

Raddio: Yeni Nesil İnternet Radyosu

Emircan Kalyoncu¹, B. Tefik Akgün¹

¹Okan Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

emircan.kalyoncu@gmail.com, tevfik.akgun@okan.edu.tr

Özet: Müziğin digital ortamlara girmesi, müziğe olan erişimi kolaylaştırmış ve buna bağlı olarak müzik tüketimini arttırmış olmasına rağmen beraberinde büyük bir problemi getirmiştir: “Ne dinleyeceğim?” Hali hazırda ticari müzik katalogları sayısının 15 milyonu geçtiğini göz önüne alırsak, “Ne dinleyeceğim?” sorusu büyük bir problem olmuş durumda. İnsanlara dinleyebilecekleri müzikleri önermek ve keşfettirmek için akademik olarak araştırmalar başlamıştır. Raddio bu noktada, iki ana öneri sistemi yöntemi olan ortak tercihlere dayalı filtreleme ve içeriğe dayalı filtrelemeyi birleştirerek oluşturduğu kendi öneri sistemini kullanarak, dinleyicilere daha doğru önerilerde bulunmaya çalışır.

Anahtar Sözcükler: Veri Madenciliği, Öneri Sistemleri, Müzik

Raddio: The New Generation Internet Radio

Abstract: Since we are in digital age, it's now very easy to access music; but increasing number of digital music services brings out a big problem: "What am I going to listen?" With the help of academic researches; listeners are recently able to find the songs match with their mood or music taste. At this point, Raddio offers a unique recommender system which combines 2 main features: collaborative filtering and content-based filtering. Raddio's commitment is to increase the customer satisfaction which is suggesting the best options to the listeners.

Keywords: Data Mining, Recommender Systems, Music

1. Giriş

Müziğin digital ortamlara girmesi, müziğe olan erişimi kolaylaştırmış ve buna bağlı olarak müzik tüketimini arttırmış olmasına rağmen beraberinde büyük bir problemi getirmiştir: “Ne dinleyeceğim?” Hali hazırda ticari müzik katalogları sayısı 15 milyonu geçmiştir. Hiç durmadan 1 milyon şarkıyı dinlemek yedi yıldan fazla süreceğini düşünürsek, “Ne dinleyeceğim?” sorusu büyük bir problem olmuş durumda. Müziğin büyük uzayında insanları kaybetmemek, dinleyebilecekleri içerikleri aramak ve bulmak için öneri sistemlerine ihtiyaç duyulmuştur [1].

Öneri sistemleri, kullanıcılara çeşitli öğeler hakkında öneriler sunmak amacıyla geliştirilmiş yazılım araçları ve bilgi keşif

teknikleri olup özellikle 90'lı yılların ortalarından itibaren önemli bir araştırma alanı haline gelmiştir. İnsanlara kişiselleştirilmiş önerilerde bulunmak, yeni içerikler keşfetmelerini sağlamak halen sıcak ve ilgi çekici bir alan olduğundan hem akademik hem de sektörel birçok araştırma ve geliştirme yapılmıştır [2].

Bu çalışmanın 2. bölümünde başlıca öneri sistemlerinde kullanılan yöntemler açıklanmış, 3. bölümünde bu yöntemlerin problem ve avantajlarından söz ederek, bu problem ve avantajları lehine çevirecek olan melez yaklaşımdan bahsedilmiştir. Çalışmanın 4. bölümünde, bu yaklaşımı müzik öneri sistemi olarak kullanan Raddio'nun kullanımı arttırmak ve hız gibi problemlere getirdiği ufak çözümlerden ipuçları verilmiş, 5. bölümünde ise, çeşitli

veri setleri ile yapılan deneysel deneylere yer verilmiştir. Çalışmanın son bölümünde sonuçlar sunulmuş ve gelecek çalışmalar hedeflenmiştir.

2. Öneri Yöntemleri

Öneri sistemlerinde bir çok farklı algoritma ve yöntem olduğu halde içlerinde en popüler olanları içeriğe dayalı filtreleme ve ortak tercihlere dayalı filtrelemedir. İçeriğe dayalı filtreleme sisteminde, kullanıcının geçmişte hoşlandığı parçalara benzer hoşlanabileceği parçaları bulmaya çalışır. Esas alınan konu parçalar arasında ki benzerliktir. Bu benzerlikler aynı sanatçı, albüm, tür, ritim gibi çok çeşitli değerlerle belirlenir [3]. Örneğin; Şekil 1'de verilen benzerliklere göre, Şekil 2'de kullanıcıların beğenilerinin benzerleri önerilmiştir.

	Şarkı 3	Şarkı 4	Şarkı 5
Şarkı 1	%20	%80	%40
Şarkı 2	%10	%70	%20
Şarkı 3	%100	%20	%70
...	...	...	...

Şekil 1 – Parçalar arası benzerlik

	Şarkı 1	Şarkı 2	Şarkı 3	Şarkı 4	Şarkı 5
A	5	X	2	4	1
B	4	5	3	X	2
...	...	...	...	...	...

