrusal Eşleştirme Algoritması, web uygulamasına ait veritabanında yer alan değerler ile mobil uygulamasından gelen analiz sonuçlarını karşılaştırarak en fazla benzerliğe sahip olan L adet ürünü listeler. Doğrusal Eşleştirme Algoritması uygulanmadan önce kullanıcıların farklı tonlardan

hoşlanabileceği de göz önünde bulundurularak ortamın analiz sonucu olarak kullanıcıya dört farklı ton (maksimum, minimum, ortalama ve zıt tonlar) sunulmaktadır. Kullanıcının bu tonlar arasından yapmış olduğu seçime göre Doğrusal Eşleştirme Algoritması uygulanmaktadır.



Şekil 4.3. İşbirlikçi Filtreleme kullanan öneri sistemine ait akış diyagramı

Doğrusal Eşleştirme Algoritması uygulandıktan sonra elde edilen L adet ürüne Şekil 4.3'deki İşbirlikçi Filtreleme Algoritması uygulanmaktadır. Algoritmada yer alan M değeri Doğrusal Eşleştirme Algoritması ve İşbirlikçi Filtreleme Algoritması uygulandıktan sonra öneri olarak kullanıcılara sunulacak olan ürünlerin sayısını göstermektedir. L değeri ise sadece Doğrusal Eşleştirme Algoritması uygulandıktan sonra elde edilen ürünlerin sayısını göstermektedir. Algoritmada ilk olarak veritabanında yer alan tüm kullanıcılar komşu olarak belirlenir. Daha sonra bu kullanıcılar arasındaki benzerlikler hesaplanır. Benzerliklerin hesaplanabilmesi için Pearson Benzerlik Bağıntısı kullanılmıştır. Daha sonra veritabanında yer alan kullanıcılardan en yüksek benzerliğe sahip olan 50 kullanıcı komşu olarak seçilmiştir. Literatürde İşbirlikçi Filtreleme Sistemleri için yeterli kullanıcı sayısının en az 20 olması gerektiği açıklanmıştır. Bu yüzden 50 kullanıcıdan az komşusu olan kullanıcılar için en az 20 komşu olacak şekilde komşular belirlenmiştir. 20 komşudan az komşusu olan kullanıcılar için ise sadece Doğrusal Eşleştirme Algoritması uygulanmıştır. Komşu seçiminde K-NN Algoritması kullanılmıştır. Daha sonra kullanıcılar için komşuları ve ürüne ait diğer isteklerde göz önünde bulundurularak, Doğrusal Eşleştirme Algoritmasından elde edilmiş olan L adet ürüne ait puan değeri Denklem (4.1) kullanılarak hesaplanmıştır.

$$P(a,j) = \bar{a} + \frac{\sum_{(b \in N) \wedge (b_oylar_j)} (b_j - \bar{b}) * \rho(a,b)}{\sum_{(b \in N) \wedge (b_oylar_j)} |\rho(a,b)|} \quad , j=1,\ldots,L \tag{4.1}$$

Denklem (4.1)'de a kullanıcısının j ürününe vereceği tahmini puan değeri hesaplanmaktadır. $\bar{a}$ değeri a kullanıcısının kullanmış olduğu oyların ortalamasını göstermektedir. b_j değeri ise a kullanıcısının komşularından olan b kullanıcısının j elemanı için verdiği gerçek oy değerini, $\bar{b}$ değeri ise b kullanıcısına ait ortalama oy değerini ifade etmektedir. Yukarıdaki formül ile L adet ürünün tahmini puanı hesaplandıktan sonra kullanıcıya hitap edeceği düşünülen en yüksek puan değerine sahip M adet ürün tavsiye olarak sunulmaktadır.

4.2. Öneri Sisteminin Test Edilmesi

Bu bölümde oluşturulmuş olan öneri sisteminin hangi koşullar altında test edildiği ve elde edilen sonuçlar ayrıntılı bir şekilde açıklanmıştır. İşbirlikçi Filtreleme

Yöntemi'nde benzerliklerin belirlenmesi aşamasında Vektör Benzerliği ve Pearson Benzerliği Yöntem'leri, komşu seçimi sırasında ise K-NN Yöntemi ve Vektör Benzerliği Yöntem'leri karşılaştırılmıştır.

4.2.1. Benzetim koşulları

Tez çalışmasında kullanılmış olan yöntemler Movilens veri kümesi üzerinde test edilmiştir. Movilens veri kümesi Minnesota Üniversitesinde Grouplens araştırma projeleri kapmasında oluşturulmuş bir veri kümesidir. Şuan proje kapsamında oluşturulmuş olan web sitesinde (movielens.umn.edu) 3000 den fazla film seçeneği ve 35000'nin üzerinde kullanıcı bulunmaktadır. Her kullanıcı en az 20 filme oy vermiştir. Ayrıca veri kümesi kullanıcılara ait yaş, cinsiyet, posta kodu gibi bilgilerini de içermektedir.

Grouplens araştırma projesi, Minnesota Üniversitesi Bilgisayar Bilimleri ve Mühendisliği Bölümü üyelerinden oluşan bir grup tarafından yürütülmüştür. Grup üyelerinin Grouplens Projesi dışında İşbirlikçi Filtreleme ve öneri sistemleri alanında geliştirilmiş pek çok projesi bulunmaktadır. Projenin yürütücülüğü John Riedl and Joseph Konstan isimli profesörler tarafından gerçekleştirilmiştir.

Veri kümesi araştırmacıların kullanımına açık olarak oluşturulmuştur ve günümüzde de pek çok kullanıcı web sitesini ziyaret ederek filmleri oylamaktadır, film önerileri almaktadır. Veri kümesinin iki farklı versiyonu bulunmaktadır. Bunlardan bir tanesi 100 bin oy içeren 100K Movilens Veri Kümesi, diğeri ise 1 Milyon Movilens Veri Kümesidir. 100K versiyonunun içerisinde 943 farklı kullanıcı tarafından 1682 tane filme verilmiş 100 bin oy bulunmaktadır. 1 Milyon versiyonunda ise 6040 kullanıcı tarafından 3900 tane filme verilmiş 1 Milyon oy bulunmaktadır [53].

Çalışma kapsamında Movilens veri kümesinin 100K versiyonu kullanılmıştır. Burada yer alan veriler 19 Ekim 1997 den 22 Nisan 1998 tarihine kadar 7 aylık bir süre içerisinde Movilens web sitesi (movielens.umn.edu) üzerinden toplanmıştır. 100 bin veriden oluşan veri kümesi %80 i eğitim %20 si test verisi olacak şekilde parçalanmıştır. Ayrıca bu parçalama işlemi de 5 farklı şekilde gerçekleştirilmiştir ve yöntemler bu beş farklı veri kümesi üzerinde de test edilmiştir. Movilens veri kümesinin seçilmesinin sebebi içerdiği bilgiler bakımından geliştirmiş olduğumuz

sistem verileri ile benzer olmasıdır. Veri kümesinin içerisinde kullanıcılar, filmler ve kullanıcılar tarafından filmlere verilen oylar yer almaktadır. Bizim geliştirecek olduğumuz sistemde yer alacak verilerde de benzer şekilde kullanıcılar ve kullanıcıların mobilya sektöründeki ürünlere vermiş oldukları puanlar yer alacaktır. Veri kümesi içerisinde filmlerin içeriklerine ait herhangi bir bilgi yer almamaktadır. Benzer şekilde bizim veri kümemizde de mobilya ürünlerinin içeriklerine ait bilgiler yer almamaktadır ve bu bilgilere gerek olacak bir algoritma kullanılmamaktadır. Movilens veri kümesi içerisinde yer alan filmler en düşük puan değeri bir ve en iyi puan değeri beş olacak şekilde birden beşe kadar puanlandırılmışlardır.

4.2.2.Benzetim sonuçları

Movilens Veri Kümesi kullanılarak yapılan analiz sonuçları ve seçilen yöntemler aşağıda detaylı olarak açıklanmıştır.

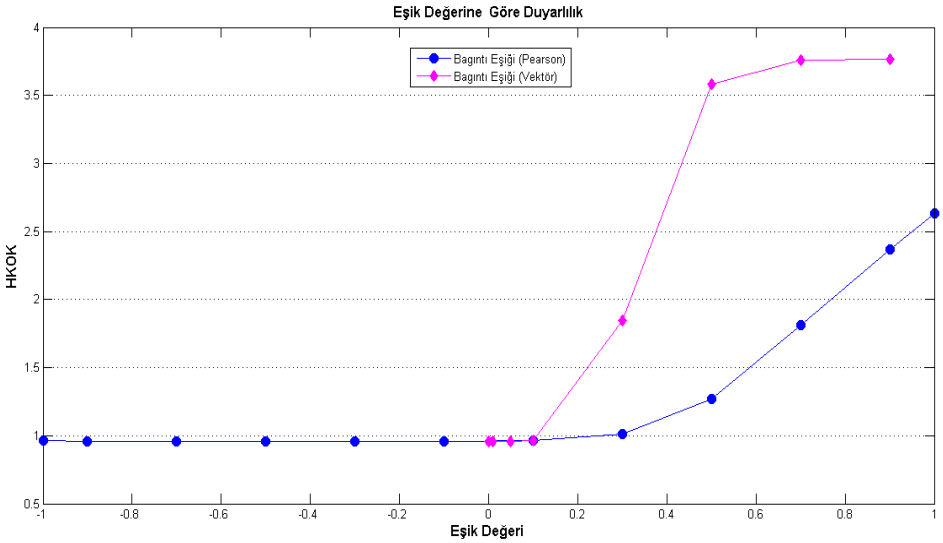
Tablo 4.1 Bağıntı Eşiği Yöntemi'nin Pearson Benzerlik Yöntemi'nden elde edilen benzerlik değerlerine uygulanmış halini, Tablo 4.2 ise Bağıntı Eşiği Yöntemi'nin Vektör Benzerliği Yöntemi'nden elde edilen sonuçlara uygulanmış halini göstermektedir. Benzerlik yöntemleri ile elde edilen değerlere göre tabloda belirtilen eşik değerlerinin yukarısında kalan kullanıcılar, komşu olarak seçilmiştir ve test kümesinde yer alan filmlerin tahmini puan değerleri hesaplanmıştır. Daha sonra tahmini puan değerleri, gerçek puan değerleri ile karşılaştırıldığında, oluşan hata değerlerinin iki farklı hata hesaplama yöntemi (MHO ve HKOK) ile elde edilen sonuçları Tablo 4.1 ve Tablo 4.2'de yer almaktadır. Bu hata değerlerine ait grafikler ise Şekil 4.4 ve 4.5'de gösterilmiştir.

Tablo 4.1. Pearson Benzerliği kullanılarak Bağıntı Eşiği Yöntemi'ne ait hata değerleri

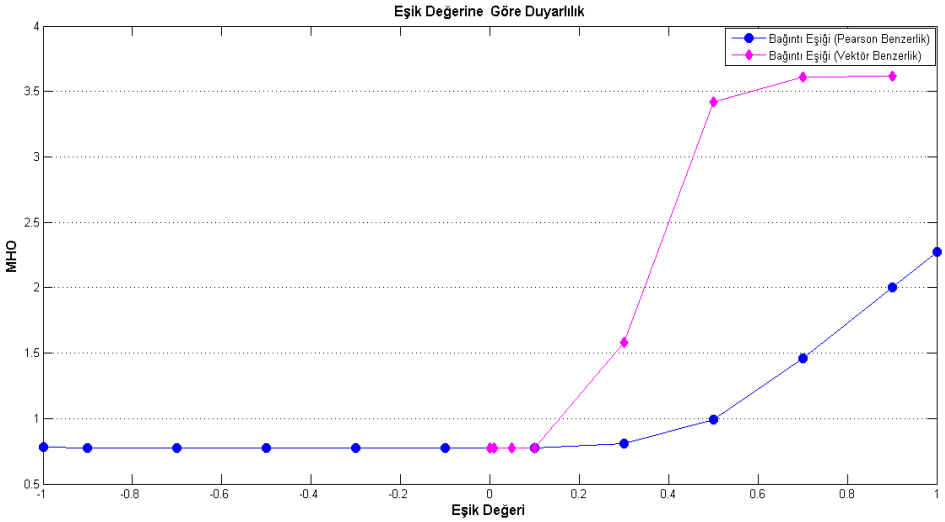
Eşik Değeri	MHO	HKOK
-1	0,781651	0,964265
-0.9	0,776210	0,957661
-0.7	0,774402	0,955845
-0.5	0,773928	0,956968
-0.3	0,772493	0,957183
-0.1	0,773782	0,960299
0.1	0,775200	0,963352
0.3	0,807274	1,013408
0.5	0,990899	1,270321
0.7	1,460177	1,813726
0.9	2,001866	2,367137
1	2,271862	2,633505

