

**KOCAELİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**ARAŞTIRMA PROBLEMLERİ**

**YAPAY ZEKA TABANLI FİLM ÖNERİ UYGULAMASI**  
**GELİŞTİRME**

**CEMDENİZ ÇAY**

**Dr.Öğr.Üyesi FİDAN KAYA**  
**GÜLAĞIZ, Kocaeli Üniv.**

.....

.....

.....

**Tezin Savunulduğu Tarih: 02.06.2021**

## **ÖNSÖZ VE TEŞEKKÜR**

Bu tez çalışması, yapay zeka altyapısını kullanarak film öneri sistemine sahip bir mobil uygulama geliştirmek amacıyla gerçekleştirilmiştir.

Tez çalışmamda desteğini esirgemeyen, çalışmalarına yön veren, bana güvenen ve yüreklendiren danışmanım Arş.Gör. BURCU KIR SAVAŞ ‘a sonsuz teşekkürlerimi sunarım.

Tez çalışmamın tüm aşamalarında bilgi ve destekleriyle katkıda bulunan hocam Dr.Öğr.Üyesi FİDAN KAYA GÜLAĞIZ ‘a teşekkür ediyorum.

Hayatım boyunca bana güç veren en büyük destekçilerim, her aşamada sıkıntılarımı ve mutluluklarımı paylaşan sevgili aileme teşekkürlerimi sunarım.

Haziran – 2021

Cemdeniz ÇAY

Bu dokümandaki tüm bilgiler, etik ve akademik kurallar çerçevesinde elde edilip sunulmuştur. Ayrıca yine bu kurallar çerçevesinde kendime ait olmayan ve kendimin üretmediği ve başka kaynaklardan elde edilen bilgiler ve materyaller (text, resim, şekil, tablo vb.) gerekli şekilde referans edilmiş ve dokümanda belirtilmiştir.

Öğrenci No: 140202009

Adı Soyadı: Cemdeniz ÇAY

İmza:

A handwritten signature in black ink on a light gray background. The signature is stylized, starting with a large 'C' and 'D' followed by a series of loops and a long horizontal stroke at the bottom.

## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR .....	i
İÇİNDEKİLER.....	iii
ŞEKİLLER DİZİNİ .....	iv
TABLolar DİZİNİ.....	v
SİMGELER VE KISALTMALAR DİZİNİ.....	vi
ÖZET .....	vii
ABSTRACT .....	viii
GİRİŞ .....	1
1. LİTERATÜR .....	2
2. METODOLOJİNİN ARKA PLANI .....	7
2.1 İşbirlikçi Filtreleme .....	7
2.2. İçerik Tabanlı Filtreleme .....	9
3. MATERYAL .....	10
4. THE MOVIEANN .....	13
5. ARAYÜZ TASARIMI .....	20
6. SONUÇ VE İLERİ ÇALIŞMALAR.....	22
KAYNAKLAR.....	23
EKLER.....	26
EK KAYNAKLAR.....	28
ÖZGEÇMİŞ .....	28

## ŞEKİLLER DİZİNİ

Şekil 1.1	Öneri sistemlerinde kullanılan ana ve alt yöntemler .....	2
Şekil 1.2	MovieANN genel yapısı .....	6
Şekil 2.1	Kullanıcı ögesi matrisi .....	7
Şekil 4.1	Film veri kümesi için küme sonuçları .....	15
Şekil 4.2	Kullanıcı veri kümesi için küme sonuçları .....	16
Şekil 4.3	Kullanıcı - film eşlemeli kümeler ve bu kümeler arasında film türünün dağılımı. ....	17
Şekil 4.4	Verilen her küme için çok katmanlı sinir ağı yapıları .....	18

## **TABLolar DİZİNİ**

Tablo 1.1	Avantaj ve dezavantajlar.....	4
Tablo 3.1	Meslek özniteliği için kodlama şeması.....	11
Tablo 3.2	Yaş özniteliği için kodlama şeması .....	12
Tablo 3.3	Her bir tür için tür özniteliği .....	13
Tablo 4.1	Her bir küme için öneri modellerinin performans sonuçları.....	17
Tablo 4.2	Her bir küme için öneri modellerinin performans sonuçları.....	19
Tablo 4.2	K-Means kümeleri için öneri modellerinin performans sonuçları .....	19

## SİMGELER VE KISALTMALAR DİZİNİ

MBE	:	Ortalama Sapma Hata (Mean Biased Error)
MAE	:	Ortalama Mutlak Hata (Mean Absolute Error)
MSE	:	Ortalama Kare Hata (Mean Squared Error)
RMSE	:	Kök Ortalama Kare Hata (Root Mean Squared Error)

### Kısaltmalar

CF	:	Collaborative filtering
ALS	:	Alternating Least Square
SD	:	Standart Sapma
KNN	:	K-Nearest Neighbors

## İŞBİRLİKÇİ FİLTRELEME TABANLI FİLM TAVSİYE SİSTEMİ

### ÖZET

İnternette bulunan veri hacmi hergün hızla artmaktadır. Kullanıcılar bu geniş veri setlerinde sıklıkla kaybolmaktadır. Günümüzde öneri sistemleri, yüksek sayıda bulunan ham verilerden istenilen yararlı bilgiyi farklı filtreleme yöntemleriyle bulmak için kullanılır. Öneri sistemlerinde bir çok farklı filtreleme ve yaklaşımlar mevcuttur. Bunlardan bazıları işbirlikçi filtreleme(collaborative filtering), içerik tabanlı filtreleme(Content-based filtering) ve hibrit(Hybrid) yaklaşımlardır. Bu proje, hibrit yaklaşımı ön plana çıkarmak için, yapay zeka alt-yapılı film öneri sistemi olan MovieANN'den yararlanmış ve filtreleme yöntemi olarak ise işbirlikçi filtreleme yöntemi kullanılmıştır. İşbirlikçi filtreleme yönteminin varolan yöntemlerinden daha iyi sonuç vermesi adına kullanıcı kümeleri ve bunun yanısıra film kümeleride geliştirilmiştir. Kullanıcı ve film kümeleri yaratılırken derece bilgisi ve içerik tabanlı bilgilerde göz önünde bulundurulmuştur. Burada kümeleme yapılırken K-Means algoritması ve X-Means algoritmasından yararlanılmıştır. Uygulama öneri sunarken kullanıcı kümelerinin içeriğindeki film kümelerinden yararlanarak öneri yapmaktadır. Bu tasarım MovieLens adı verilen bir veri seti ile test edildi.MovieLens çok sayıda kullanıcı, film ve derecelendirmeden oluşan bir veri setidir.

**Anahtar Kelimeler:** yapay zeka, hibrit yaklaşım, öneri tabanlı sistemler, içerik tabanlı filtreleme yöntemi, işbirlikçi filtreleme yöntemi



# **COLLABORATIVE FILTERING BASED MOVIE RECOMMENDATION SYSTEM**

## **ABSTRACT**

The volume of data available on the internet is increasing rapidly every day. Users are often lost in these large data sets. Currently, suggestion systems are used to find useful information requested from numerous raw data using different filtering methods. There are many filtering and approaches in Recommendation Systems. Some of these are collaborative filtering, Content-based filtering, and hybrid approaches. In order to highlight the hybrid approach, this project used Movieann, an artificial intelligence sub-structured film recommendation system, and a collaborative filtering method was used as a filtering method. In order for the collaborative filtering method to work better than existing methods, user sets and film sets have been developed. Degree knowledge and content-based information were taken into account when creating user and movie sets. Here, the k-Means algorithm and The X-Means algorithm were used when clustering. The application makes suggestions by using movie sets in the content of user sets while offering suggestions. This design was tested with a dataset called MovieLens. MovieLens is a data set consisting of many users, movies, and ratings.

