CONTEN-BASED FİLTRELEME VE COLLOBORATIVE FİLTRELEME TEKNİKLERİYLE ANİME ÖNERİ SİSTEMİ

Selimhan Aygün – Ahmet Eren Tumbul
s.aygun2019@gtu.edu.tr – a.tumbul2019@gtu.edu.tr
Elektronik Mühendisliği Bölümü, GTÜ, Kocaeli, Türkiye

ÖZET

Bu projede, content-based (içerik temelli) ve colloborative (işbirlikçi) filtreleme teknikleri kullanarak öneri sistemi üzerine çalışıldı. İki filtreleme tarzı ayrı olarak incelendi.

Anahtar Kelimeler: Makine Öğrenmesi, Sinir Ağları, Öneri Sistemleri, Content-based Filtreleme, Coloborative Filtreleme

I. GİRİŞ

Bu proje, makine öğrenmesi yöntemlerini kullanarak bir öneri sistemi geliştirme sürecini ele almaktadır. Öneri sistemleri, kullanıcının geçmiş davranışlarına dayanarak onlara özel öneriler sunan ve kullanıcının ilgi alanlarını ve tercihlerini tahmin etmeyi amaçlayan sistemlerdir. Bu sistemler, büyük veri kümelerini analiz ederek kullanıcıların benzer davranışlarını ve tercihlerini belirlemek için karmaşık algoritmalar kullanır ve bu bilgileri kullanarak kullanıcıya özel öneriler sunar.

Content-based filtreleme, özellikle kullanıcının geçmiş tercihleri, beğenileri veya içerikle ilgili diğer özelliklere dayalı olarak benzer içerikleri önerme veya gösterme amacını taşır. Ancak bu çalışmada content-based filtreleme içeriklerin kendi içindeki özelliklerine göre yapıldı.

Collaborative filtreleme ise, kullanıcıların benzer tercihlerine dayanarak önerilerde bulunur. Bu yöntemde, kullanıcıların geçmiş davranışları ve tercihleri analiz edilir ve benzer davranışlara sahip diğer kullanıcıların tercihlerine göre öneriler yapılır.

Bu raporda, bir öneri sistemi geliştirme sürecinin temel adımlarını anlatmak ve bu adımların nasıl gerçekleştirileceğini açıklanacaktır. Bulunan veriler hazırlanacak, farklı makine öğrenmesi algoritmalarını tanıtılacak ve bu algoritmalar uygulanacak. Son olarak, öneri sisteminin performansını değerlendirmek için kullanılan metrikleri ele alınacak.

Veri seti Kaggle üzerinden güncel olarak hazırlanmış "Anime Dataset 2023" adlı çalışmadan alındı.

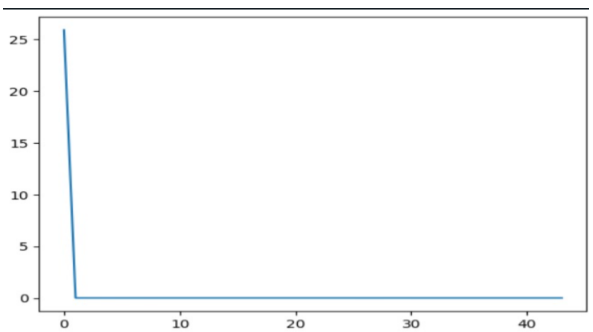
II. TEORİ VE YÖNTEM

Content based öneride 2 ayrı kod üzerinden deneme yapılmıştır, bunların ilkinde feature vektorler Cosine simularity metodu kullanılıp birbirlerine yakınlıkları ölçülerek girilen animenin vektorüne en yakın vektörler çıktıda öneri olarak sunulur[1]. Feature vektörler ise anime veri setindeki; animenin özeti, janraları, popülerlik sıralaması ve yapan stüdyonun count vectorizer ile ağırlık vektörlerine dönüştürülmeleriyle oluşur.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (2.1)$$

İkinci kodda ise sinir ağları ve train kısmı eklenir. 3 fully connected katmanlı ve 512 dimli bir sinir ağı modeli ve bunun yanında katmanlar birbirlerine RELU ile bağlanmıştır. Bu sinir ağı modelinin karmaşıklığı arttırmasının yanında lineer olmaması, veriler arasındaki karmaşık ilişkileri tespit etmesini kolaylaştırır.