Tablo 4.2. Vektör Benzerliği kullanılarak Bağıntı Eşiği Yöntemi'ne ait hata değerleri

Eşik Değeri	MHO	HKOK
0	0,774419	0,958813
0.01	0,774410	0,958807
0.05	0,774054	0,958802
0.1	0,776934	0,964412
0.3	1,580198	1,847446
0.5	3,418578	3,584878
0.7	3,613011	3,756436
0.9	3,619784	3,763345



Şekil 4.4. Bağıntı Eşiği Yöntemi HKOK değerleri



Şekil 4.5. Bağıntı Eşiği MHO değerleri

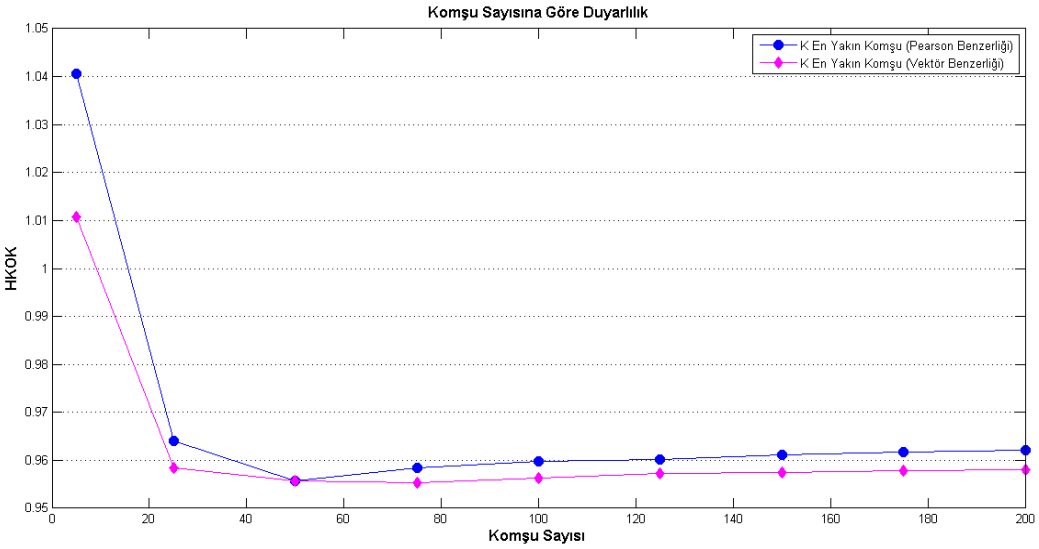
Şekillerden de anlaşılacağı gibi Bağıntı Eşiği Yöntemi için her iki benzerlik yöntemi de benzer sonuçlar göstermiştir. 0-0,2 eşik değerinde her iki yönteminde hata değerleri en aza inmiştir ve çok yakındır. Tablo 4.3 Pearson Benzerlik Yöntemi kullanılarak elde edilen benzerlik değerlerinin K-NN Yöntemi'yle birlikte kullanılması sonucu elde edilen hata değerlerini, Tablo 4.4 ise Vektör Benzerlik Yöntemi kullanılarak elde edilen benzerlik değerlerinin, K-NN Yöntemi'yle birlikte kullanılması sonucu elde edilen hata değerlerini göstermektedir. Şekil 4.6 ve Şekil 4.7'den de anlaşıldığı gibi her iki benzerlik yöntemi de 50 komşu için en düşük hata değerlerini göstermiştir.

Tablo 4.3. Pearson Benzerliği kullanılarak K En Yakın Komşu Yöntemi'ne ait hata değerleri

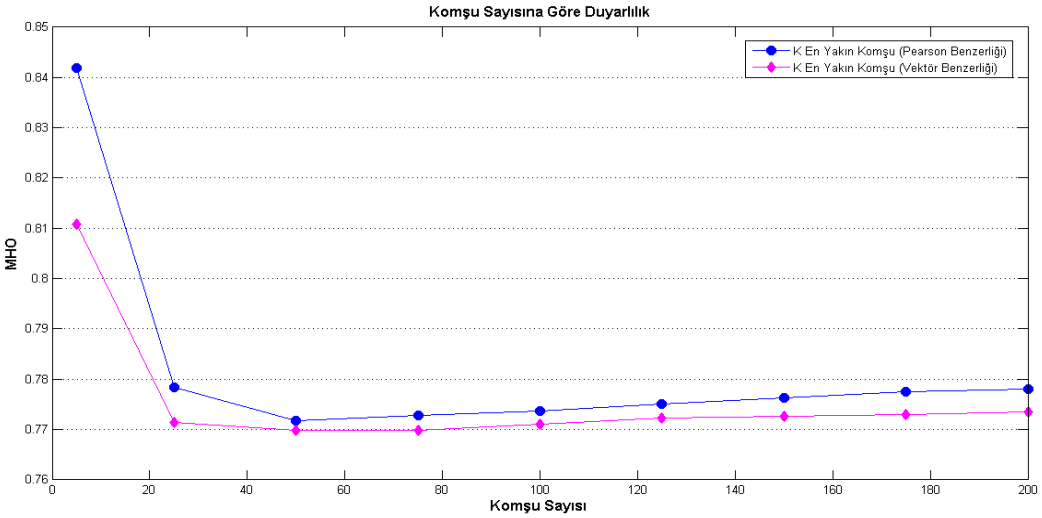
Komşu Sayısı	MHO	HKOK
5	0,841847	1,040622
25	0,778398	0,964094
50	0,771704	0,955769
75	0,772887	0,958492
100	0,773709	0,959755
125	0,775011	0,960186
150	0,776260	0,961103
175	0,777497	0,961738
200	0,778072	0,962139

Tablo 4.4. Vektör Benzerliği kullanılarak K En Yakın Komşu Yöntemi'ne ait hata değerleri

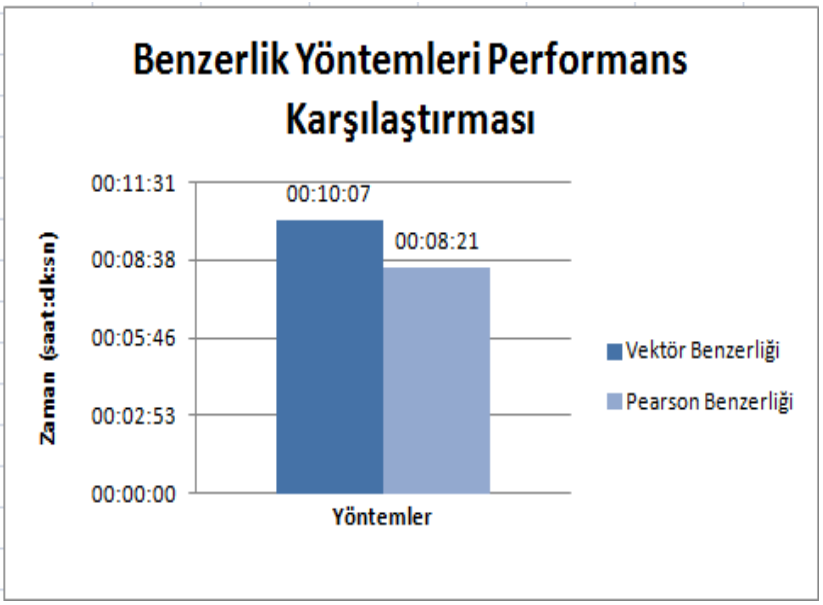
Komşu Sayısı	MHO	HKOK
5	0,810760	1,010673
25	0,771448	0,958405
50	0,769801	0,955693
75	0,769787	0,955278
100	0,771146	0,956257
125	0,775011	0,957220
150	0,772732	0,957500
175	0,773063	0,957768
200	0,773493	0,958112



Şekil 4.6. K En Yakın Komşu Yöntemi HKOK değerleri

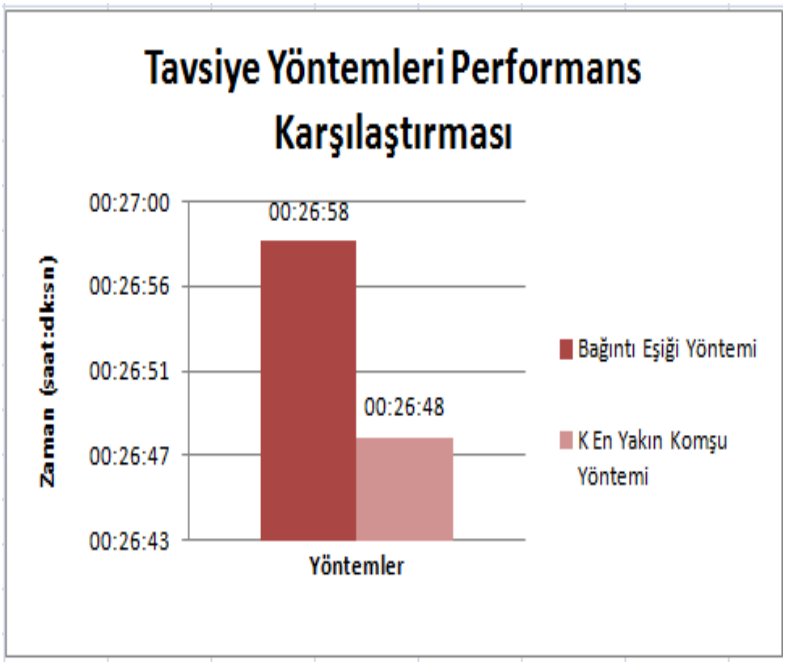


Şekil 4.7. K En Yakın Komşu Yöntemi MHO değerleri



Şekil 4.8. Benzerlik yöntemleri performans karşılaştırması

Hata değerleri birbirine çok yakın olduğundan benzerlik yöntemleri, cevap zamanları dikkate alınarak performans bakımından karşılaştırılmıştır. Şekil 4.8'de benzerlik yöntemlerinin performans karşılaştırması gösterilmiştir. Performans karşılaştırması .Net ortamında yapılmıştır. Toplamda 943 kullanıcı için benzerliklerin hesaplanması gerçekleştirilmiştir ve Pearson Benzerlik Yöntemi Vektör Benzerlik Yöntemi'ne göre çok daha iyi sonuç göstermiştir. Bu yüzden benzerlikler hesaplanırken Pearson Benzerlik Yöntemi'nin kullanılmasına karar verilmiştir.

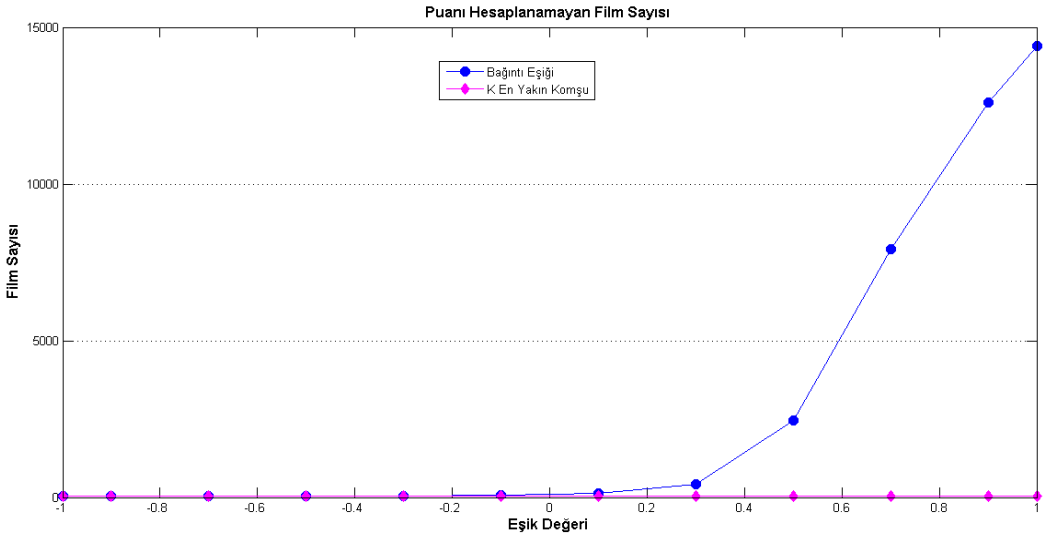


Şekil 4.9. Tavsiye yöntemleri performans karşılaştırması

Yukarıdaki grafiklerden de anlaşılacağı gibi K-NN Yöntemi'nde de Bağıntı Eşiği Yöntemi'nde de ortalama hata değerleri benzer çıkmıştır. Bu yüzden yöntemlerin cevap zamanı dikkate alınmıştır. Şekil 4.9'dan da anlaşılacağı gibi K-NN ve Bağıntı Eşiği Yöntem'leri arasında 20 saniyelik bir cevap zamanı farkı bulunmaktadır. Bu fark değeri de çok küçük olduğundan iki yöntem arasında seçim yapılmasında etkili olmamıştır. Bu yüzden yöntemler farklı bir açıdan; puanı hesaplanamayan film sayısı dikkate alınarak karşılaştırılmıştır.