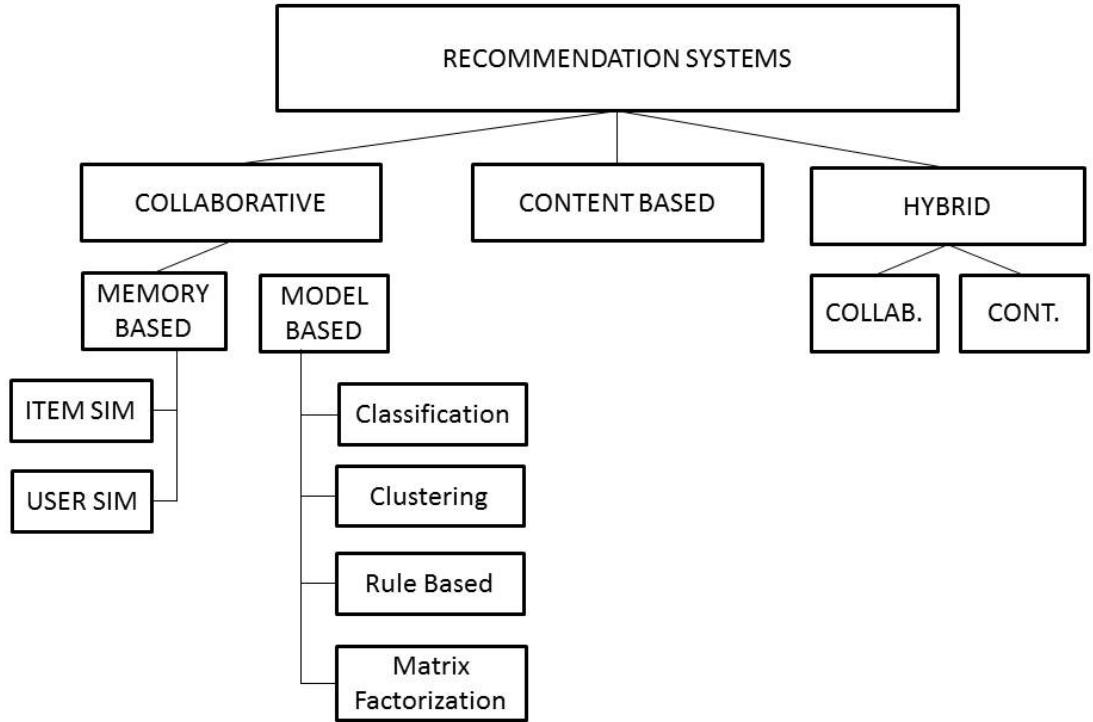
**Keywords:** artificial intelligence, hybrid approach, suggestion-based systems, content-based filtering method, collaborative filtering method

## GİRİŞ

Öneri sistemleri, web'deki büyük miktarda veriyi filtrelemek ve kullanıcılara en alakalı bilgileri sunmak için geliştirilmiştir. Bu şekilde, çevrimiçi kullanıcılar için zamandan ve emekten tasarruf ederler (Koohi, 2017). Tavsiye sistemlerinin arkasındaki ana fikir, bilgileri kullanıcının geçmiş eylemlerinden öğrenilen kullanıcının ilgi alanına göre sıralamaktır (Haruna ve ark., 2017). Hedefe ulaşmak için, tavsiye sistemleri üç ana filtreleme yaklaşımı üzerine kurulmuştur. Bunlar işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımlardır. İşbirlikçi filtreleme, benzer kullanıcıları ve/veya ürünleri bulmaya odaklanır. Öte yandan, içerik tabanlı filtreleme daha kişiselleştirilmiş bir yaklaşım önermektedir. Hibrit modeller, her iki yöntemin dezavantajının üstesinden gelmek için bu iki yaklaşımı birleştirir. Ağırlıklandırma, anahtarlama, karışık, özellik büyütme (Attarde, 2017; Lekakos, 2008; Burke, 2002) genellikle hibrit bir modelde farklı filtreleme mekanizmalarının kombinasyonu için kullanılır.

## 1. LİTERATÜR

Her ana öneri yaklaşımı alt gruplara ayrılabilir, her biri öneri algoritması, benzerlik hesaplaması ve boyut küçültme metodolojileri ile farklılık gösterir. Şekil 1.1, öneri sistemlerinde kullanılan ana ve alt yöntemleri temsil etmektedir.



Şekil 1.1. Tavsiye sistemlerinde kullanılan metodolojilerin hiyerarşik gösterimi

Öneri sistemleri oteller gibi birçok farklı alanda uygulanmaktadır, restoranlar, haberler, kitaplar, sosyal medya vb. (Sridevi ve ark., 2016). Bunlar arasında, film öneri sistemleri literatürde çok popülerdir ve birçok farklı örneğe sahiptir.

MovieLens (<https://grouplens.org>) iyi bilinen bir film tavsiyesidir. İşbirlikçi filtrelemeye dayanmaktadır. Tek bir benzerlik şeması kullanan işbirlikçi yaklaşımların çoğunun yanı sıra, MovieLens hem kullanıcı hem de öğe benzerliklerini birleştirir. Boyutsallık problemini ele almak için matris çarpanlarına ayırmayı kullanır. Her yeni kullanıcı, kayıt işlemi sırasında farklı türde bir dizi filmi değerlendirmelidir. Bu şekilde, kullanıcı tercihleri öğrenilir ve soğuk başlatma

sorunu önlenir. Bununla birlikte, güvenilirlik derecelendirme bilgileri sorgulanabilir, çünkü kayıt sürecinde sağlanır. Movielens'in bir diğer dezavantajı, en yüksek oranlara sahip filmlere önyargılı olmasıdır.

MovieREC (Kumar ve ark., 2015), gelişmiş bir K-Means algoritması kullanır. Filmler, k-Means algoritmasına girmeden önce öznitelik ağırlıklarına göre filtrelendirir. Beş özellik seçilir ve derecelendirme bilgilerine en yüksek ağırlık verilir. Kümeleme algoritması, film alanını dört kümeye böler. Ve bu kümelere göre Filmler önerilir. Bu yaklaşımın bazı dezavantajları vardır. Küme numarası (k) dörde sabitlenir. Optimize edilmiş bir k numarası daha iyi kümelere neden olabilir. Ağırlıklandırma şeması, filmleri önceden filtreler ve bu da K-araçlarının karmaşıklığını azaltır. Ancak, yalnızca beş özniteliklere dayanır. Ağırlıklandırma şemasında tüm haraclar dikkate alınır, sistem daha iyi performans gösterebilir. Daha da önemlisi, bu sistem küçük bir veri kümesi üzerinde test edilir ve geçerli bir istatistiksel değerlendirme yapılmaz.

Grupta (2015) hiyerarşik kümelemeye dayalı işbirlikçi bir filtreleme yaklaşımı önerdi. Bu çalışmada, kullanıcılar hiyerarşik kümelemeye göre kümelere ayrılır ve öneri küme oylaması ile yapılır. Bukalemun hiyerarşik kümeleme için K-araçları üzerinde seçilir. Derecelendirme önerileri ortalama Kare hatası ile değerlendirilir. Hiyerarşik kümeleme ve oylama şeması için optimizasyon yapılmasına rağmen, sistemin farklı film veri kümeleri üzerinde genelleştirilmesi bazı soruları gündeme getirmektedir. Hiyerarşik kümeleme algoritması çoğunlukla göreceli birbirine bağlılığa dayanır. Ara bağlantı için tek, ortalama ve tam bağlantı dikkate alınmalıdır. Belirli bir kümedeki öğelerin benzerliğini tanımlamak için küme içi benzerlik de hesaplanmalıdır. Bu benzerlik düşükse, kümelerin zayıf bir şekilde oluştuğu ve önerilere güvenilemeyeceği anlamına gelir. Algoritma, mevcut kümeleri bölmek veya birleştirmek için 25 bölüm noktası kullanır. Bu noktalar veriye bağlıdır ve daha küçük veya daha büyük olanlar için yüksek veya düşük olabilir kümeler. Sabit bir bölüm numarası yerine, veri hacmine dayalı dinamik bir optimizasyon yapılabilir.

İçerik tabanlı filtreleme daha kişiselleştirilmiş bir yaklaşım önermektedir. Benzer kullanıcılar yerine, öneri için belirli bir kullanıcının tercihleri kullanılır. Bu amaçla, bir kullanıcı profili vektörü oluşturulur. Kullanıcı profilinde vektör, kullanıcının belirli bir film üzerindeki faaliyetlerini kapsar(Cami ve ark ., 2017). Bayes ağ

algoritması, benzer filmlerin kümelerini oluşturmak için kullanılır. Belirli bir kümeye bir film atama olasılığı hesaplanır. Daha sonra, öneri listesi koşullu olasılık hesaplamaları ile oluşturulur. Olasılık hesaplamalarının zamanı ve karmaşıklığı bu yöntemin temel dezavantajlarıdır.

Yukarıdaki ve literatürdeki çalışmalar göz önüne alındığında, hem içerik hem de işbirlikçi yaklaşımların kendi avantaj ve dezavantajlarına sahip olduğunu görüyoruz. İşbirlikçi filtreleme, benzer kullanıcıların tercihlerini genelleştirir. Bu nedenle, içerik tabanlı filtrelemeye kıyasla yeni bir kullanıcı için öneri kolaydır. İçerik tabanlı filtreleme daha kişiselleştirilmiştir ve yeni bir kullanıcının tercihleri önceden bilinmemektedir. İşbirlikçi filtrelemedeki yeni öğeler de benzer bir dezavantaj sunar. Yeni bir öğe henüz derecelendirilmemiştir, bu nedenle belirli bir kullanıcı grubuyla mükemmel bir şekilde eşleşse bile tavsiye edilemez. Bu avantaj ve dezavantajlar Tablo 1.1'de listelenmiştir.