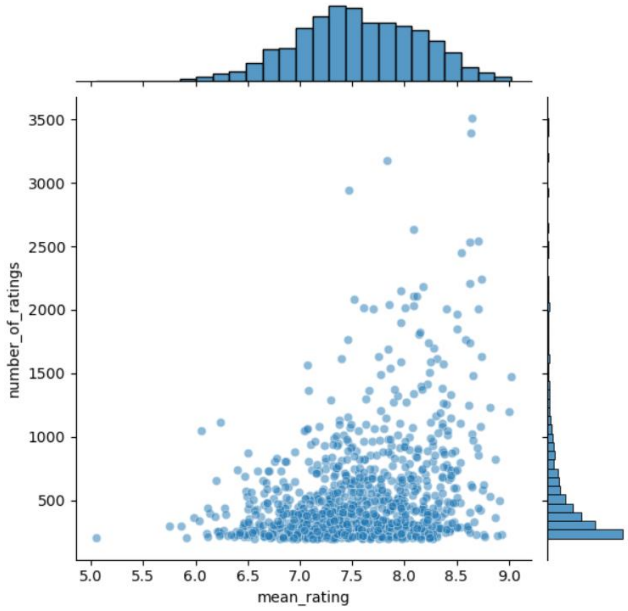
Sinir ağı kullanılan modelin kayıp grafiği çıktısı ağağıdaki gibidir:



Şekil 2 Kayıp Fonksyionu Grafiği

Şekle bakıldığında loss çok hızlı bir şekilde düşüp sabit kaldığı görünmekte, bu da ya bizim kullandığımız train yöntemiyle ya da verinin doğasında bir sıkıntı olduğu göstermektedir.
Colloborative filtreleme kullanıcı tabanlı ve öğe tabanlı olmak üzere ikiye ayrılır. Kullanıcı tabanlıda kullanıcının beğenileri, benzer profillere sahip diğer kullanıcıların beğenileriyle karşılaştırılır . Öğe tabanlı ise beğenilme durumu, benzer içeriklere sahip diğer içeriklerle karşılaştırılır Aralarındaki temel fark benzerlik matrislerinin farklı olmasıdır. Bu yüzden sadece kullanıcı tabanlı yöntem kullanıldı.

Veriler uygun hale getirildikten sonra Keşifçi veri analizi (Exploratory Data Analysis (EDA)) yapıldı. Kullanıcıların oylamalarıyla ortalama anime skorlarının dağılımı Şekil 2'de gözükmektedir. Ayrıca en çok oylanan ilk 10 animenin sıralanışı Şekil 3' de verilmiştir.



Şekil 2 Ortalama Anime skorlarının dağılımı

	Anime Title	mean_rating	number_of_ratings
2964	Fullmetal Alchemist	8.647763	3509
2142	Death Note	8.638561	3392
1246	Bleach	7.837107	3180
7001	Naruto	7.469111	2946
2684	Elfen Lied	8.087666	2635
1929	Cowboy Bebop	8.705350	2542
1855	Code Geass: Hangyaku no Lelouch	8.626382	2532
9577	Suzumiya Haruhi no Yuuutsu	8.544712	2449
3745	Sen to Chihiro no Kamikakushi	8.742640	2242
7649	Ouran Koukou Host Club	8.629864	2210

Şekil 3 En çok oylanan animelerin sıralanışı

Algoritmanın ilk adımında normalizasyon yapıldıktan sonra, kullanıcılar ve öğeler matris haline getirildi, ardından kullanıcıların skorları arasındaki ilişkiyi ölçmek için "Pearson Korelasyonu" yöntemi kullanıldı. Korelasyon, Denklem (1.1) kullanılarak sağlanıyor. Kullanıcılar arasındaki lineer ilişkinin sayısal değerleri matris olarak alındı.

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}} \qquad (2.2)$$

Ardından kullanıcıların etkileşimlerinden ortak kullanıcılar bulundu (Şekil 4). Diğer kullanıcılar tarafından sevilen ve seçilen kullanıcı tarafından sevilebilecek öğeler kullanıcının tahmini oylamasının da bulunabileceği şekilde tespit edildi (Şekil 5).

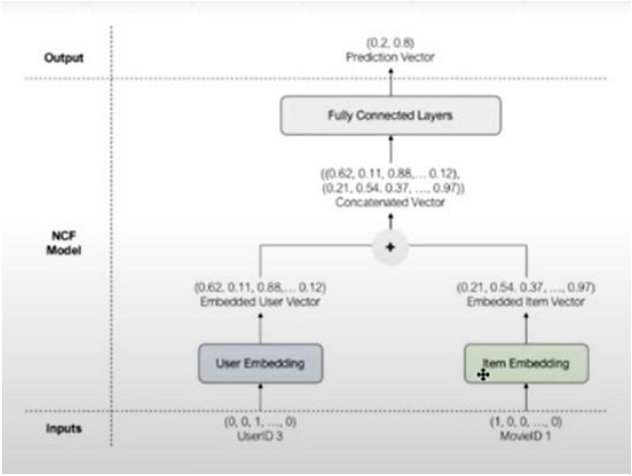
```
The similar users for user 4 are user_id
9813    1.0
8286    1.0
5407    1.0
4288    1.0
6959    1.0
4319    1.0
9792    1.0
4358    1.0
6894    1.0
6885    1.0
Name: 4, dtype: float64
```