Tablo 4.5. Puanı hesaplanamayan film sayısı

Yöntem	K En Yakın Komşu	Vektör Benzerliği
Puanı Sıfır Olarak Hesaplanan Film Sayısı	42	42
	42	42
	42	42
	42	50
	42	59
	42	121
	42	423
	42	2454
	42	7910
	42	12596
	42	14412



Şekil 4.10. Komşuluk yöntemleri performans karşılaştırması

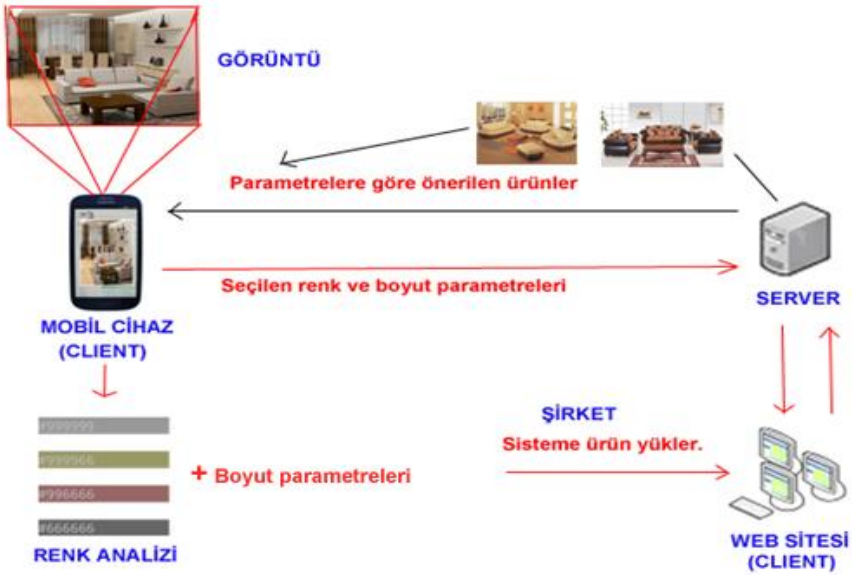
Şekil 4.10 ve Tablo 4.5'de de görüldüğü gibi K-NN Yöntemi'nde puanı hesaplanamayan film sayısı hangi şart altında olursa olsun 42 olarak tespit edilmiştir. Bağıntı Eşiği Yöntemi'nde ise bu değer eşik değeri yukarı çıktıkça yani benzerlik arttıkça yeterli sayıda komşu bulunamadığından artış göstermektedir. K-NN

Yöntemi'nde puanı sıfır olarak hesaplanan filmlerin sebebi ise, puanı hesaplanamayan film için kullanıcının o filme oy vermiş hiçbir komşusunun olmamasıdır. Bağıntı Eşiği Yöntemi'nde ise ortak oy veren komşu olsa bile benzerlik değeri belirlenen, eşik değerinin altında kaldığı için dikkate alınmadığından kullanıcının ilgisini çekecek bir film bile olsa dikkate alınmamasına sebep olmaktadır. Bu durumda hatalı işlem yapılmasına sebep olacağından komşuların belirlenmesi aşamasında K-NN Yöntemi tercih edilmiştir. Yukarıdaki sonuçlar dikkate alınarak benzerlik hesaplamalarında Pearson Benzerlik Yöntemi'ni ve komşu seçiminde de K-NN Yöntemi'ni kullanan bir İşbirlikçi Filtreleme Sistemi geliştirilmiştir.

5. İŞBİRLİKÇİ FİLTRELEMEYE DAYALI ÖNERİ SİSTEMİ UYGULAMASI: DEKORASİST SİSTEMİ

5.1. Mobil Uygulama

Tez çalışmasında geliştirilmiş olan sistemin genel bir görünümü Şekil 5.1'de verilmiştir. Şekilde de görüldüğü gibi uygulama mobil ve web uygulama olmak üzere iki bölümden oluşmaktadır.



Şekil 5.1. Sistemin mimarisi [54]

Mobil uygulama bölümünde kullanıcının kendi telefonuyla çektiği ya da resim galerisinden seçtiği bir resmin renk analizi yöntemleri ile analizi yapılmaktadır. Analiz sonuçları en yoğun renk, en az yoğun renk, ortalama renk ve karşıt renk tonlarından oluşmaktadır. Daha sonra kullanıcı bu renk tonlarından birisini seçer ve öneri almak istediği ürünün boyut parametrelerini de ekler [54]. Bu bilgiler server'a gönderilir. Server'da bulunan web uygulaması tarafından bu veriler değerlendirilir ve öneri olarak sunulacak ürün bilgileri server tarafından mobil uygulamaya gönderilir. Mobil uygulamada sonuçlar listelenir. Web uygulaması bölümünde mobil uygulamada kullanılmış olan analiz algoritmaları ile aynı algoritmalar kullanılarak

analiz yapılmak ve web tabanlı olarak da sistem kullanılabilmektedir. Aynı zamanda firmalar ürünlerini web uygulamasını kullanarak sisteme yükleyebilmektedir. Tez çalışması Şekil 5.1'in sağ bölümünde yer alan sunucuda gerçekleşen web uygulaması kapsamaktadır.

Mobil uygulama bölümünde mobil cihaz olarak Android 4.0.2 işletim sistemli Samsung Nexus marka mobil cihaz kullanılmıştır. Programlama dili olarak JAVA kullanılmıştır. Renk analizi bölümünde ise Skaler Nicemleme Algoritması kullanılarak sonuçlar elde edilmiştir [54].

5.2. Kullanılan Teknolojiler

Web uygulaması geliştirilirken Visual Studio ortamında C# dili ve Asp.Net kullanılmıştır. Veritabanı olarak ise Mssql ve Stored Procedure'ler kullanılmıştır. Asp.Net Microsoft tarafından geliştirilmiş bir uygulama geliştirme teknolojisidir. Dinamik web sayfaları, web uygulamaları ve XML tabanlı web hizmetleri oluşturulabilmesine olanak sağlar [55].

Asp.Net klasik Asp'den farklı olarak .Net Framework'ün altında IIS (Internet Information Server) ile bütünleşik çalışan bir mimari olup .NET Framework denilen çekirdek sistem üzerinde çalışmaktadır [56]. .Net Framework dağıtık uygulamaların geliştirilmesini ve farklı programlama dillerinin eş zamanlı olarak çalışabilmesini ve yürütülmesini destekleyen bir ortam sağlamaktadır. Bunu yapabilmek içinde CLR teknolojisini kullanır. CLR teknolojisi taşınılabilirliği sağlamaktadır. C# ile oluşturduğumuz bir programımızı derlediğimizde derleyici tarafından MSIL denilen bir kod oluşturulur. MSIL tarafından oluşturulmuş olan kodlar özel assembly kodları olarak nitelendirilebilirler. Bu ara kodun ne anlama geldiğini yani dönüşümünü CLR teknolojisi gerçekleştirildiğinden program çalıştırıldığında bu kodlar CLR tarafından çalıştırılır. Bu yüzden bu .Net teknolojisi ile oluşturulmuş olan kodlar CLR teknolojinin çalıştırılabildiği her ortamda çalıştırılabilirler. Uygulamanın farklı diller kullanılarak geliştirilmesi de mümkündür ancak öneri sistemi web tabanlı olarak çalışacağından Asp.Net teknolojisinin daha güvenilir ve daha hızlı olması sebebiyle web uygulaması .Net ortamında geliştirilmiştir.

Asp.Net teknolojisinin sağladığı pek çok avantaj bulunmaktadır. Bunlardan en önemlileri; programlama dilinden bağımsızdır, mobil programlama desteği sağlar, geliştirilmiş performans iyileştirmelerine sahiptir, basit ve güvenilirdir, kolay yönetilebilir bir ortam sağlar, kodlar isteğe bağlı olarak değiştirilebilir, ADO.net sayesinde bağlantısız bir veri iletişi ortamı sunar, çoklu dil desteği sağlar.

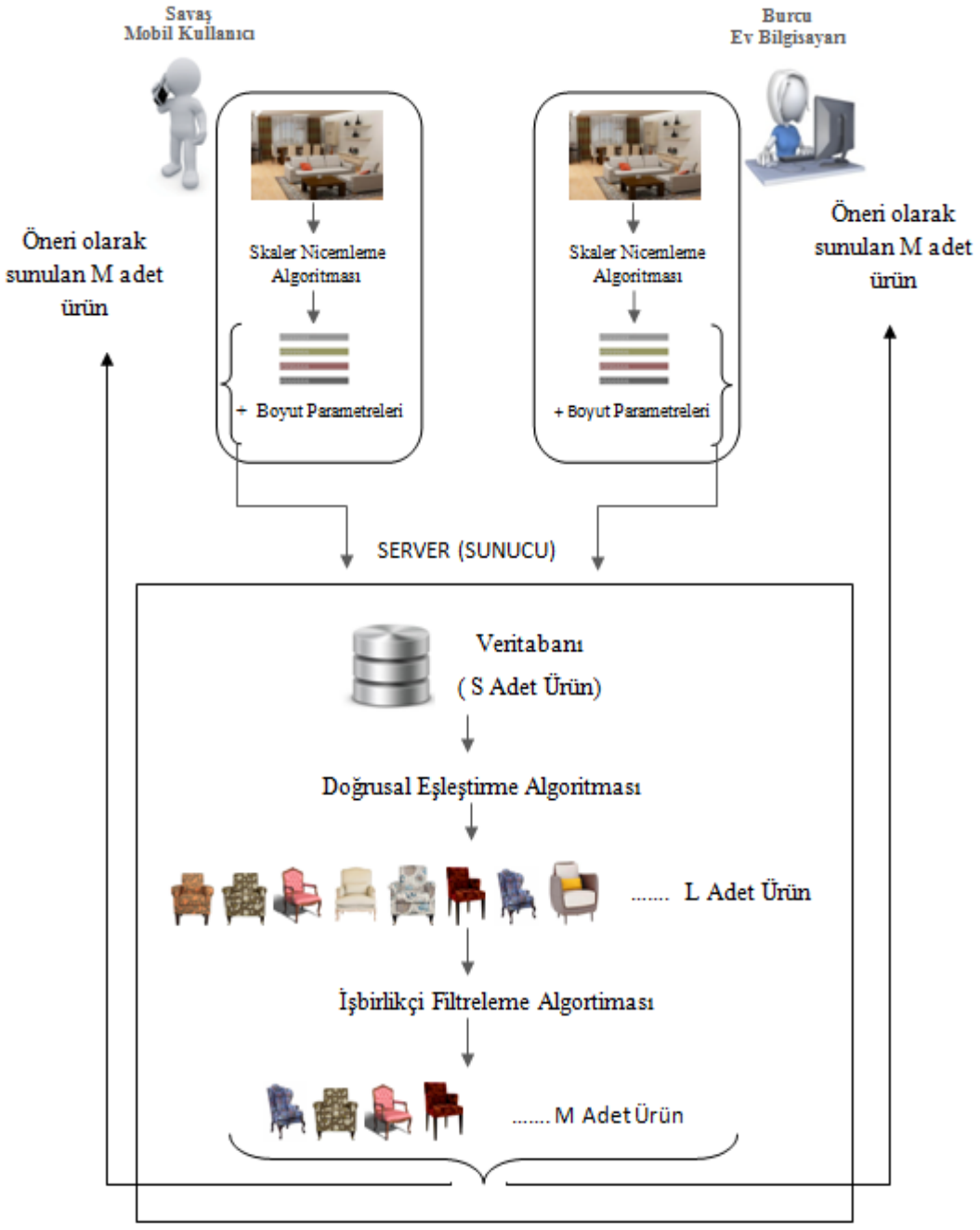
Veritabanında ise işlemlerin daha hızlı yapılabilmesi için Stored Procedure'ler kullanılmştır. Stored Procedure'ler Türkçe de saklı yordam olarak ifade edilirler. Sunucuda yer alan Sql Server içerisinde saklanırlar. Stored Procedure'ler diğer programlama dilleri ile benzer şekilde çalışırlar. Parametre alıp diğer programlama dillerindeki gibi parametre döndürebilirler ya da bir procedure içerisinde farklı procedurelerin çağırılmasını sağlayabilirler.