Approach	Advantage	Disadvantage
Content Based	Personalized Accurate	Always recommend similar things Cannot recommend to a new user Content Problems (Limited, Over-Specialized) Cold-Start
Collaborative	New users are handled easily No need to store content Info	No Content Information Not Personalized Difficult to recommend a new item Not Personalized Items >> users: not enough ratings (Sparsity, Scalability)

Tablo 1.1. Öneri sistemlerinde filtreleme yaklaşımlarının avantaj ve dezavantajları

Avantajları birleřtirmek ve bazı dezavantajların üstesinden gelmek için, iřbirlikçi ve içerik tabanlı filtrelemeyi birleřtiren hibrit yöntemler kullanılır. Virk ve diğ. (2015) böyle bir sistemin bir örneğidir. Bu çalışmada, kullanıcıların movielens algoritmasında olduđu gibi bazı filmleri önceden derecelendirmeleri gerekmektedir. Bu ön derecelendirme bilgileri, kullanıcı tercihlerini tanımlamak ve benzer kullanıcıları gruplamak için kullanılır. Öneri aşaması esas olarak iřbirlikçi filtrelemeye dayanmaktadır. İçerik tabanlı yaklaşım, bir film ve kullanıcı veritabanı oluşturmak için kullanılır, ancak öneriler için kullanılmaz. Bu nedenle, sistem bir melez yerine iřbirlikçi bir tavsiye olarak kabul edilebilir.

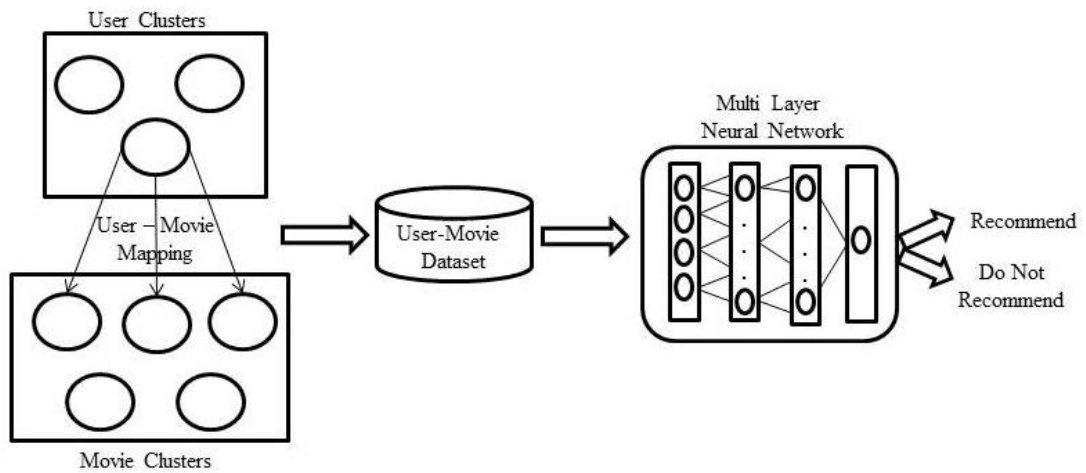
Tüysüzoğlu (2018), grafik tabanlı bir hibrit tavsiye sistemi önerdi. Bu çalışma, iřbirlikçi filtrelemeyi grafik teorisi ile birleřtirir. Filmler ve kullanıcılar dikey olarak temsil edilir ve kenarlar derecelendirmeleri temsil eder. İstatistiksel hesaplamalar en çok izlenen film, tür vb. listeler. Ne yazık ki, grafik teorisini kullanmanın öneri performansı üzerindeki etkisi belirtilmemiřtir.

Rombouts ve Verhoef (2002) iřbirlikçi ve içerik tabanlı filtreleme kullanan karma bir model tasarladı. İçerik tabanlı filtreleme, kullanıcıyı sınıflandıran naive Bayes modelini kullanır. Bu model her kullanıcı için inşa edilmiřtir. Önceki olasılık hesaplamaları Laplace yumuřatma dayanmaktadır. Daha sonra bu modeller dođrusal olarak birleřtirilir. İřbirlikçi aşamadaki korelasyonları tespit etmek için bir ağırlık řeması kullanılır. Bize göre, bu çalışmanın iki ana dezavantajı var. Birincisi, naive Bayes modelinin sayısı, kullanıcı sayısı ile dođru orantılıdır. Bu, daha fazla kullanıcının daha fazla hesaplama yaptıđı anlamına gelir. Bu nedenle yazarlar sistemin gerçek performansını bir bütün olarak hesaplayamazlar. İkinci dezavantaj, öğrenme verilerinde önyargıya neden olabilecek Laplace yumuřatma kullanımıdır.

Christakou ve Stafylopatis (2005) bir sinir ağı tabanlı hibrid recommender geliřtirdi. Sadece üç film niteliđi, tür, yıldız ve özet dikkate alınır. Her kullanıcı için, bu film öznitelikleri için üç sinir ağı oluşturulur. Sadece eğitim, test ve dođrulama veri kümelerini kullanılır ve çapraz dođrulama kullanılmaz ve sistemin performansı %82 ile yüksektir. Daha büyük veri kümeleri için sistem her bir kullanıcı için üç farklı sinir ağı oluşturulduđundan, model karmařıklıđından muzdarip olacaktır. Literatürdeki hibrit sistem örneklerine baktıđımızda, iki ana yaklaşım vardır.

Birincisi, temel bir filtreleme yöntemi, esas olarak makine öğrenimine dayanan model tabanlı bir yöntemle birleştirilir. Bu yaklaşım, gerçek bir hibrit sistemden ziyade model tabanlı filtreleme olarak düşünülebilir. İkinci yaklaşımda, içerik tabanlı ve işbirlikçi filtreleme yöntemleri birleştirilir. Araştırmalar incelendiğinde, öneri aşamasında makine öğrenimini kullanan sistemlerin temel eksikliğinin karmaşıklığı olduğunu gözlemledik.

Bu karmaşıklığın temel nedeni, ilgili hibrit sistemlerin her kullanıcı veya her ürün için belirli bir karar verme modeli geliştirmesidir. İlgili problemi çözmek için işbirlikçi ve içerik tabanlı filtreleme yöntemleri karma bir yaklaşımla birleştirildi (Burke, 2002) ve öneri aşamasında çok katmanlı sinir ağı modeli kullanıldı. Her iki filtreleme yöntemini de karma yaklaşımla birleştirerek, içerik tabanlı filtrelemenin soğuk başlatma sorunu ve işbirlikçi filtrelemenin veri seyrekliği sorunu çözülür. Yeni bir kullanıcı sisteme katıldığında, bireysel tercihleri bilinmemektedir (soğuk başlangıç), ancak işbirlikçi filtreleme yoluyla benzer bir kullanıcı grubuna atanır. Öneriler yeni kullanıcıya yapılır, kullanıcı grubunun tercihlerine göre. Öte yandan, nadiren derecelendirilen öğeler veya henüz derecelendirilmemiş öğeler, içerik tabanlı filtrelemeye sahip kullanıcılara önerilebilir. Hibrit model MovieANN olarak adlandırılır ve Şekil 1.2'de verilmiştir.



Şekil 1.2. Bu MovieANN genel yapısı

Bu sistemde, kullanıcılar ve filmler seçilen özniteliklere göre kümelenir. Öneri adımı, kullanıcı kümesini film kümelerine eşlemekle başlar. Bu haritalamadan elde edilen veri kümesi, Yapay Sinir Ağı modelinin eğitiminde kullanılır. İlgili model, yeni bir filmin kullanıcı tarafından beğenilip beğenilmeyeceğini sınıflandırır. Kağıdın akışı şu şekilde düzenlenmiştir: Bölüm-2, filtreleme yaklaşımları için kısa bir arka plan verir, Bölüm-3 materyalleri sunar, filmin detayları ve sonuçları bölüm-4'te verilir ve kağıt Bölüm-5'te sona erer.

## 2. METODOLOJİ ARKA PLANI

Bu bölümde, işbirlikçi ve içerik tabanlı filtrelemeyi kapsayan kısa bir metodoloji arka planı verilmiştir.

### 2.1 İşbirlikçi Filtreleme

İşbirlikçi filtreleme, öneri sistemlerinde yaygın olarak kullanılmaktadır (Chen ve ark., 2018). İşbirlikçi filtreleme metodolojileri, kullanıcı ögesi matrisini temel alır. Bu matris, her kullanıcı için her ögenin derecelendirme bilgilerini saklar. Bu matris şekil 2.1'de verilmiştir.