Şekil 4 Benzer kullanıcılar

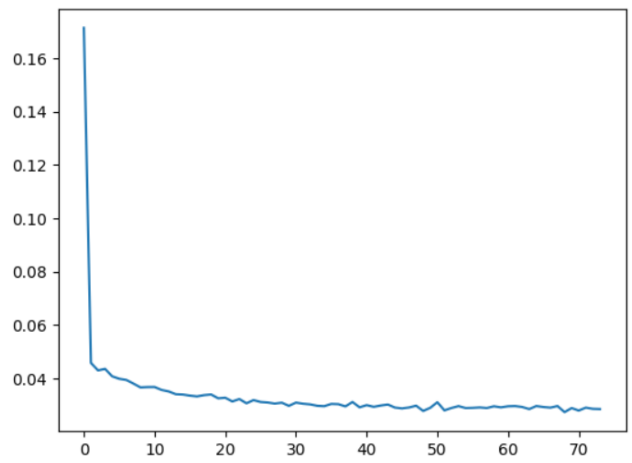
	anime	anime_score	predicted_rating
3	Azumanga Daiou The Animation	2.419355	9.602361
6	Bishoujo Senshi Sailor Moon S	2.419355	9.602361
68	Rurouni Kenshin: Meiji Kenkaku Romantan - Seis...	2.419355	9.602361
9	Boogiepop wa Warawanai	2.419355	9.602361
26	Gravitation	2.000000	9.183007
61	Ouran Koukou Host Club	2.000000	9.183007
45	Kino no Tabi: The Beautiful World	1.785714	8.968721
49	Lucky☆Star	1.500000	8.683007
67	Rurouni Kenshin: Meiji Kenkaku Romantan	1.419355	8.602361
69	Rurouni Kenshin: Meiji Kenkaku Romantan - Tsui...	1.419355	8.602361

Şekil 5 Önerilen Animeler

Bu yöntemin ardından matris ayrışımı yöntemi kullanıldı. Bu yöntemde kullanıcı ve içeriklerden oluşan tek bir matris ikiye ayrılıyor ve çarpımı yapılıyor. Sinir ağları kullanılarak bu yöntem, daha etkin bir şekilde kullanılmak istendi (Şekil 6). İkiye ayrılıp embedding yapılan sistem daha sonra tek bir tam bağlayıcı katman sayesinde bağlanıyor.



Şekil 6 Nöral İşbirlikçi Filtreleme



Şekil 7 Loss Function

Veriler eğitildikten sonra çıkan kayıp fonksiyonu şekil 7'de görülmektedir.

```
precision @ 100: 0.9826900929095181
recall @ 100: 0.9949666367671463
F1 score @ 100: 0.9887902607713253
```

Şekil 8 Metirkler

III. ANALİZ VE YORUM

Şekil 1'e bakıldığında loss çok hızlı bir şekilde düşüp sabit kaldığı görünmekte, bu da ya bizim kullandığımız train yöntemiyle ya da verinin doğasında bir sıkıntı olduğu göstermektedir.

Kullanıcı tabanlı yöntemde matrislerin çarpılması ve tutulması sonucunda çıkan bellek kullanım yoğunluğu yüzünden verilerde azaltılma yapıldı.

Çoğu yöntemde sistemin doğruluğunun tahminini yapabilecek metrikler alınamadı.

KAYNAKÇA

[1] https://www.kaggle.com/datasets/dbdmobile/myanimelist-dataset
[2] https://medium.com/@toprak.mhmt/content-based-recommender-system-bdfc60b1bee8#:~:text=A%20Content-based%20recommendation%20system,perhaps%20even%20liked%20those%20items.
[3] https://developers.google.com/machine-learning/recommendation/content-based/basics?hl=tr
[4] https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/
[5] https://www.youtube.com/watch?v=1xtrIEwY_zY&t=3671s
[6] https://medium.com/data-science-in-your-pocket/recommendation-systems-using-neural-collaborative-filtering-ncf-explained-with-codes-21a97e48a2f7
[7] https://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/recommender/neural_collaborative_filtering
[8] https://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/recommender/neural_collaborative_filtering