Stored Procedure kullanmanın pek çok faydası bulunmaktadır. Bunlardan en önemlileri hızlı bir şekilde çalışmalarıdır. İçerlerinde çok büyük miktarda kod bulundursalar bile ilk çalıştırıldıklarında derlenme ve optimizasyon işlemleri gerçekleştirilir ve daha sonraki kullanımlarında ise cache bellekten bilgileri alırlar. Bu yüzden ilk çalıştırmadan sonraki kullanımlarından sonra yeniden optimizasyon ve ayrıştırma işlemlerine zaman harcamazlar böylece çalışma zamanında çok daha hızlı yanıt almamızı sağlarlar. Ayrıca Stored Procedure'ler güvenlik mekanizmasına sahip olduklarından yetkisi olmayan kullanıcılar tarafından çalıştırılamazlar. En önemli özelliklerinden bir tanesi de internet trafiğini azaltmalarıdır. Bir Stored Procedure içerisinde pek çok T-Sql komutu bulunmaktadır ve bu komutların her biri tek tek çalıştırıldığında ayrı bir hattı meşgul edecektir. Fakat Stored Procedure içerisinde yer alan komutların bütünü tek bir komut gibi işlem görür ve böylece tek bir hat üzerinden çalıştırılır [57].

Uygulama geliştirilirken göz önünde bulundurulması gereken en önemli faktör kullanıcıya hızlı bir şekilde öneri sunabilmek olduğundan Asp.Net ve Stored Procedure teknolojilerinin hız bakımından diğer teknolojilere göre avantajları ve mobil uygulama destekleri göz önünde bulundurulmuştur ve bu yüzden sistemin geliştirilmesinde bu teknolojiler tercih edilmiştir.

5.3. Sunulan Öneri Modeli

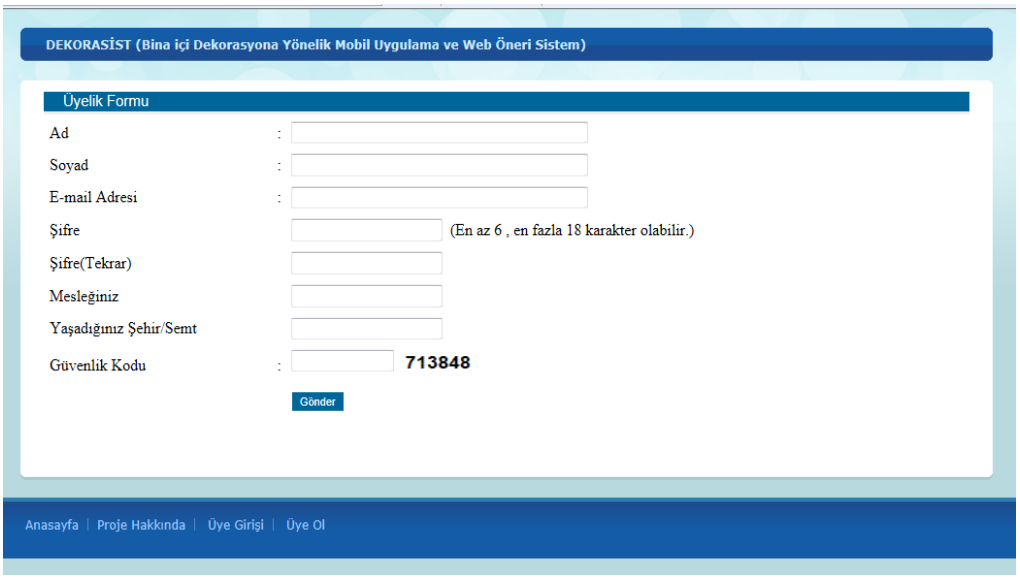
Dekorasist isimli uygulama web ve mobil uygulamadan oluşmaktadır. Web bölümünde öneri alabilmek için sisteme üye olunması gerekmektedir. Ayrıca web bölümünün firmalara ait ürün yükleyebilmeleri için bir yönetim paneli bulunmaktadır.



Şekil 5.2. İşbirlikçi Filtreleme Sistemi'nin genel yapısı [58]

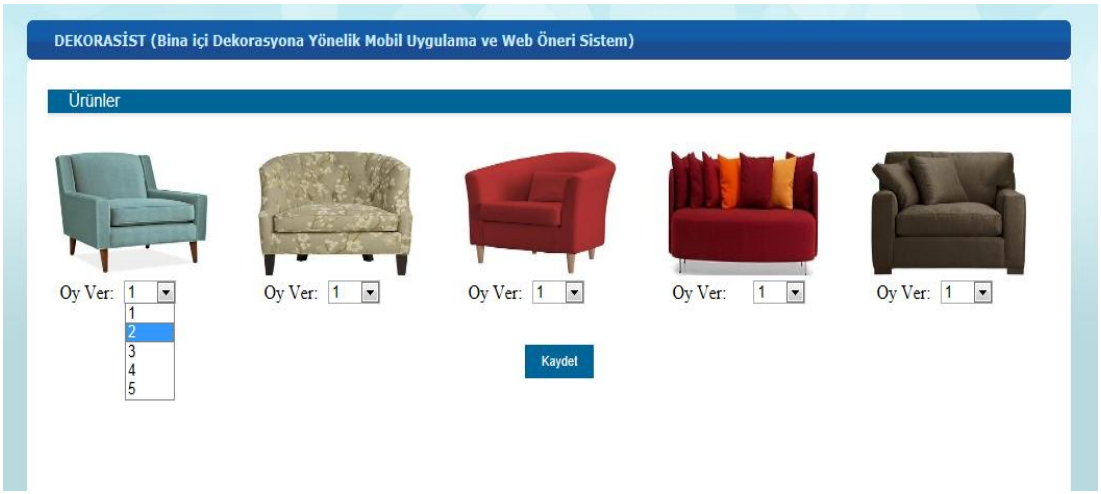
Uygulamanın genel görünümü Şekil 5.2'de gösterildiği gibidir. Kullanıcılar web üzerinden ya da mobil telefonları aracılığıyla sisteme giriş yaparak öneri alabilmektedir. İlk olarak web ortamından ya da telefon aracılığıyla yüklenmiş olan ortam resminin Skaler Nicemleme Algoritması kullanılarak analizi yapılmaktadır. Daha sonra analiz sonuçlarına boyut parametreleri (en, boy, yükseklik), firma bilgileri ve ürünün kategorisi de eklenerek sunucuya gönderilmektedir. Sunucu tarafında ise iki aşamalı bir filtreleme yöntemi uygulanmaktadır. İlk olarak Doğrusal Eşleştirme Yöntemi kullanılarak veritabanında yer alan S adet ürün içerisinden ton olarak en benzer olan L adet ürün filtrelenmektedir. Bu filtreleme işlemi sırasında firma ve boyut bilgileri de göz önünde bulundurulmaktadır. Daha sonra filtrelenmiş olan L adet ürün İşbirlikçi Filtreleme Algoritması'ndan geçirilmektedir. Böylece kullanıcıların daha önceki seçimleri de göz önünde bulundurularak M adet ürün kullanıcıya sunulmaktadır. İşbirlikçi Filtreleme Algoritması'nda kullanıcının geçmiş bilgileri olarak daha önce puan vermiş olduğu ürünler dikkate alınmaktadır. Öneri alma işlemi aşağıda adım adım gösterilmiştir.

Adım 1: Sisteme üye olunması: Sistemin kullanılabilmesi için ilk olarak sisteme üye olunması gerekmektedir. Sisteme üye olurken kullanıcıya ait ad, soyad, meslek, yaşanılan şehir gibi çeşitli bilgiler alınmaktadır. Üyelik sayfasının tasarımı Şekil 5.3'de gösterilmiştir.



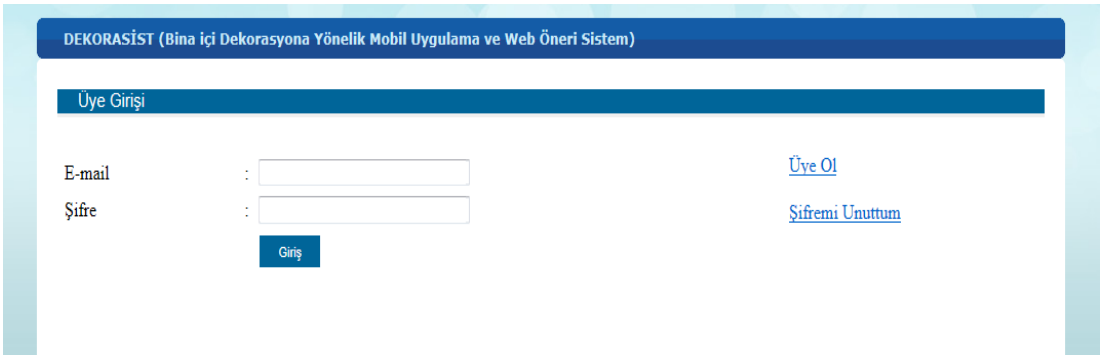
Şekil 5.3. Sisteme üye olabilmek isteyen kullanıcıların karşılaşacağı sayfa

Adım 2: Ürün değerlendirme sayfası: Kullanıcılar sisteme üye olduklarında veritabanından rastgele olarak seçilmiş olan en az 20 adet ürün puanlamaları için karşılarına çıkarılmaktadır. Böylece kullanıcılara öneri sunulacağı zaman daha önce oy vermiş olduğu ürünler değerlendirilerek kullanıcıya en çok hitap edecek M adet ürün öneri olarak sunulmaktadır. Ürün değerlendirme işleminin gerçekleştirileceği sayfa Şekil 5.4'de gösterilmiştir. Puan verme işlemi 1-5 aralığında gerçekleştirilmektedir. 1 puanı en düşük puanı ve 5 puanı da en iyi (en yüksek) puanı ifade etmektedir.



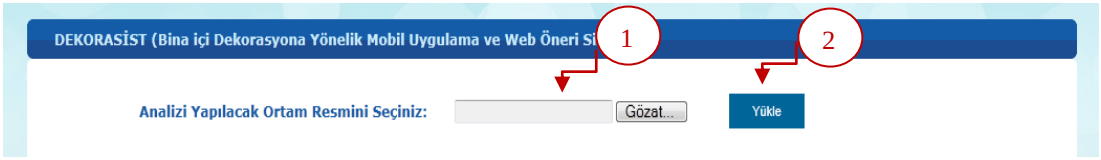
Şekil 5.4. Kullanıcılar tarafından ürünlerin değerlendirildiği sayfa

Adım 3: Sisteme giriş yapma: Sistemdeki kullanıcıların öneri alabilmeleri için üye olduktan sonra sisteme giriş yapmaları gerekmektedir. Giriş işlemi kullanıcı adı ve şifre kullanılarak yapılmaktadır. Kullanıcı adı yerine e-mail adresi kullanılmaktadır. Sayfanın tasarımı Şekil 5.5'de gösterilmiştir.



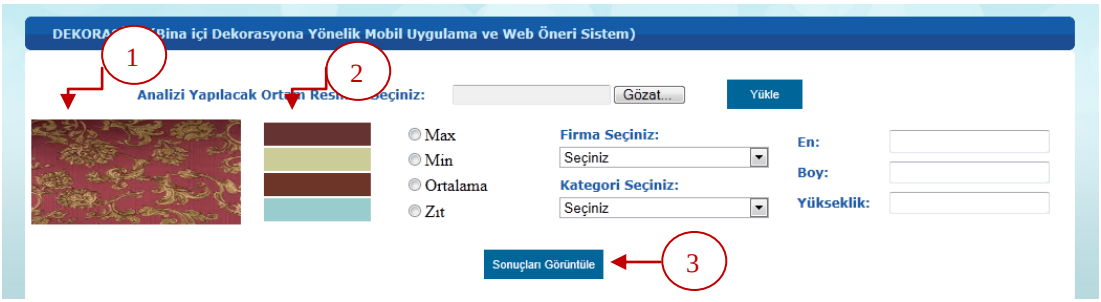
Şekil 5.5. Üye giriş sayfası

Adım 4: Öneri alma işlemi: Sisteme giriş yapıldıktan sonra karşılaşılan sayfadır. Bu sayfada kullanıcıya öneri olarak ürünler sunulmaktadır. Şekil 5.6'da öneri sayfasına ait giriş ekranı yer almaktadır.