	$P_1$	$P_2$	$P \dots$	$P_N$
$U_1$	$R_{1-1}$	$R_{1-2}$	$R_{1- \dots}$	$R_{1-N}$
$U_2$	$R_{2-1}$	$R_{2-2}$	$R_{2- \dots}$	$R_{2-N}$
$U_{\dots}$	$R_{\dots-1}$	$R_{\dots-2}$	$R_{\dots- \dots}$	$R_{\dots-N}$
$U_M$	$R_{M-1}$	$R_{M-2}$	$R_{M- \dots}$	$R_{M-N}$

Figure 2.1. The user – item matrix, P indicates products, U indicates users and R indicates the rating given by a user for a specific product

Bu matris, kullanıcının geçmiş eylemleri tarafından oluşturulur, bu nedenle bazı çalışmalarda bu işbirlikçi yaklaşım, bellek tabanlı işbirlikçi filtreleme olarak adlandırılır (Chen ve ark., 2018]. Öğe-kullanıcı matrisine dayanarak, kullanıcıların benzerliğine veya öğelerin benzerliğine göre öneriler verilir. Kullanıcı tabanlı yaklaşım, belirli bir etki alanında benzer tercihlere sahip kullanıcıları gruplandırmaya



çalışır. Bir ürün gruptaki kullanıcıların çoğu arasında yüksek sırada yer alıyorsa, ürünü henüz sıralamamış olan kullanıcılara önerilir. Öge tabanlı yaklaşım, öge derecelendirmelerine odaklanır. Belirli bir kullanıcı için, yeni bir ögenin sıralaması, daha önce sıralanan öğelere benzerliğine göre hesaplanır. Her iki durumda da, benzerlikler hesaplanmalı, mahalle oluşturulmalı ve derecelendirmeler atanmalıdır. Kosinüs benzerliği (Vıt, 2018), Öklid mesafesi (Draisma ve ark., 2014), Pearson korelasyonu (Pearson, 1895) ve Jaccard benzerliği (Michael, 1971), ürünlerin veya kullanıcıların benzerliğini hesaplamak için kullanılır. Bunlar arasında kosinüs benzerliği ve Pearson korelasyonu yaygın olarak kullanılmaktadır (Bobadilla ve ark., 2013).  $P = \{P_1, P_2, \dots, p_n\}$  olan iki kullanıcı ( $u_1, u_2$ ) ve  $n$  ürünü olduğunu varsayalım. Ürünlerin derecelendirme vektörü  $r = \{r_1, r_2, \dots, r_n\}$  olarak verilir. Daha sonra  $U_1$  için derecelendirme vektörü  $ru_1 = \{ru_{1p1}, ru_{1p2}, \dots, ru_{1pn}\}$  olarak tanımlanır ve  $u_2$  için derecelendirme vektörü  $ru_2 = \{ru_{2p1}, ru_{2p2}, \dots, ru_{2pn}\}$  olarak tanımlanır. Bu durumda, derecelendirme vektörleri  $ru_1$  ve  $ru_2$  arasındaki benzerlik, aşağıdaki gibi kosinüs benzerliği ile hesaplanabilir:

$$\cos(r_{u_1}, r_{u_2}) = \frac{\sum_{i=1}^n r_{u_1pi} \cdot r_{u_2pi}}{\sqrt{\sum_{i=1}^n (r_{u_1pi})^2} \sqrt{\sum_{i=1}^n (r_{u_2pi})^2}} \quad (2.1)$$

Pearson korelasyonu, belirli ürün derecelendirmesi ile ürünlerin ortalama derecelendirmesi arasındaki farkın nokta ürünlerinin toplamıdır (kök ürün toplamına bölünür. Daha resmi olarak:

$$\text{sim}(r_{u_1}, r_{u_2}) = \frac{\sum_{i=1}^n (r_{u_1pi} - \bar{r}_{u_1}) \cdot (r_{u_2pi} - \bar{r}_{u_2})}{\sqrt{\sum_{i=1}^n (r_{u_1pi} - \bar{r}_{u_1})^2} \cdot \sqrt{\sum_{i=1}^n (r_{u_2pi} - \bar{r}_{u_2})^2}} \quad (2.2)$$

Mahalle üretimi genellikle  $K - \text{En Yakın Komşu (KNN)}$  algoritmasına dayanmaktadır (Portekiz ve ark., 2018). Bu algoritma, etkin kullanıcı ( $u_A$ ) ve diğer kullanıcılar ( $u$ ) arasındaki mesafeyi hesaplar. Daha sonra önceden tanımlanmış  $k$  sayısına göre, en yakın  $k$  kullanıcıları aktif kullanıcının mahallesini oluşturur. Mesafe hesaplamaları için Öklid mesafesi, Manhattan mesafesi ve Minkowski mesafesi gibi farklı metrikler kullanılır.

$$Euclidian (u_A, u) = \sqrt{\sum_{i=1}^n (u_A - u_i)^2} \quad (2.3)$$

$$Manhattan (u_A, u) = \sum_{i=1}^n |u_A - u_i| \quad (2.4)$$

$$Minkowski (u_A, u) = (\sum_{i=1}^n |u_A - u_i|^x)^{1/x} \quad (2.5)$$

Kullanıcı benzerlikleri ve komşuluk belirlendikten sonra, yeni bir öge için aktif kullanıcının ( $u_A$ ) derecelendirme tahmini,  $i$ , formüle göre hesaplanır (Chen ve ark., 2018; Resnick et al., 1994) aşağıda:

$$r_{u_{A_i}} = \bar{r}_{u_A} + \frac{\sum_{u \in N} sim(u_A, u) (r_{ui} - \bar{r}_u)}{\sum_{u \in N} |sim(u_A, u)|} \quad (2.6)$$

Benzerlik, mahalle ve derecelendirme tahmini hesaplamaları yerine, bazı işbirlikçi filtreleme yaklaşımları bir karar modeli oluşturmak için makine öğrenme yöntemlerini kullanır (Mahadevan, 2016). Bu şemada, model belirli bir ürün için benzer kullanıcıların tercihlerini öğrenir. Öğrenme şeması sınıflandırma, kümeleme ve matris çarpanlarına dayanabilir (Xiao, 2019).

## 2.2 İçerik Tabanlı Filtreleme

Bu filtreleme yöntemi, metin alma, haber, kitap veya diğer metinsel veri önerileri için yaygın olarak kullanılır (Pazzani, 2007). Metodoloji, ögenin içeriğini tanımlar ve bu içerik kullanıcının profiliyle eşleştirilir. Kullanıcı aynı veya benzer içerikle ilgileniyorsa, öge kullanıcıya önerilir. Bu şema iki önemli adım gerektirir. Belirli bir ögenin içeriğini tanımlamak ve bir kullanıcı profili oluşturmak için. Ögeler vektörler olarak temsil edilir,  $\vec{I} = (a_1, a_2, \dots, a_n)$   $n$  özniteliklerinden oluşur. Amaç, giriş vektörünü sınıflandıran bir  $f(\vec{I})$  işlevi bulmaktır. Gerçek bir dünya senaryosunda niteliklerin önemi eşit değildir. Böylece, ağırlıklar önemini belirtmek için özniteliklere atanır. Ağırlıklar TF-IDF kullanılarak belirlenir (Salton, 1975). Birçok öge  $a_1$  özniteliğini içeriyorsa, bu öznitelik önemli olabilir ve önemi TF (terim frekansı) ile hesaplanır. Ancak, ögeleri birbirinden ayırırken yüksek frekans nitelikleri önemli olmayabilir. Düşük frekans öznitelikleri ögeleri ayırt etmek için daha önemli olabilir. Bu IDF (ters terim frekansı) ile temsil edilir.  $D$  belge kümesinin

n belgeden oluştuğunu varsayalım  $D = \{d_1, d_2, \dots, d_n\}$  ve terimler  $t = \{t_1, t_2, \dots, t_m\}$  teriminde verilir, daha sonra  $d_1$ 'deki  $t_1$  terimi için TF aşağıdaki gibi hesaplanır:

$$TF(t_1, d_1) = \sum_{i=1}^m \frac{t_i \in d_1}{t_i \in d_1} \quad (2.7)$$

$$IDF(t_1, D) = \log \sum_{i=1}^n \frac{d_i}{t_1 \in d_i} \quad (2.8)$$

Daha sonra  $t_1$  teriminin ağırlığı şu şekilde hesaplanır:

$$W_{t_1} = TF_{t_1} * IDF_{t_1} \quad (2.9)$$

Bu ağırlıklar içerik vektörlerini oluşturmak için kullanılır. Bu vektörler daha sonra kullanıcı tercihlerini öğrenmek için kullanılır. Benzerlik hesaplamaları (Pazzani, 1999) ve / veya modele dayalı yaklaşımlar (Campos ve ark., 2010) bu öğrenme sürecinde kullanılabilir.

### 3. MATERYALLER

Movieann'in performansını test etmek için MovieLens 1m veri seti (Maxwell, 2015) kullanılır. Bu veri seti, dört bin film için altı bin kullanıcının bir milyon derecelendirmesini içerir. Bu veriler üç ayrı olarak düzenlenmiştir".dat "olan dosyalar " kullanıcılar.dat", " filmler.dat " ve " derecelendirme.dat".Kullanıcı.dat " dosyası kullanıcı bilgilerini saklar. Nitelikler kimlik, yaş, cinsiyet, meslek ve zaman damgasıdır. Cinsiyet özniteliği ikili, F dişi ve m erkeği gösterir. Nitelikler yaş ve meslek ayrıktır. Yaş özniteliğinin yedi grubu vardır ve meslek özniteliğinin yirmi grubu vardır. Bu grupların kodlanması tablo 3.1 ve Tablo 3.2'de verilmiştir.

Group Code	Data
1	academic/educator
2	artist
3	clerical/admin
4	college/grad student
5	customer service
6	doctor/health care
7	executive/managerial
8	farmer
9	homemaker
10	K-12 student
11	lawyer
12	programmer
13	retired
14	sales/marketing
15	scientist
16	self-employed
17	technician/engineer
18	tradesman/craftsman
19	unemployed
20	writer

Tablo 3.1. Meslek özniteliği için kodlama şeması

Group Code	Data Range
1	<18
18	18-24
25	25-34
35	35-44
45	45-49
50	50-55
56	>=56

Tablo 3.2. Yaş özniteliği için kodlama şeması

“Filmler.dat ” dosyası filmler hakkında bilgi depolar. Nitelikler film kimliği, film adı ve türlerdir. Türler on sekiz alt gruba ayrılır. Bir filmin birden fazla türü olabilir. Film türü kodlaması ve belirli bir türün film sayısı tablo 3.3'te verilmiştir.

Genre	Coding	Number of Movies
Action	1	503
Adventure	2	283
Animation	3	105
Children's	4	252
Comedy	5	1200
Crime	6	211
Documentary	7	127
Drama	8	1602
Fantasy	9	68
Film-Noir	10	44
Horror	11	343
Musical	12	114
Mystery	13	106
Romance	14	471
Sci-Fi	15	276
Thriller	16	492
War	17	143
Western	18	68

Tablo 3.3. Her bir tür için tür özniteliği ve film sayısı için kodlama şeması

“Verimler.dat ” dosyası derecelendirme bilgilerini saklar. Öznitelikler kullanıcı kimliği, film kimliği, derecelendirme puanı ve zaman damgasıdır. Derecelendirme puanı beş yıldız ölçeği olarak verilir ve beşi en yüksek puanı gösterir.

#### 4. THE MOVIEANN

MovieANN, işbirlikçi ve içerik tabanlı filtrelemeyi birleştiren ve öneri aşamasında çok katmanlı bir yapay sinir ağı kullanan karma bir film öneri sistemidir. MovieANN, işbirlikçi ve içerik tabanlı filtrelemeyi birleştiren ve öneri aşamasında çok katmanlı bir yapay sinir ağı kullanan karma bir film öneri sistemidir. İçerik tabanlı yaklaşımda

soğuk başlangıç problemini ele almak için, kullanıcılar benzerliklerine göre gruplandırılır. Bu şekilde, belirli bir kullanıcıya yeni bir kullanıcı atanır. benzer tercihlere sahip kullanıcı grubu. İşbirlikçi filtrelemede veri seyrekliği sorununu ele almak için içerik tabanlı bir yaklaşım kullanılır. Filmler içerik benzerliklerine göre gruplandırılır. Bu nedenle, derecelendirme bilgisi az olan veya hiç olmayan filmler de önerilebilir. Benzer kullanıcıları ve benzer filmleri gruplamak için kümeleme kullanılır.

Kümeleme için KMeans (Zahra ve diğerleri, 2015) ve X-Means (Pelleg, 2000) algoritmaları tercih edilir. K-Means algoritmasındaki  $k$  için en uygun sayı,  $k \leq n$  varsayımı ile belirlenir, burada  $n$ , veri kümesindeki toplam örnek sayısıdır. X - means, Bayes bilgi kriterlerine göre optimum  $k$ 'yi hesaplar (Schwarz, 1978). Küme performansları ortalama küme içi mesafe (AID) ve Davies-Bouldin indeksi (DBI) ile ölçülür (Davies, 1979). Küme içi ortalama mesafe, belirli bir sınıftaki bir öğenin benzerliğini tanımlar. DBI, kümeler arası farklılık ve kümeler arası benzerlik oranını verir. Rapidminer 9.1 (Mierswa, 2019) aracı kümeleme algoritmalarını uygulamak için kullanılır. Film kümelemesi için hem K-Means hem de X-Means, film veri kümesini iki kümeye böler.

AID ve DBI da her iki yöntem için aynıdır ve sırasıyla 0.247 ve 0.495 olarak hesaplanır. Sadece kümelerdeki film sayısı biraz farklıdır. Şekil4. 1, film veri kümesi için K-Means ve X-Means tarafından oluşturulan kümeleri temsil eder.

## k-Means - Summary

**Cluster 0**

1,893

Average Distance: 0.242

**Cluster 1**

1,990

Average Distance: 0.251

## x-Means - Summary

**Cluster 0**

1,991

Average Distance: 0.251

**Cluster 1**

1,892

Average Distance: 0.242

Şekil 4.1. Film veri kümesi için küme sonuçları

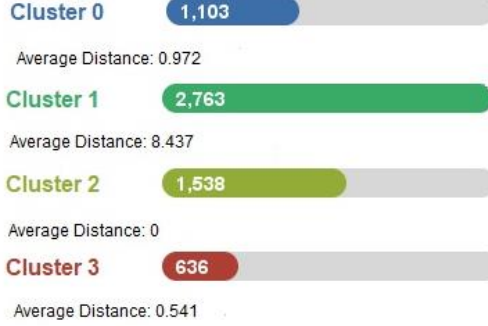
Kullanıcıların benzerliğini tanımlamak için bir kullanıcı kümesi oluşturulur. Bu küme özniteliklerini temel alır "kullanıcı.dat " DataSet, timestamp özniteliği dışında. Zaman damgası benzer kullanıcıları bulmak için kullanılabilir. Örneğin, mesleği olan kullanıcılar geceleri film izleyebilir, mesleği olmayan kullanıcılar gündüz film izleyebilir. Ne yazık ki, bu tür bilgileri veri dosyasındaki timestamp özniteliğinden çıkaramayız.

Kullanıcıların kümelenmesi için K-Means ve X-Means farklı sonuçlar verir. K-means algoritması kullanıcıları iki kümeye ayırırken, X-Means onları dört kümeye böler. Sonuçlar Şekil 4.2'de karşılaştırılmıştır.



### x-Means - Summary

Number of Clusters: 4  
Distance Measure: Euclidean Distance  
Average Cluster Distance: 4.094  
Davies-Bouldin Index: 1.126



### k-Means - Summary

Number of Clusters: 2  
Distance Measure: Squared Euclidean Distance  
Average Cluster Distance: 5.894  
Davies-Bouldin Index: 0.871



Şekil 4.2. Kullanıcı veri kümesi için küme sonuçları

Şekil-4.2, K-Means algoritmasının DBI açısından daha iyi sonuçlar verdiğini göstermektedir. Ancak, kümeler heterojendir ve verilerin çoğunluğu ilk kümede (küme 0) gruplandırılır. X-Means dört küme oluşturdu ve K-Means kümelerine kıyasla daha homojendir. Bu durum yardım tarafından doğrulanır. X-Means algoritması yardım açısından daha iyi sonuçlar verir. K-araçlarının kümelerinin uzak olduğunu söyleyebiliriz, ancak belirli bir kümedeki veriler de uzaktır. Aksine, X-kümelerinin daha yakın olduğu anlamına gelir. Ayrıca, her belirli kümede, veri noktaları daha benzerdir. Öneri aşamasında, bir kullanıcıya bir film önerilir. Bu adımda, kullanıcının ait olduğu belirli küme, küme-X seçilir. Küme-x'deki kullanıcılar tarafından derecelendirilen filmler alınır ve kullanıcılarla eşleştirilir. Kullanıcı-film eşleme kümeleri ve bu kümeler arasındaki film türü dağılımı Şekil 4.3'te verilmiştir.