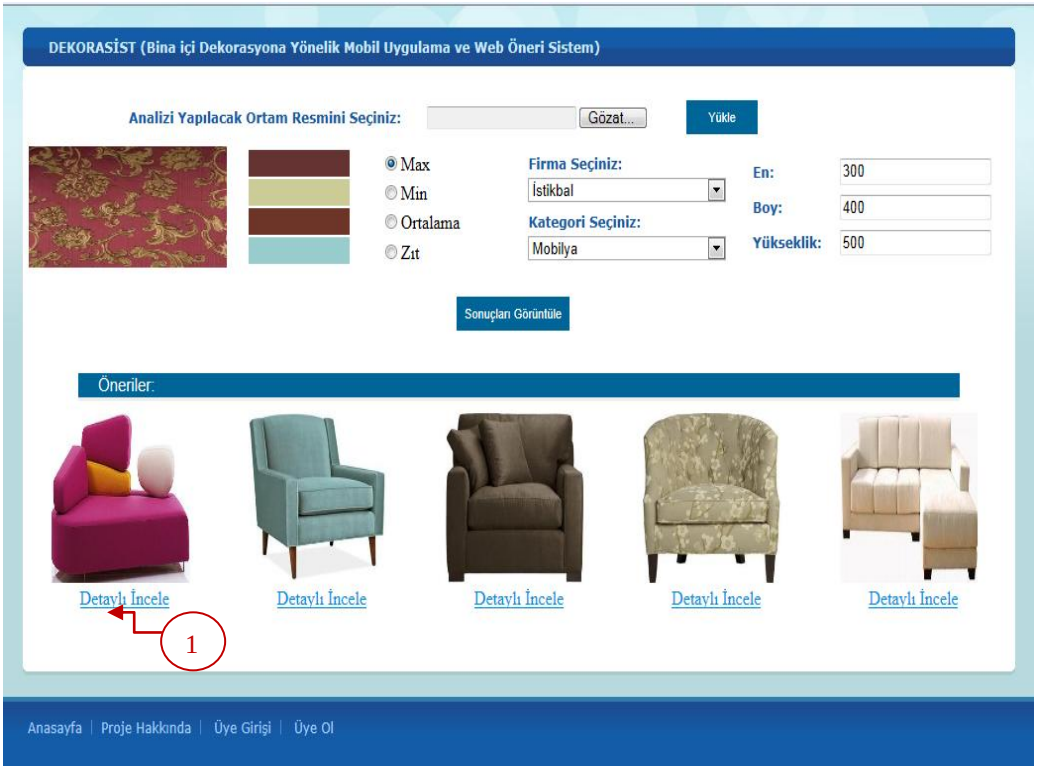
Şekil 5.6. Öneri sayfası giriş ekranı

Kullanıcılar 1 numara olarak gösterilen dosya yükleme butonunu kullanarak analizi yapılacak ortamı sisteme yükleyerek 2 numaralı butona bastıklarında Şekil 5.7'de gözüken analiz sonuçları listelenmektedir.



Şekil 5.7. Öneri işlemi analiz sonuçları

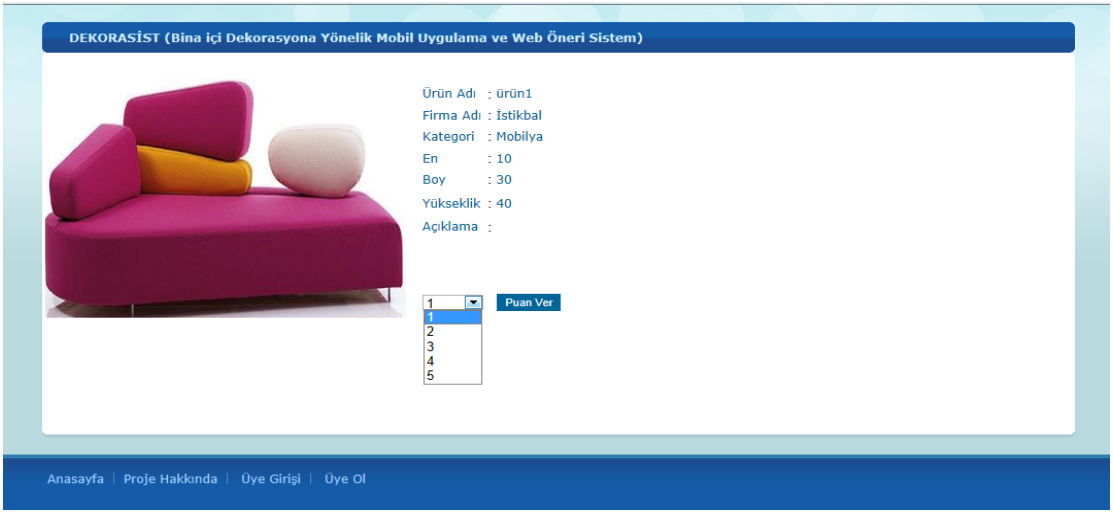
Şekil 5.7'de 1 numara ile gösterilen resim kullanıcı tarafından analizi yapılmak için sisteme yüklenmiş olan ortam resmini göstermektedir. 2 numara ile gösterilen alan ise 1 numara ile gösterilen resme ait analiz sonuçlarının listesidir. Bu liste içerisinde ortamda yer alan renk tonlarının maksimumu, minimumu, ortalama değeri ve zıt tonu yer almaktadır. Bu alan kullanıcının seçimine bağlıdır. Seçilen tona en yakın benzerliğe sahip olan 5 adet ürün listelenecektir. Renk tonlarının sağ yanında firma ve kategori seçenekleri de yer almaktadır. Ayrıca ortamın en, boy ve yükseklik değerlerinin girilebileceği alanlarda bulunmaktadır. Buradaki seçenekler seçildikten sonra 3 numaralı butona basıldığında Şekil 5.8'de görülen öneriler görüntülenecektir.



Şekil 5.8. Öneri sayfası

Öneriler bölümünde listelenmiş olan ürünler Doğrusal Eşleştirme ve İşbirlikçi Filtreleme Yöntem'leri birlikte kullanılarak elde edilmiş nihai sonuçları göstermektedir. Öneri olarak sunulan ürünlerin altında yer alan, 1 numarala ile gösterilen detaylı incele linki ile kullanıcılar ürünleri detaylı olarak görüntüleyebileceği ürün detay sayfasına yönlendirilecektir.

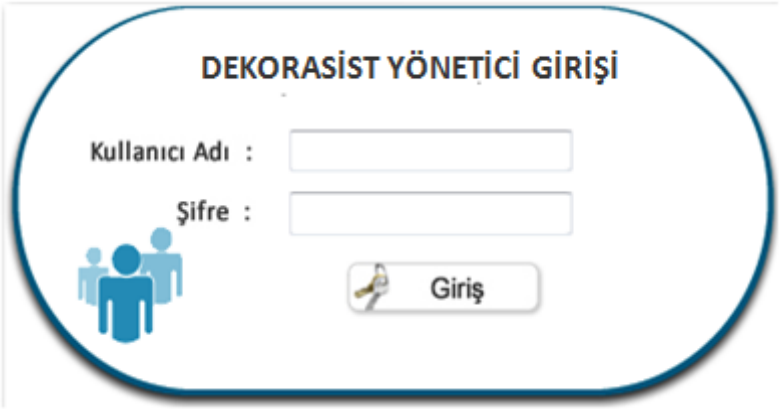
Adım 5: Ürünleri detaylı inceleme: Şekil 5.8'de gösterilen 4 numaralı linke tıklandığında ilgili ürüne ait detaylı bilgilerin yer alacağı Şekil 5.9'da gösterilen ürün detay sayfası görüntülenecektir. Burada ürüne ait firma, kategori, en,boy ve yükseklik değerleri görüntülenmektedir. Ayrıca ürüne puan verilebilmesi için bir buton yer almaktadır. Ancak ürüne puan verilebilmesi için üye girişi yapılması gerekmektedir. Üye girişi yapılmadan puan verilmek istenirse sistem kullanıcıyı otomatik olarak üye giriş sayfasına yönlendirecektir.



Şekil 5.9. Ürünlere ait bilgilerin görüntülendiği sayfa

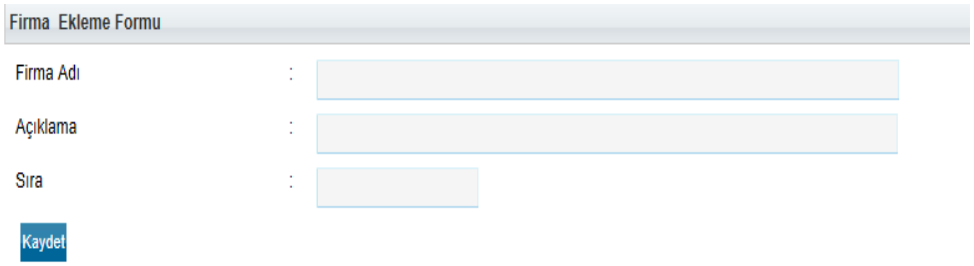
Web uygulamasında öneri üretme bölümü haricinde firmaların ürünlerini yükleyebilmeleri için bir yönetim paneli de bulunmaktadır. Yönetim panelinin kullanımı aşağıda adım adım açıklanmıştır.

Yönetim paneli giriş sayfası: Panele giriş yapabilmek için kullanıcı adı ve şifre bilgileri gereklidir. Yönetim paneline giriş yetkisi sistemin ana yöneticisi tarafından sadece yetkili firmalara verilmektedir. Yönetim paneli giriş sayfası Şekil 5.10'de gösterilmiştir.



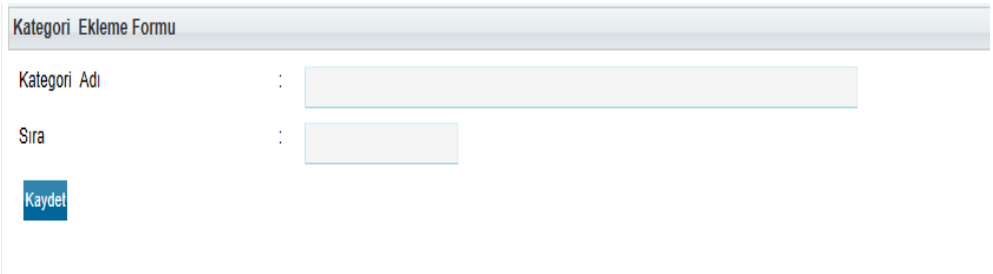
Şekil 5.10. Yönetim paneline giriş sayfası

Firma ekleme: Sisteme farklı dekorasyon firmalarının eklenebileceği firma ekleme sayfası bulunmaktadır. Sayfanın görünümü Şekil 5.11'de gösterilmiştir. Firma ekleme işlemi sadece sistem yöneticisi tarafından gerçekleştirilebilmektedir.



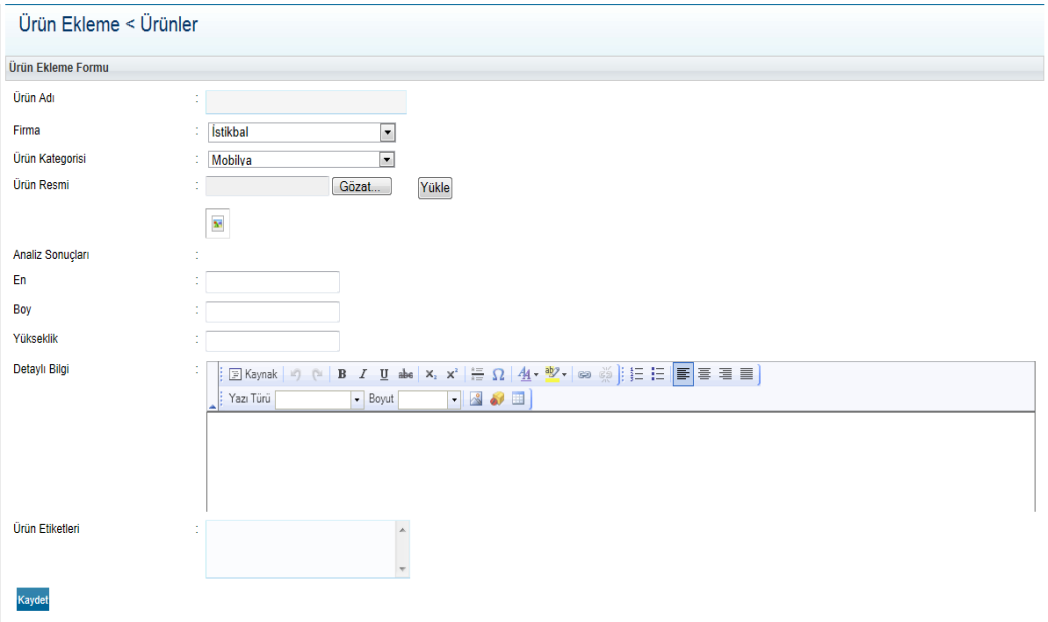
Şekil 5.11. Firma ekleme sayfası

Kategori ekleme: Sisteme firmalar tarafından eklenecek olan ürünlere mobilya, koltuk, tekli koltuk gibi kategorilerin eklenebileceği kategori ekleme sayfası bulunmaktadır. Kategori ekleme işlemi de firma ekleme işlemi gibi sadece sistem yöneticisi tarafından gerçekleştirilebilmektedir. Şekil 5.12'de kategori ekleme sayfası gösterilmiştir.