Şekil 4.3. Kullanıcı - film eşlemeli kümeler ve bu kümeler arasında film türünün dağılımı

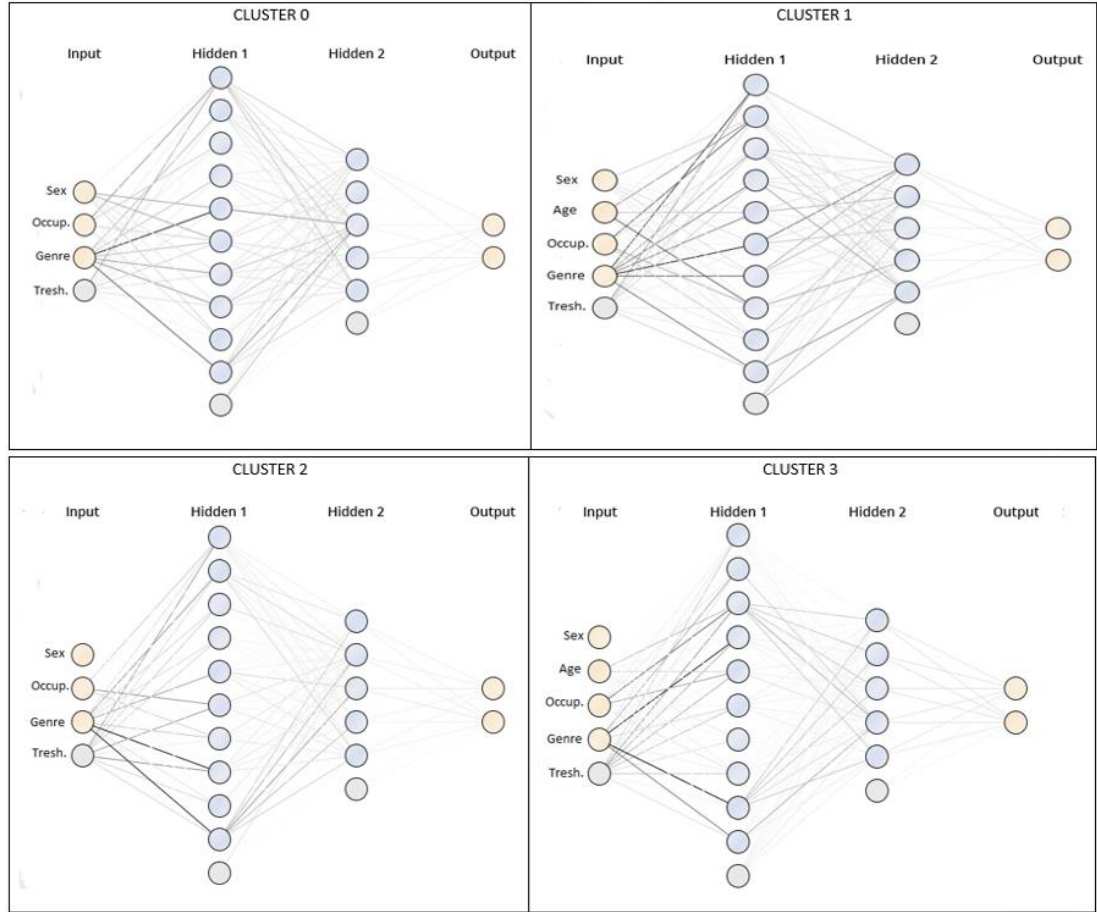
Şekil 4.3'e göre, küme 0'daki en popüler film türleri aksiyon ve maceradır. Küme 1'deki kullanıcılar çoğunlukla komedi, aksiyon ve dramayı değerlendirdi. Küme 2'de, aksiyon filmleri çoğunlukla tercih edilir ve bunu komedi ve drama izler.

Film kullanıcı eşlemesinden sonra, eşlenen veri kümesinin üzerine çok katmanlı bir yapay sinir ağı modeli oluşturulur. Sinir ağı mimarisinin iki gizli katmanı vardır. İlk gizli katman on düğümden oluşur ve ikinci gizli katman beş düğümden oluşur. Her katmanda sigmoid aktivasyon fonksiyonu kullanılır. Model Rapidminer 9.3'te uygulanmaktadır. Öğrenme hızı, eğitim döngüleri ve momentum parametreleri optimize edilmiştir. Her değer aralığı dört adım boyutuna göre optimize edilmiştir ve bu test etmek için 125 kombinasyon verir. Değer aralıkları ve optimum değerler Tablo 4.1'de verilmiştir.

	Min. Value	Max. Value	Optimum Value Cluster-0	Opt. Val. Cluster-1	Opt. Val. Cluster-2	Opt. Val. Cluster-3
Training Cycle	50	200	200	200	200	200
Learning Rate	0	1	0.25	0.25	0.25	0.25
Momentum	0	1	0.333	0.25	0.25	0.25

Tablo 4.1. Her bir küme için öneri modellerinin performans sonuçları

Her bir küme için Optimum eğitim döngüleri, momentum ve öğrenme oranları kullanılır. Her küme için çok katmanlı modeller Şekil 4.4'te verilmiştir.



Şekil 4.4. Verilen her küme için çok katmanlı sinir ağı yapıları

Şekil 4.4'te gösterildiği gibi, age özneliği 0 ve 1 kümeleri için bir girdi olarak verilmez. Bunun nedeni, tek bir yaş aralığının her kümeyi temsil etmesidir. Öte yandan, 1.ve 3. kümelerde, age özneliği girdi olarak verilir, çünkü bu kümelerde farklı yaş aralıkları çıkar. Her mimarinin detayları ek materyalde verilmiştir. Her öneri modeli için, veri kümesi %70 eğitim %30 test olarak bölünmüştür. Bu modellerin performansları doğruluk, hassasiyet ve hatırlama açısından ölçülür. Performans sonuçları Tablo 4.2'de verilmiştir

	Model-1 (Cluster 0)	Model-2 (Cluster 1)	Model-3 (Cluster 2)	Model-4 (Cluster 3)
Accuracy	80.86	85.80	82.09	89.33
Precision	80.86	85.87	82.09	89.33
Recall	100	99.9	100	100

Tablo 4.2. Her bir küme için öneri modellerinin performans sonuçları

Tablo 4.2, tüm kümeler için önerilen filmlerin doğruluk, hassasiyet ve hatırlama açısından uygun olduğunu göstermektedir. Filmin genel doğruluğu %84.52'dir. Hassasiyet ve hatırlama açısından, genel performans sırasıyla %84.54 ve %99.98'dir. K-Means ve X-Means kümelerini karşılaştırmak için aynı ANN yaklaşımı kullanılır. KMeans, Şekil 4.2'de verilen iki kümeye sahiptir. Böylece iki ayrı ANN modeli oluşur. Bu modellerin performans göstergeleri Tablo 4.3'te verilmiştir.

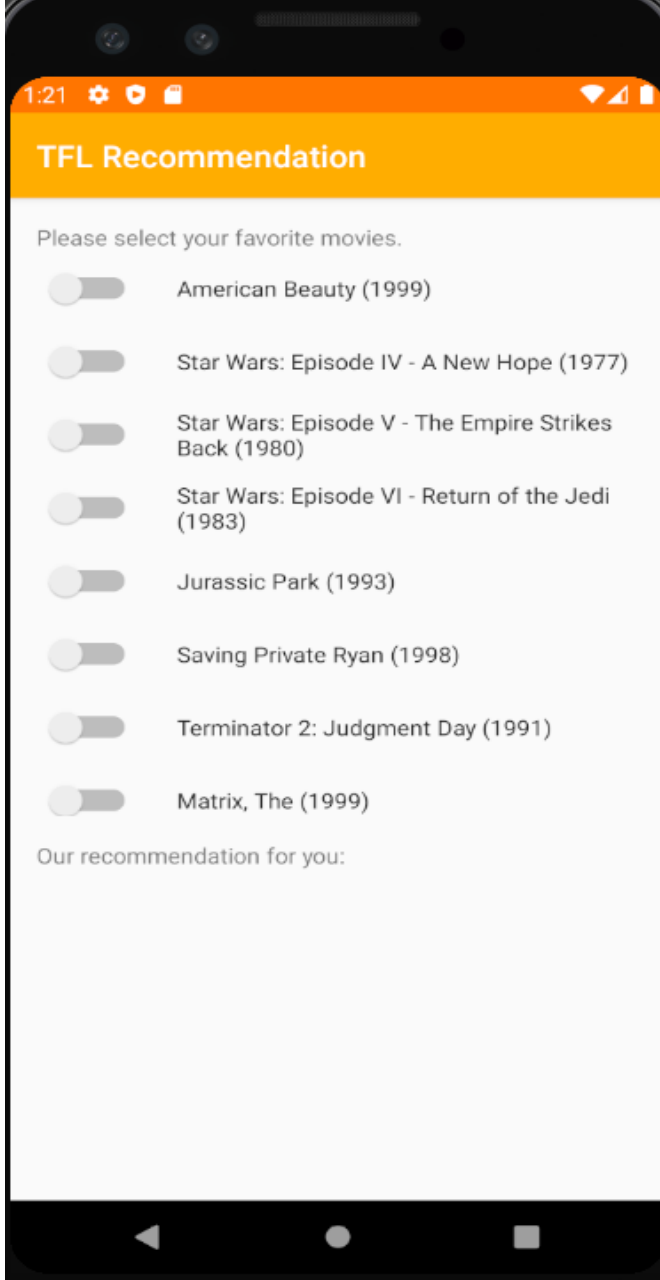
	Model-1 (Cluster 0)	Model-2 (Cluster 1)
Accuracy	83.2	87.92
Precision	100	80
Recall	~0	2.68

Tablo 4.3. K-Means kümeleri için öneri modellerinin performans sonuçları

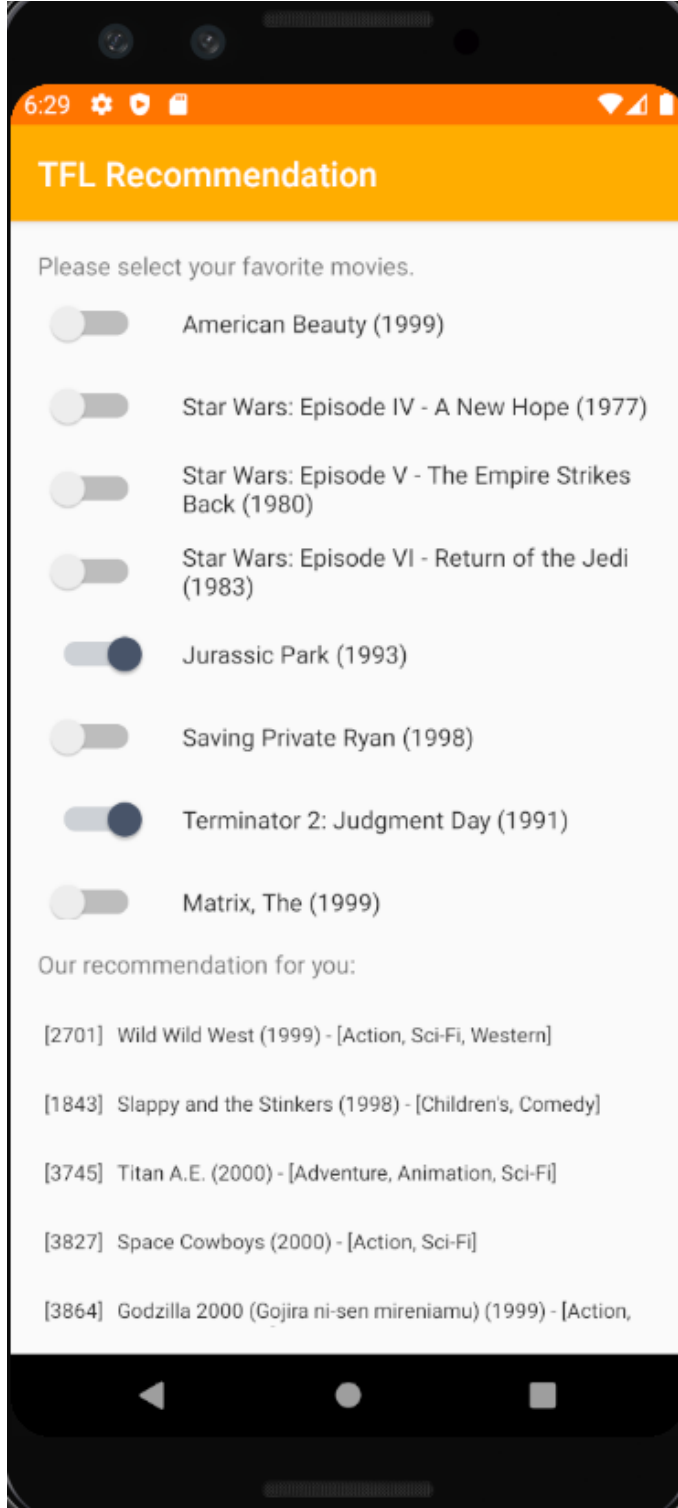
Tablo 4.3'te belirtildiği gibi, K-Means kümelerinin oluşturduğu karar modelleri düşük geri çağırma oranlarından muzdariptir. Düşük AUC sonuçları, 0.588 ve 0.621 de bunu pekiştiriyor. X-Means cluster tarafından oluşturulan karar modelleri daha sağlamdır ve geri çağırma da dahil olmak üzere genel performans göstergeleri daha umut verici ve daha yüksektir.

## 5. ARAYÜZ TASARIMI

İşbirlikçi filtreleme yöntemini kullanan ve gelişim aşamasında bulunan mobil uygulama arayüzünden görseller aşağıdaki gibidir;



Kullanıcı farklı film gruplarından oluşan bir listeden daha önce izlediği ve beğendiği filmlerden istediği kadar seçebilecek.



Her farklı seçimde algoritma tekrar çalıştırılarak kullanıcıya seçimlerinden yola çıkarak en çok beğenme ihtimali bulunan film önerileri sıralanacak.

## 6. SONUÇ VE İLERİ ÇALIŞMALAR

Öneri sistemleri, büyük miktarda veriyi filtrelemek ve ilgili bilgileri kullanıcı tercihlerine göre sunmak için kullanılır. Bu sistemler birçok farklı alana uygulanabilir ve bunlar arasında film önerileri de vardır. Film öneri sistemleri üç ana yaklaşım kullanır. Bunlar işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit sistemlerdir. İşbirlikçi filtreleme esas olarak veri seyrekliğinden muzdariptir ve içerik tabanlı filtreleme soğuk başlangıç probleminden muzdariptir. Dezavantajların üstesinden gelmek için hibrit sistemler önerilmektedir. Bu sistemler, iki ana filtreleme yaklaşımını ağırlıklandırma veya büyütme gibi farklı metodolojilerle birleştirir. Bazı hibrit sistemler, bir makine öğrenme modelini bir filtreleme yöntemiyle birleştirir. Bu tür sistemlerde asıl sorun karar modellerinin sayısıdır. Genellikle, her bir kullanıcı veya öge için bir karar modeli oluşturulur. Bu, hesaplamanın yanı sıra karmaşıklığı da artırır. İçerik tabanlı filtrelemenin soğuk başlangıç probleminin ve işbirlikçi filtrelemenin veri seyrekliğinin üstesinden gelmek için, çok katmanlı yapay sinir ağına dayanan bir hibrit model olan MovieANN oluşturduk. Bu model, film ve kullanıcı kümeleri oluşturur. Film kümeleri içerik tabanlı bir şekilde oluşturulur ve kullanıcı kümeleri işbirlikçi bir şekilde oluşturulur. Kümelemede K-means ve X-Means algoritmaları kullanılır ve son kümeler Davies-Bouldin indeksine ve ortalama mesafe ölçümlerine göre seçilir. Öneri aşamasında, kullanıcı ve film kümeleri eşleştirilir. Belirli bir kullanıcı için, bu kullanıcının kümesi seçilir ve öneri modeli bu belirli kümeye dayanır. Öneri modeli çok katmanlı sinir ağına dayanmaktadır. Ağ iki gizli katmandan oluşur.

MovieANN, Movielens'in 1m veri kümesi üzerinde test edilmiştir ve genel doğruluk %84.52'dir. 84.54% 'in genel hassasiyeti ve 99.98%' in genel hatırlamasıyla, Movieann'ın performans göstergeleri çok yüksektir.

Movieann'ın tavsiye performansı yüksek olsa da, performansı daha da artırmak için iyileştirmeler yapacağız. Hem kullanıcı hem de film öznitelikleri, kullanılan veri kümesine bağlı olarak çok sınırlıdır. Film sitelerinden bilgi almak için bir ayrıştırıcı yazılacaktır. Bu bilgiler, içerik tabanlı yaklaşıma uygun olarak TF \* IDF yöntemi ile işlenecektir. Bu, filmler için daha fazla özniteliğin işlenmesini sağlar.