Şekil 5.12. Kategori ekleme sayfası

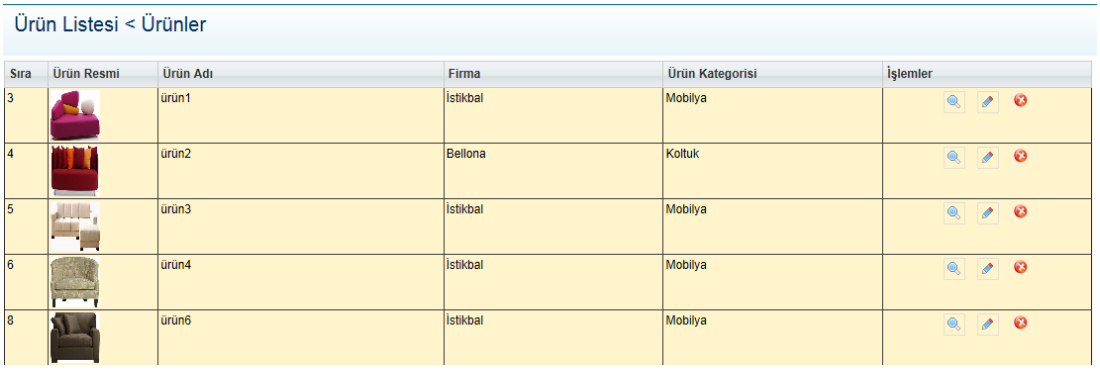
Ürün ekleme: Firmalara ürünlerini yükleyebilmelerini sağlayacak bir panel sunulmaktadır. Panelde ürüne ait firma, kategori, ürün resmi, en, boy ve yükseklik değerleri, ürün etiketleri gibi bilgiler yer almaktadır. Ürün ekleme sayfasının ekran görüntüsü Şekil 5.13'de gösterilmiştir.



Şekil 5.13. Ürün ekleme sayfası

Ürün ekleme sayfasından ürüne ait resim yüklendiği zaman ürünün skaler nicemleme yöntemi kullanılarak analizi yapılmaktadır. Analiz işlemi sonrasında sonuçlar veritabanına kaydedilmektedir. Böylece her ürün için analiz işlemi ürünün yükleme anında yapılarak öneri üretme aşamasında sistemin gereksiz yere çalıştırılması önlenmektedir.

Ürün listeleme: Sistemde yer alan ürünlerin listelendiği sayfadır. Ürün listeleme sayfası Şekil 5.14'de gösterilmiştir. 1 numaralı buton kullanılarak kayıtlı olan ürünün düzenleneceği sayfa görüntülenebilir ya da 2 numaralı buton kullanılarak ürünün silinmesi işlemi gerçekleştirilebilmektedir.



Şekil 5.14. Sistemde kayıtlı olan ürünlerin tamamının listelendiği sayfa

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuç

Öneri sistemlerinde yapılan gelişmeler her geçen gün artarak devam etmektedir. Bu çalışma kapsamında da İşbirlikçi Filtreleme Yöntemi kullanılarak bir öneri sistemi tasarlanmıştır. Çalışma sonucunda İşbirlikçi Filtreleme Yöntemi'nin mobilya ve geliştirilmiş olan öneri sistemine başarılı bir şekilde uygulanabileceği gösterilmiştir. Yöntemler performans zaman ve doğruluk oranları gibi farklı yönlerden test edilmiştir ve yöntemin uygunluğu kanıtlanmıştır.

Geliştirilmiş olan sistem kullanıcılara en kısa sürede ilgilerini çekebilecek ürünleri bulma konusunda yardımcı olmaktadır. Ayrıca oluşturulmuş olan sistem geliştirilmeye açık bir sistemdir. Uygulama sayesinde firmalarına ürünlerini müşterilerine ulaştırabilmeleri için alternatif bir yol sağlanmıştır. Böylece dekorasyon firmaları teknolojiyi etkin bir şekilde kullanmış olacaklar hem de ürünlerin satışı ve tanıtılması bakımından fayda sağlanacaktır.

Uygulama sayesinde İşbirlikçi Filtreleme Yöntemi farklı bir alana uygulanmıştır. Bu yüzden mobilya ve dekorasyon alanında yapılacak olan benzer çalışmalara öncülük etmektedir.

6.2. Yapılacak Çalışmalar

İleride sistemin iki farklı şekilde geliştirilmesi sağlanabilir. Bunlar ürün tabanlı ve kullanıcı tabanlı olarak yapılacak geliştirmelerdir.

Ürün tabanlı olarak yapılacak geliştirme bölümünde ürünlerin etiketleri ve temel özellikleri de dikkate alınarak İşbirlikçi Filtreleme Yöntemine İçerik Tabanlı Filtreleme Yöntemi'de eklenebilir. Böylece kullanıcıların seçimleriyle birlikte seçilen ürünlerin nitelikleri de dikkate alınacaktır. Böylece İşbirlikçi Filtreleme

Yöntemi ile güncel veriler kullanılarak öneri verilirken ürünlerin niteliklerinin de işleme alınması öneri kalitesini arttırabilir.

Kullanıcı seçiminde yapılabilecek çalışma ise, sistem kullanıcı sayısındaki artışı kaldıramayacak duruma geldiğinde gerçekleştirilebilir. Kullanıcılar seçimlerindeki çeşitli özellikler dikkate alınarak gruplandırılabilirler ve daha sonra benzerlikler hesaplanırken sadece kullanıcının bulunduğu gruptaki komşularıyla benzerlikleri hesaplanarak işlem süresinin azaltılması ve sistem performansının arttırılması sağlanabilir.

Ayrıca daha sonraki çalışmalarda mobilya ürünlerinin üç boyutlu olarak kullanıcı tarafından ortama yerleştirilmesi, ortam içerisinde yerinin değiştirilebilmesi sağlanabilir. Kullanıcı tarafından yüklenmiş olan ortam resimlerinin görüntü kalitesi farklılık göstereceğinden görüntü üzerinde çeşitli teknikler uygulanarak görüntünün kalitesini arttırmaya yönelik çalışmalar da yapılabilir.

KAYNAKLAR

[1] Özcan A., Sosyal ağ verisi kullanarak mobil telefonlar için öneri altyapısı tasarlanması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2010, 293819.

[2] Firan C. S., Nejdl W., Paiu R., The benefit of using tag-based profiles, *Fifth Latin American Web Congress*, Santiago de Chile, Chile, 31 October-2 November 2007.

[3] Yu X., Sun S., Research on personalized recommendation system based on web mining, *International Conference on E-Business and E-Goverment*, DC, USA, 7-9 May 2010.

[4] Pazzani M. S., A framework for collaborative, contend-based and demographic filtering, *Artifical Intelligence Review*, 1999, **13,** 393-408.

[5] Weiyang L., Alvarez S., Ruiz C., Efficient adaptive-support association rule mining for recommender systems, *Data Mining and Knowledge Discovery*, 2002, **6,** 83-105.

[6] Shani G., Heckerman D., Brafman R., An MDP-based recommender system, *Journal of Machine Learning Research, ,* 2005, **6** ,1265-1295.

[7] Yan T., Jacopsen W., Garcia-Molina M., Dayal H. U., From user accesspatterns to dynamic hypertext linking, *The Fifth World Wide Web Conference*, Paris, Fransa, 6-10 May 1996.

[8] Mobasher B., Dai H., Luo T., Nakagawa M., Using sequential and non-sequential patterns in predictive web usage mining tasks, *IEEE International Conference on Data Mining*, Maebashi City, Japan, 9-12 December 2002.

[9] Miller B. N., Toward a personal Recommender System, Phd Thesis, University of Minnesota, USA.

[10] Breese J. ,Heckerman D., Kadie C., Empirical analysis of predictive algorithms for collaborative filtering, *14th Conference On Uncertainty in Artificial Intelligence,* Madison, USA, 26-30 July 1998.

[11] Basu C., Hirsh H., Cohen W., Recommendation as classification: using social and content-based information in recommendation, *Fifteenth National Conference on Artifical Intelligence*, Monona Terrace, USA, 26-30 July 1998.

[12] Ungar L. H., Foster D.P., Clustering methods for collaborative filtering, *Workshop on Recommender Systems, 15th National Conference on Artificial Intelligence*, Monona Terrace, USA, 26-30 July 1998.

[13] Sarwar B. M., Karypis G., Konstan J. A., Riedi J., Application of dimensionality reduction in recommender system-A case study, *ACM WebKDD Workshop at the ACM-SIGKDD Conference on Knowledge Discovery in Databases*, Boston, Massachusetts, USA, 20-23 August 2000.

[14] Pennock D. M., Horvitz E., Lawrence S., Giles C. L., Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach, *16th Conference on Uncertainy in Artificial Intelligence*, Stanford, CA, USA, 30 June-3 July 2000.

[15] Miyahara K., Pazzani M. J., Improvement of collaborative filtering with the simple bayesian classifier, *Transaction of Information Processing Society of Japan*, 2002, **43**(11), 3429-3437.

[16] Lian R., The construction of personalized web page recommendation system in e-commerce, *International Conference on Computer Science and Service system*, Weifang, China, 27-29 June 2011.

[17] Resnick P., Lacovou N., Suchak M., Bergstrom P., Riedl J., Grouplens: An open architecture for collaborative filtering of netnews, *ACM Conference on Computer-Supported Cooperative Work*, North Carolina, USA, 22-26 October 1994.

[18] Sarwar B. M., Konstan J. A., Borchers A., Herlocker J., Miller B., Riedl J., Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system, *ACM Conference on Computer Supported Cooperative Work*, Seattle, WA, USA, 14-18 November 1998.

[19] Wittenburg K., Duco D., Will H., Larry S., Group Asynchronous Browsing on the World Wide Web, http://www.w3.org/Conferences/WWW4/Papers/98/ (Ziyaret Tarihi:12.10.2012).

[20] Herlocker L. J., Konstan J. A., Borchers A., Riedl J., An algorithmic framework for performing collaborative filtering, *22nd Annual International ACM SIGIR Conference*, Berkeley, CA, USA, 15-18 August 1999.

[21] Sarwar B., Karypis G., Konstan J., Rield J., Analysis of recommendation algorithms for e-commerce, *2nd ACM Conference on Electronic Commerce*, Minnesota, USA, 17-20 October 2000.

[22] Fisher D., Hidrum K., Hong J., Newman M., Thomas M., Vuduc R., SWAMI: Framework for collaborative filtering algorithm development and evaluation, *23rd Annual International SCM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 24-28 July 2000.

[23] Goldberg K., Roeder T., Grupta D., Perkins C., Eigenstate:A constant time collaborative filtering algorithm, *Information Retrieval*, 2001, **4**(2), 133-151.

[24] Hu R., Lu Y., A hybrid user and item-based collaborative filtering with smoothing on sparse data, *Artifical Reality and Telexistence*, China, 29 November-1 December 2006.

[25] Chen J., Yin J., A collaborative filtering recommendation algorithm based on influence sets, *Ruan Jian Xue Bao/Journal of Software*, 2007, **18**, 1685-1694.

[26] Chandrashekhar H., Bhasker B., Collaborative filtering based on the entropy measure, *4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services*, Tokyo, Japan, 23-26 July 2007.

[27] Baltrunas L., Amatriain X., Towards time dependent recommendation based on implicit feedback, *3rd ACM Conference on Recommender Systems*, NewYork, USA, 22-25 October 2009.

[28] Baltrunas L., Ricci F., Context-dependent items generation in collaborative filtering, *3rd ACM Conference on Recommender Systems*, New York, USA, 22-25 October 2009.

[29] Rafter R., Evaluation and conversation in collaborative filtering, Phd Thesis, University College Dublin, Dublin, Ireland, 2010.

[30] Wang Z., Wang Y., Wu H., Tags meet ratings:Improving collaborative filtering with tag-based neighborhood method, *Workshop on Social Recommender Systems*, Hong Kong, China, 7 February 2010.

[31] Hang C. , Meifang Z., Improve tagging recommender system based on tags semantic similarity, *3rd International Conference on Communication Software and Network*, China, 27-29 May 2011.

[32] Özkan Y., *Veri Madenciliği Yöntemleri*, 1. baskı, Papatya Yayıncılık Eğitim, İstanbul, 2008.

[33] Alan M. A., Veri madenciliği ve lisansüstü öğrenci verileri üzerine bir uygulama, *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi,* 2012, **33**, 165-174.

[34] Weiss S. M., Zhang T., *Performance analysis and evaluation, The handbook of data minig*, Lawrence Erlbaum Associates, Nong Ye Arizona State University, Mahwah, 2003.

[35] Han J., Kamber M., *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, USA, 2011.

[36] Bramer M., *Principles of Data Mining*, Springer, London, 2007.