## KAYNAKLAR

- 1) Attarde D.V., Singh M., 2017. Survey on Recommendation System Using Data Mining and Clustering Techniques. *International Journal for Research in Engineering Application and Management (IJREAM)*, 3(9). ISSN : 2454-9150.
- 2) Bobadilla J., Ortega F., Hernando A., Gutiérrez A., 2013. Recommender Systems Survey. *Knowledge-Based Systems*, 46: 109–132.
- 3) Burke R., 2002. Hybrid recommender systems: Survey and experiments. *User Model User Adapt Interact*, 12: 331–370.
- 4) Cami B. R., Hassanpour H., Mashayekhi H., 2017. A Content-Based Movie Recommender System Based on Temporal User Preferences. *Third Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*. DOI: 10.1109/ICSPIS.2017.8311601.
- 5) Campos L.M, Fernández-Luna J.M., Huete J.F., Rueda-Morales M.A., 2010. Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. *International Journal of Approximate Reasoning*, 51: 785–799.
- 6) Chen R., Hua Q., Chang Y.S., Wang B., Zhang L., Kong X.A, 2018. Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based On Social Networks. *IEE Access*. DOI: 10.1109/ACCESS.2018.2877208.
- 7) Christakou C., Stafylopatis A., 2005. Hybrid Movie Recommender System Based on Neural Networks. *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*. DOI:10.1109/ISDA.2005.9.
- 8) David D.L., Bouldin D.W., 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-1(2): 224–227. DOI:10.1109/TPAMI.1979.4766909



- 9) Draisma J., Horobeț E., Ottaviani G., Sturmfels B., Thomas R., 2014. The Euclidean Distance Degree. arXiv:1309.0049.
- 10) Gupta U., Patil N., 2015. Recommender System Based On Hierarchical Clustering Algorithm Chameleon. IEEE International Advance Computing Conference (IACC). DOI: 10.1109/IADCC.2015.7154856.
- 11) Harper M.F., Konstan J.A., 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4).
- 12) Haruna K., Ismail M. A., Damiasih D., Sutopo J., 2017. A Collaborative Approach for Research Paper. PloS One, 12(10): 1-17.
- 13) Karimi, M., Jannach, D. & Jugovac, M.(2018). News recommender systems – Survey and roads ahead Information Processing & Management, 54(6): 1203-1227.
- 14) Koohi H., Kiani K., 2017. A New Method To Find Neighbor Users That Improves The Performance Of Collaborative Filtering. Expert Systems with Applications: An International Journal, 83(C): 30-39.
- 15) Kumar M., Yadav D.K., Singh A., Gupta V.K., 2015. A Movie Recommender System: MOVREC. International Journal of Computer Applications, 124(3).
- 16) Lekakos G., Caravelas, P., 2008. A Hybrid Approach for Movie Recommendation. Multimed Tools Appl, 36: 55–70.
- 17) Levandowsky M., David W., 1971. Distance Between Sets. Nature, 234(5): 34–35.
- 18) Mahadevan A., Arock M., 2016. A Study and Analysis of Collaborative Filtering Algorithms for Recommender Systems. International Journal of Circuit Theory and Applications, 9(27): 127-136.
- 19) Mierswa I., Klinkenberg R., 2019. Rapidminer Studio 9.1: Data Science, Machine Learning, Predictive Analytics.

- 20) Pazzani M.J., 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13: 393–408.
- 21) Pazzani M.J., Billsus D., 2007. Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. Eds. *The Adaptive Web. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg. 4321.
- 22) Pearson K., 1895. Notes on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58: 240–242.
- 23) Pelleg D., Moore A., 2000. X-Means: Extending K-Means with Efficient Estimation of The Number Of Clusters. In *Proceedings of the 17th International Conf. on Machine Learning*, 727 – 734.
- 24) Portugal I., Alencar P., Cowan D., 2018. The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. *Expert Systems with Applications*, 97: 205 – 227.
- 25) Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., 1994. Grouplens: An Open Architecture for Collaborative Filtering Of Netnews. In *Proceedings of ACM Conference on Computer Supported Cooperative Work*. DOI: 10.1145/192844.192905.
- 26) Rombouts J., Verhoef T., (Date of access: July 2019). A Simple Hybrid Movie Recommender System.  
[http://www.fon.hum.uva.nl/tessa/Verhoef/Past\\_projects\\_files/Eind\\_Rombouts\\_Verhoef.pdf](http://www.fon.hum.uva.nl/tessa/Verhoef/Past_projects_files/Eind_Rombouts_Verhoef.pdf).
- 27) Salton G., Wong A., Yang C.S., 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11).
- 28) Sridevi M., Rao R.R., Rao M.V., 2016. A survey on recommender system. *International Journal of Computer Science and Information mSecurity (IJCSIS)*, 14(5).
- 29) Schwarz G.E., 1978. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2): 461– 464.

- 30) Tüysüzoğlu G., Işık Z., 2018. Hybrid Movie Recommendation System Using Graph-Based Approach. International Journal of Computing Academic Research (IJCAR), 7(2): 29-37.
- 31) Virk H.K., Singh M., Singh A., 2015. Analysis and Design of Hybrid Online Movie Recommender System. IJIT 5(2).
- 32) Vít N., 2018. Implementation Notes for The Soft Cosine Measure. In Proceedings The 27th ACM International Conference On Information And Knowledge Management, 1639–1642.
- 33) Xiao T., Shen H., 2019. Neural Variational Matrix Factorization with Side Information for Collaborative Filtering. In: Yang Q., Zhou Z.H., Gong Z., Zhang M.L., Huang S.J., Eds. Advances in Knowledge Discovery and Data Mining. PAKDD 2019. Lecture Notes in Computer Science. Springer, Cham.
- 34) Zahra S., Ghazanfar M.A., Khalid A., Azam M.A., Naeem U., Bennett A.P., 2015. Novel Centroid Selection Approaches for K-Means Clustering Based Recommender Systems. Information Sciences, 320(1): 156-189.

## **EKLER**

### **Kayıp Hesaplama Fonksiyonları**

- **Ortalama Sapma Hata (Mean Biased Error-MBE):**

Genellikle model hatasının bir ölçüsü olarak kullanılmaz, çünkü öngörme de yüksek ama bireysel hatalarda düşük MBE üretebilir. MBE tahmindeki ortalama sapmayı yakalar. Şu şekilde tanımlanır:

$$MBE = \frac{1}{n} \sum_{i=1}^n (P_i - O_i) \quad (3)$$

- **Ortalama Mutlak Hata (Mean Absolute Error-MAE(L1):**

Tahmin doğruluğunun en basit ölçüsüne denir. Adında da anlaşılacağı gibi, mutlak hataların ortalamasıdır. Mutlak hata öngörülen değer ile gerçek değer arasındaki farkın mutlak değeridir. Sürekli değişkenler için doğruluğu ölçer. Değerlerin yönlerini dikkate almadan bir dizi tahmindeki hataların ortalama büyüklüğünü ölçer. Şu şekilde tanımlanır:

$$MAE = \frac{1}{n} \times \sum_{i=1}^n |O_i - P_i| \quad (4)$$

- **Ortalama Kare Hata (Mean Squared Error-MSE-L2):**

MSE, hata vektörünün iki normunun karesinin örnek sayısına oranını belirtir. Bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu gösterir. Karesini aldığımız için her zaman pozitif değer verir. Sıfırı yakın olan tahminleyicilerin daha iyi performans gösterdiği söylenebilir. Şu şekilde tanımlanır:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \quad (5)$$

- **Kök Ortalama Kare Hata (Root Mean Squared Error-RMSE):**

Bir model tarafından tahmin edilen değerler ile modellemekte olan ortamdan gerçekte gözlemlenen değerler arasındaki farkın sık kullanılan bir ölçüsüdür. Kök ortalama kare hatası iki veri kümesi arasında ne kadar hata olduğunu ölçer. Bir başka deyişle tahmini bir değer ile gözlemlenen veya bilinen bir değeri karşılaştırır. Şu şekilde tanımlanır:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n}} \quad (6)$$

#### **Ek Kaynaklar**

1. <http://neuralnetworksanddeeplearning.com/chap1.html>
2. [https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function)
3. <https://agrimetsoft.com/calculators>
4. <http://neuralnetworksanddeeplearning.com/chap2.html>

## **ÖZGEÇMİŞ**

Cemdeniz Çay 1995'te Mersin'de doğdu. Lise öğrenimini İçel Kolejinde tamamladı. 2014 yılında girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde eğitim-öğretim hayatına devam ediyor. Stajını Ankara Softtech A.Ş firmasında Mobil Uygulama Geliştirme üzerine tamamladı.

**KOCAELİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**LİSANS TEZİ**

**YAPAY ZEKA TABANLI FİLM ÖNERİ UYGULAMASI GELİŞTİRME**

**CEMDENİZ ÇAY**

**KOCAELİ 2021**