[37] Rohini U., Ambati V., Improving re-ranking of search results using collaborative filtering, *Third Asia Conference on Information Retrieval Technology*, Singapore, 16-18 October 2006.

[38] Avancini H., Candela L., Straccia U., Recommenders in a personalized, collaborative digital library environment, *Journal of Intelligent Information Systems*, 2007, **1**, 253-283.

[39] Cai X., Bain M., Krzywicki A., Wobcke W., Kim Y. S., Compton P., Mahidadia A., Learning collaborative filtering and its application to people to people recommendation in social networks, *10th IEEE International Conference on Data Mining*, Sydney, Australia, 14-17 December 2010.

[40] Campos L. M., Luna J. M. F., Huete J. F., Morales M. A. R., Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian Networks, *International Journal of Approximate Reasoning*, 2010, **51**(7), 785-799.

[41] Şakiroğlu A. M., Web siteleri için yapay bağışıklık tabanlı bir öneri sistemi, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2005.

[42] Kaya N., Recommender system framework based on data mining techniques, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, 2011.

[43] Castelluccio M., M. G. P. the music genemo project, *Strategic Finance 88*, 2006, **6**, 57-58.

[44] Joyce J., Pandora and the music genome project, *Scientific Computing 23*, 2006, **10,** 10-41.

[45] Shardanand U., Maes P., Social information filtering: Algorithms for automating "word of mouth", *ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, 14-18 April 1995.

[46] Linden G., Smith B., York J., Amazon.com recommendations: item-to-item Collaborative Filtering, *IEE Internet Computing* ,2003, **7**, 76-80.

[47] Goldberg D., Nichols D. M., Oki B., Terry D., Using collaborative filtering to weave an information Tapestry, *Communications of the ACM* , 1992, **35**, 61-70.

[48] Khabbaz M., LaksManan L., Top Recs: Top-k algorithms for item based collaborative filtering, *14th Internetional Conference on Extending Database Technology*,Uppsala, Sweden, 21-25 March 2011.

[49] Rafter R., Coyle L., Nixon P., Smyth B., Sticking with a winning team: Better neighbour selection for conversational collaborative recommendation, *18th Irish Conference on Artifical Intelligence and Cognitive Science*, Dublin, Ireland, 29-31 August 2007.

[50] Redpath J., Glass D. H., McClean S., Chen L., User based collaborative filtering: sparsity and performance, *5th Starting AI Researchers' Symposium*, Lisbon, Protugal, 16-20 August 2010.

[51] Sarwar B. M., Karypis G., Konstan J. A., Rield J., Item based collaborative filtering recommendations algorithms, *10th Internatioanl World Wide Web Conference*, Hong Kong, 1-5 May 2001.

[52] http://mars.wiwi.huberlin.de/mediawiki/teachwiki/index.php/Performance_of_Classification_Methods_-_A_Monte_Carlo_Study2 (Ziyaret Tarihi: 21 Ekim 2012).

[53] http://movielens.umn.edu (Ziyaret Tarihi: 11 Temmuz 2012).

[54] Yıldız G., Renk analizi tabanlı öneri sistemleri için mobil uygulama geliştirme, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2012.

[55] http://tr.wikipedia.org/wiki/ASP.NET (Ziyaret Tarihi: 01 Kasım 2012).

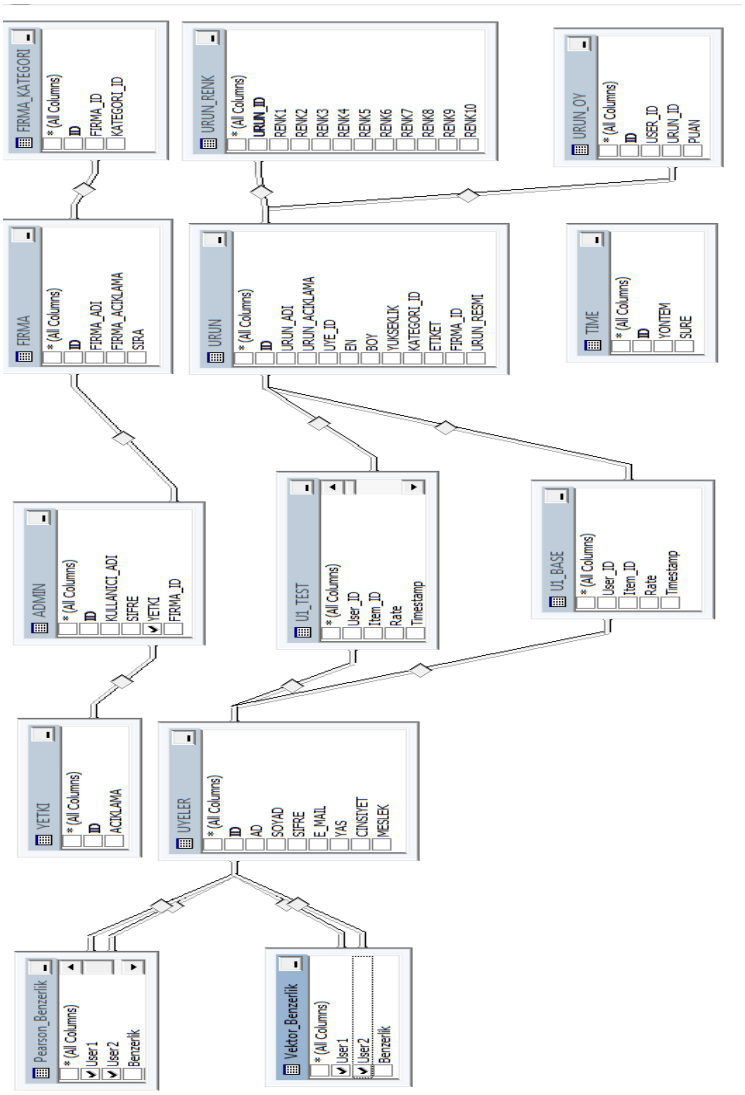
[56] Garipcan A. M., Varol A., Asp.Net teknolojisi ile web tabanlı bir e-ticaret uygulamasının geliştirilmesi, *Elektrik-Elektronik ve Bilgisayar Sempozyumu,* Elazığ, 5-7 Ekim 2011.

[57] http://msdn.microsoft.com/en-us/library/aa214299(SQL.80).aspx (Ziyaret Tarihi : 01 Kasım 2012).

[58] Kaya F., Yıldız G., Kavak A., Evalution of collaborative filtering techniques for mobile and web application based recommendation system, *9th International Conference on Electronics, Computer and Computation*, Ankara, Turkey, 1-3 November 2012.

EKLER

Ek−A İLİŞKİSEL VERİTABANI DİYAGRAMI



Şekil A.1. Geliştirilmiş olan ilişkisel veritabanı diyagramı

Ek−B GELİŞTİRİLEN SİSTEM İÇİN YAZILAN KAYNAK KODLAR

KOD BÖLÜMLERİ	
Pearson_Similarity	Pearson Benzerliğini kullanarak kullanıcılar arasındaki benzerlikleri hesaplar
Vector_Similarity	Vector Benzerliğini kullanarak kullanıcılar arasındaki benzerlikleri hesaplar
KNN_Algorithm	KNN Algortiması ile komşuları seçer
Correlation_Threshold_Algorithm	Correlation Threshold Algortiması ile komşuları seçer
Hkok	Hkok ile hata hesaplar
Mho	Mho ile hata hesaplar

Pearson_Similarity

```
#region pearson_similarity
        System.Diagnostics.Stopwatch diff = new
System.Diagnostics.Stopwatch();
        diff.Start();
        DataTable dt = DtBase.DT("pearsonSimilarity1");
        double aktifKullanici_ortalama_oy = 0;
        double sKullanicisi_ortalama_oy = 0;
        for (int i = 0; i < 943; i++)
        {
            aktifKullanici_ortalama_oy =
Convert.ToDouble(dt.Rows[i]["ortalama"].ToString());
            DataTable dt2 = DtBase.DT("pearsonSimilarity2",
Convert.ToDouble(dt.Rows[i]["User_ID"].ToString()));
            int s = 0;
            double pay = 0, payda1 = 0, payda2 = 0, payda;
            for (int sayac = 0; sayac < dt2.Rows.Count; sayac++)
            {
                s = Convert.ToInt32(dt2.Rows[sayac]["User_ID1"].ToString());
                for (int m = 0; m < dt.Rows.Count; m++)
                {
                    if (s ==
Convert.ToInt32(dt.Rows[m]["User_ID"].ToString()))
                    {
                        sKullanicisi_ortalama_oy =
Convert.ToDouble(dt.Rows[m]["ortalama"].ToString());
                        break;
                    }

                }
                pay = 0;
                payda1 = 0;
                payda2 = 0;
                while (true)
```

```
                {
                    if ((sayac + 1) >= dt2.Rows.Count)
                        break;

                    pay                                                                 +=
(Convert.ToInt32(dt2.Rows[sayac]["Rate"].ToString())                                    -
aktifKullanici_ortalama_oy)                                                             *
(Convert.ToInt32(dt2.Rows[sayac]["Rate1"].ToString())                                   -
sKullanicisi_ortalama_oy);
                    payda1                                                              +=
(Convert.ToInt32(dt2.Rows[sayac]["Rate"].ToString())                                    -
aktifKullanici_ortalama_oy)                                                             *
(Convert.ToInt32(dt2.Rows[sayac]["Rate"].ToString())                                    -
aktifKullanici_ortalama_oy);
                    payda2                                                              +=
(Convert.ToInt32(dt2.Rows[sayac]["Rate1"].ToString())                                   -
sKullanicisi_ortalama_oy)                                                               *
(Convert.ToInt32(dt2.Rows[sayac]["Rate1"].ToString())                                   -
sKullanicisi_ortalama_oy);

                    if            (Convert.ToInt32(dt2.Rows[sayac                       +
1]["User_ID1"].ToString())                                                              !=
Convert.ToInt32(dt2.Rows[sayac]["User_ID1"].ToString()))
                    {

                        break;
                    }
                    sayac++;

                }

                payda = System.Math.Sqrt(Convert.ToDouble(payda1 * payda2));
                double sonuc = 0;
                sonuc = pay / payda;

                if (pay == 0.0 || payda == 0.0)
                {
                    sonuc = 0;
                }

                string    sql1    =    "INSERT    INTO    Pearson_Similarity
(User1,User2,Benzerlik)  VALUES(" + dt.Rows[i]["User_ID"].ToString() + "," +
s + "," + String.Format("{0:0.##########}", sonuc).Replace(",",".") + ")";
                //Sonuçları Veritabanına Kaydediyoruz.
            }
        }

        #endregion
```

Vector_similarity

```
#region vector_similarity
        DataTable dt3 = DtBase.DT("vectorSimilarity1");
        for (int i = 0; i < 943; i++)
        {
            int                          anakullanıcı                          =
Convert.ToInt32(dt3.Rows[i]["User_ID"].ToString());
            DataTable dt4 =DtBase.DT("vectorSimilarity2", anakullanıcı);
            double toplam = 0;
            for (int n = 0; n < dt4.Rows.Count; n++)
            {

                double toplam1 = 0, toplam2 = 0;
                toplam = 0;
                int                            suserid                           =
Convert.ToInt32(dt4.Rows[n]["User_ID1"].ToString());
                double snormalizasyon = 0;
                for (int m = 0; m < dt3.Rows.Count; m++)
                {
                    if                          (suserid                        ==
Convert.ToInt32(dt3.Rows[m]["User_ID"].ToString()))
                        snormalizasyon                                          =
Convert.ToDouble(dt3.Rows[m]["NORMALIZASYON"].ToString());
                }

                while (true)
                {
                    if ((n + 1) == dt4.Rows.Count)
                        break;

                    if    (Convert.ToInt32(dt4.Rows[n]["User_ID1"].ToString())
!= Convert.ToInt32(dt4.Rows[n + 1]["User_ID1"].ToString()))
                    {

                        break;
                    }
                    toplam1                                                    +=
Convert.ToDouble(dt4.Rows[n]["Rate"].ToString())                                *
Convert.ToDouble(dt4.Rows[n]["Rate1"].ToString());
                    n++;
                }
                toplam2                                                         =
Convert.ToDouble(dt3.Rows[i]["NORMALIZASYON"].ToString()) * snormalizasyon;
                toplam = toplam1 / toplam2;
                if (n == dt4.Rows.Count)
                    break;
                string     sql1     =     "INSERT     INTO     Vektor_Similarity
(User1,User2,Benzerlik)        VALUES("    +    (i    +    1)    +    ","    +
Convert.ToInt32(dt4.Rows[n]["User_ID1"].ToString())       +       ","       +
String.Format("{0:0.##########}", toplam).Replace(",", ".") + ")";
                //Sonuçları Veritabanına Kaydediyoruz.
            }
        }
        #endregion
```

KNN_Algorithm

```
#region KNN_Algorithm

        sql = "SELECT * FROM U1_TEST";
        dt = new DataTable();
        DataTable userdt = new DataTable();
        DtBase.DT(dt, sql);
        for (int i = 0; i < dt.Rows.Count; i++)
        {

            userdt = new DataTable();
            string sql1 = "";
            try
            {
               //En benzer 100 Kullanıcıyı seçiyoruz. Userdt Datatable'ına
aktarıyoruz.
                if (userdt.Rows.Count != 0 &&
userdt.Rows[0]["SONUC"].ToString() != null &&
userdt.Rows[0]["SONUC"].ToString() != "")
                {
                    sql1 = "UPDATE U1_TEST SET [KNN]=" +
String.Format("{0:0.##########}",
Convert.ToDouble(userdt.Rows[0]["SONUC"].ToString())).Replace(",", ".") + "
WHERE User_ID=" + dt.Rows[i]["User_ID"].ToString() + " AND Item_ID=" +
dt.Rows[i]["Item_ID"].ToString();

                }
                else
                {
                    sql1 = "UPDATE U1_TEST SET [KNN]=0 WHERE User_ID=" +
dt.Rows[i]["User_ID"].ToString() + " AND Item_ID=" +
dt.Rows[i]["Item_ID"].ToString();

                }


            }
            catch
            {
                sql1 = "UPDATE U1_TEST SET [KNN]=0 WHERE User_ID=" +
dt.Rows[i]["User_ID"].ToString() + " AND Item_ID=" +
dt.Rows[i]["Item_ID"].ToString();

            }
            //Sonuçları Veritabanına Kaydediyoruz.
        }
#endregion
```

Correlation_Threshold_Algorithm

```
       #region Correlation_Threshold_Algorithm
        sql = "SELECT * FROM U1_TEST";
        dt = new DataTable();
        DataTable userdt = new DataTable();
        DtBase.DT(dt, sql);
        for (int i = 0; i < dt.Rows.Count; i++)
        {
```

```
            userdt = new DataTable();
  //Eşik Değerinin Üzerinde Kalan Kullanıcıları seçiyoruz.
  // userdt Datatable'ına aktarıyoruz.
            string sql1 = "";

            if (userdt.Rows.Count != 0 && userdt.Rows[0]["SONUC"].ToString()
!= null && userdt.Rows[0]["SONUC"].ToString() != "")
            {
                sql1 = "UPDATE U1_TEST SET CT=" +
                String.Format("{0:0.##########}",
   Convert.ToDouble(userdt.Rows[0]["SONUC"].ToString())).Replace(",", ".") +
" WHERE User_ID=" + dt.Rows[i]["User_ID"].ToString() + " AND Item_ID=" +
dt.Rows[i]["Item_ID"].ToString();

            }
            else
            {
                sql1 = "UPDATE U1_TEST SET CT=0 WHERE User_ID=" +
                dt.Rows[i]["User_ID"].ToString() + " AND Item_ID=" +
                dt.Rows[i]["Item_ID"].ToString();
            }
            //Sonuçları Veritabanına Kaydediyoruz.
        }

#endregion
```

Hkok

```
#region Hkok
        string sql = "SELECT DISTINCT USER_ID FROM U1_TEST";
        DataTable dt = new DataTable();
        Dtbase.DT(dt, sql);
        Double toplamp = 0;
        Double toplamv = 0;
        for (int i = 0; i < dt.Rows.Count; i++)
        {
            DataSet            sonuc            =            Dtbase.DS("HKOK",
dt.Rows[i]["USER_ID"].ToString());
            toplamp                                                          +=
Convert.ToDouble(sonuc.Tables[0].Rows[0]["SONUC"].ToString());
            toplamv                                                          +=
Convert.ToDouble(sonuc.Tables[1].Rows[0]["SONUC"].ToString());
        }

#endregion
```

Mho

```
#region Mho


       string sql = "SELECT DISTINCT USER_ID FROM U1_TEST";


        DataTable dt = new DataTable();


        DtBase.DT(dt,sql);
```

```
        Double toplamp = 0;
        Double toplamv = 0;

        for (int i = 0; i < dt.Rows.Count; i++)
        {

           DataSet                    sonuc=                    DtBase.DS("Mho",
dt.Rows[i]["USER_ID"].ToString());

toplamp+=Convert.ToDouble(sonuc.Tables[0].Rows[0]["SONUC"].ToString());
            toplamv                                                              +=
Convert.ToDouble(sonuc.Tables[1].Rows[0]["SONUC"].ToString());
        }

#endregion
```

STORED PROCEDURE	
PearsonSimilarity1, PearsonSimilarity2	Person Benzerliği hesaplanırken veri tabanındaki işlemleri yapar
VectorSimilarity1, VectorSimilarity2	Vector Benzerliği hesaplanırken veri tabanındaki işlemleri yapar
K_Means_Select	KNN Algoritması ile Veritabanından komşuları seçer
Correlation_Threshold_Select	Correlation Threshold Algoritması ile Veritabanından komşuları seçer
Hkok	Hkok ile veritabanındaki gerekli işlemleri gerçekleşirir
Mho	Mho ile veritabanındaki gerekli işlemleri gerçekleşirir

PearsonSimilarity1

```
BEGIN
SET NOCOUNT ON;
    SELECT   (SUM(CAST(Rate AS REAL))/
              COUNT(CAST(Item_ID AS REAL))) AS ortalama,
              User_ID FROM U1_BASE GROUP BY User_ID
              ORDER BY User_ID ASC;

END
```

PearsonSimilarity2

```
BEGIN
   SELECT  dbo.U1_BASE.User_ID,
              U1_BASE_1.Item_ID AS Expr1, dbo.U1_BASE.Rate,
              U1_BASE_1.Rate AS Rate1, dbo.U1_BASE.Item_ID,
              U1_BASE_1.User_ID AS User_ID1
   FROM         dbo.U1_BASE INNER JOIN
        dbo.U1_BASE AS U1_BASE_1 ON
        dbo.U1_BASE.Item_ID =    U1_BASE_1.Item_ID
        AND dbo.U1_BASE.User_ID <> U1_BASE_1.User_ID
   WHERE U1_BASE.User_ID=@user_id1
        AND U1_BASE.User_ID < U1_BASE_1.User_ID
   ORDER BY dbo.U1_BASE.User_ID, User_ID1, Expr1

END
```

VectorSimilarity1

```
BEGIN
SET NOCOUNT ON;
   SELECT  SQRT(Sum(SQUARE(Rate))) as     NORMALIZASYON,
           User_ID FROM U1_BASE
           GROUP BY User_ID ORDER BY User_ID ASC

END
```

VectorSimilarity2

```
BEGIN
     SELECT  TOP (100000)  dbo.U1_BASE.User_ID,
             U1_BASE_1.Item_ID AS Expr1,
             dbo.U1_BASE.Rate, U1_BASE_1.Rate AS Rate1,
             dbo.U1_BASE.Item_ID,
             U1_BASE_1.User_ID AS User_ID1
     FROM         dbo.U1_BASE INNER JOIN
             dbo.U1_BASE AS U1_BASE_1 ON
             dbo.U1_BASE.Item_ID = U1_BASE_1.Item_ID
             AND dbo.U1_BASE.User_ID <> U1_BASE_1.User_ID
     WHERE U1_BASE.User_ID=@user_id1 AND
           U1_BASE.User_ID<U1_BASE_1.User_ID
           ORDER BY dbo.U1_BASE.User_ID, User_ID1, Expr1

END
```

K_Means_Select

```
BEGIN
SET NOCOUNT ON;
 DECLARE @TABLO TABLE
   (
```

```
   Benzerlik FLOAT,
      RATE INT ,
      AVERAGE1 REAL,
      AVERAGE2 REAL

   )

  INSERT INTO @TABLO
  SELECT TOP 50  PB.Benzerlik,U1.Rate,US.AVERAGE AVERAGE1,
         US2.AVERAGE                AVERAGE2
  FROM
    Pearson_Benzerlik PB
    LEFT JOIN U1_BASE U1 ON U1.User_ID = PB.User2
    AND U1.Item_ID = @item_id
    LEFT JOIN USERS US ON US.USER_ID = PB.User1
    LEFT JOIN USERS US2 ON US2.USER_ID =PB.User2
  WHERE PB.User1 = @user_id1
    AND EXISTS ( SELECT 'X' FROM U1_BASE
    WHERE User_ID = PB.User2 AND Item_ID = @item_id)
  ORDER BY PB.Benzerlik DESC

  DECLARE @TABLO2 TABLE
  (
     SONUC FLOAT
  )
  INSERT INTO @TABLO2
  SELECT  ((SELECT TOP 1 AVERAGE1 FROM @TABLO)+
          (SELECT SUM((RATE-AVERAGE2)* Benzerlik)
          AS PAY FROM @TABLO)
          / (SELECT SUM (ABS(Benzerlik))
          AS PAYDA FROM @TABLO))    AS SONUC

          SELECT * FROM @TABLO2

  END
```

Correlation_Threshold_Select

```
      BEGIN
SET NOCOUNT ON;
 DECLARE @TABLO TABLE
 (
      Benzerlik FLOAT,
      RATE INT ,
      AVERAGE1 REAL,
      AVERAGE2 REAL
  )
  INSERT INTO @TABLO
  SELECT  PB.Benzerlik,U1.Rate,US.AVERAGE AVERAGE1,
          US2.AVERAGE AVERACE2 FROM
          Pearson_Benzerlik PB
          LEFT JOIN U1_BASE U1 ON U1.User_ID = PB.User2
          AND U1.Item_ID = @item_id
          LEFT JOIN USERS US ON US.USER_ID = PB.User1
          LEFT JOIN USERS US2 ON US2.USER_ID =PB.User2
    WHERE PB.User1 = @user_id1  AND PB.Benzerlik>=-0.5
          AND EXISTS ( SELECT 'X' FROM U1_BASE
```

```
          WHERE User_ID = PB.User2 AND Item_ID = @item_id)
   ORDER BY PB.Benzerlik DESC

   DECLARE @TABLO2 TABLE
   (
      SONUC FLOAT
   )
   INSERT INTO @TABLO2
   SELECT  ((SELECT TOP 1 AVERAGE1 FROM @TABLO)+
           (SELECT SUM((RATE-AVERAGE2)* Benzerlik)
           AS PAY FROM @TABLO)/      (SELECT SUM (ABS(Benzerlik))
           AS PAYDA FROM @TABLO))   AS SONUC
   SELECT * FROM @TABLO2

END
```

Hkok

```
      BEGIN
SET NOCOUNT ON;
    SELECT SQRT(SUM( SQUARE(Vektor_0_5-Rate)))/
    SQRT(COUNT(User_ID)) AS SONUC FROM U1_TEST
    WHERE User_ID=@user_id1

END
```

Mho

```
      BEGIN
SET NOCOUNT ON;
    SELECT (SUM( ABS(Vektor_0_5-Rate)))/
    COUNT(User_ID) AS SONUC FROM U1_TEST
    WHERE User_ID=@user_id1
END
```

KİŞİSEL YAYINLAR VE ESERLER

A. Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler

1. **Kaya F.**, Yıldız G., Kavak A., Evalution of collaborative filtering techniques for mobile and web application based recommendation system, *9th International Conference on Electronics, Computer and Computation*, Ankara, Turkey, 1-3 November 2012.

2. Yıldız G., **Kaya F.**, Kavak A., A Mobile application and web recommendation system for assisting indoor decoration, *International Symposium on INnovations in Intelligent SysTems and Applications,* Trabzon, Turkey, 2-4 July 2012.

B. Ulusal Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında Basılan Bildiri ve Posterler

1. **Kaya F.**, Yıldız G., Kavak A., Dekorasist: Bina içi dekorasyona yönelik öneri sistemi, *Uluslar arası Katılımlı 14. Ulusal Biyoistatistik Kongresi*, Kayseri, Türkiye, 4-7 Eylül 2012.

2. **Kaya F.**, Kumru Y., P., Fuzzy Topsis Yöntemi ile üniversitelerin web sitelerinin kullanilabilirlik degerlendirilmesi, *Yöneylem Araştırması ve Endüstri Mühendisliği 32. Ulusal Kongresi*, İstanbul, Türkiye, 20-22 Haziran 2012.

ÖZGEÇMİŞ

1988 yılında Kütahya'da doğdu. İlk, orta ve lise öğrenimini Kütahya'da tamamladı. 2006 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği'nden 2010 yılında, bölüm 1. olarak mezun oldu. 2011 yılında başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bilgisayar Bilimleri Anabilim Dalı'ndaki Yüksek Lisans eğitimine devam etmektedir. 2012 yılından itibaren Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak çalışmaktadır.