Şekil 2 – İçeriğe dayalı filtreleme

Ortak tercihlere dayalı filtreleme sisteminde ise, kullanıcıların tercihlerinin ortaklığına göre öneriler yapılmaktadır. Esas alınan konu kullanıcıya benzeyen kullanıcıların beğendiği müzikleri önermektir. Kullanıcı benzerlikleri beğenilen müziklerle belirlenir [3]. Örnek olarak verilen Şekil 3' de A ve B kullanıcılarının beğeni benzerliklerine göre

bu kullanıcılar benzer kabul edilmiş ve böylece B kullanıcıasına yapılabilecek öneri gösterilmiştir.

	A	B	C
Şarkı 1	5	5	1
Şarkı 2	1	2	4
Şarkı 3	5	5	4
Şarkı 4	2	3	3
Şarkı 5	4	X	...
...	...	...	...

Şekil 3 – Ortak tercihlere dayalı filtreleme

Ortak tercihlere dayalı filtreleme sistemlerinde kullanıcı benzerliklerini hesaplayabilmek için birçok benzerlik ölçütü vardır. Bu çalışmada kullandığımız Manhattan distance, benzerlik ölçütleri içinde en yaygın kullanılan ve en kolay hesaplanan algoritmalarından biridir [4]. Basitçe söylemek gerekirse iki kullanıcı arasındaki uzaklığı bulmak için parçalara verilen oyların farkının mutlak değerleri toplamını hesaplamamız gerekmektedir.

$$d = \sum_{i=1}^n |W_{ai} - W_{bi}|$$

Şekil 4 - Manhattan distance formülü

Şekil 4'de verilen formülü Şekil 3'de ki örnek veriler üzerinde uygularsak, A – B kullanıcıları arasındaki uzaklığı 2, A – C kullanıcıları arasındaki uzaklığı ise 9 olarak buluruz. Bu durumda A ve B kullanıcıları birbirine yakındır diyebiliriz.

3. Melez Yaklaşım

İki geleneksel yöntem olan içeriğe dayalı filtreleme ve ortak tercihlere dayalı filtrelemenin problemlerini çözmeyi ve

avantajlı olduđu yönlerini birleştirmeyi amaçlayan bir yaklaşımdır. Bu yaklaşımda başta ölçeklenebilirlik, önerilerin kalitesi, esneklik, soğuk başlangıç gibi problemler karşılaştırılır ve avantajları ortaya çıkararak sistemin verimini arttırmak hedeflenir [5].

Ölçeklenebilirlik problemi çoğu sistemde olduđu gibi öneri sistemlerinin de problemlerinin başında gelir. İyi bir öneri sistemi önerilerini hızlı ve verimli bir biçimde sunmalıdır. Ortak tercihlere dayalı yöntem her öneri için bütün veritabanını kullanmalıdır. Bu yüzden çok yavaş çalışan bir yöntemdir. Oysa ki içeriğe dayalı yöntem de çok daha küçük verilerle öneriler yapabildiğinden daha hızlı çalışır. Ancak önerilerin kalitesi düşünüldüğünde, tüm veritabanını kullandığından dolayı daha doğru öneriler yapacağı için ortak tercihlere dayalı yöntem nispeten daha iyidir. Yine aynı şekilde her öneri yapıldığında veritabanını kullandığı için değişiklik yapmak çok kolay olduğundan esneklik bu yöntemde problem değildir [6]. İçeriğe dayalı filtreleme yönteminin en önemli avantajları arasında gösterilen soğuk başlangıç sorunu, başka hiçbir kullanıcı ile eşleşemeyen kullanıcılarda yaşanır. Bu eşleşememe, kullanıcının sistemde yeterli ölçüde dinleme, beğenme gibi kayıtlarının olmadığı durumlarda ortaya çıkar. Bu gibi durumlarda ortak tercihler olmadığı için içeriğe dayalı filtreleme yöntemi daha başarılıdır.

Sonuç olarak tüm problemler ve avantajlar göz önünde bulundurulduğunda bu çalışma için kullanılan yaklaşım; öncelikle kullanıcıya ortak tercihlere dayalı filtreleme ile beş yakın kullanıcı bulunur. Benzer kullanıcıların beğendiğı parçalar ile öneri yapılacak kullanıcının beğendiğı parçaların benzerlikleri dikkate alınarak içeriğe dayalı filtreleme sonucu parçalar kullanıcıya önerilir.

4. Tasarım ve Uygulama

Raddio, web tabanlı bir şarkı öneri sistemidir. Kullanıcılara dinleme geçmişlerine göre

gerçek zamanlı şarkı önerilerinde bulunur. Bu önerileri daha doğru yapabilmek, sistemin hızı ve verimli olması için içeriğe dayalı filtreleme ve ortak tercihlere dayalı filtreleme sistemlerini melez bir şekilde kullanmaktadır.

4.1. Yazılımı

Raddio, geliştirme araçları, basitliği, kodlama kalitesi ve hızı gibi birçok açıdan değerlendirilerek, bir NodeJS [10] web framework'ü olan Sails [11] üzerinde kodlanmıştır. Kodlamanın basitliğine ve kolaylığına özellikle dikkat edilerek yazılımı geliştirilmiştir.

4.2. Veri Seti

Raddio, Last.fm [7]'in milyonlarca kayıttan oluşan veri setini kullanmaktadır. Bu veri setinde şarkı, sanatçı, albüm, tür gibi birçok şarkı metadatası dışında yaklaşık 100.000 kullanıcının 1 yıla kadar dinleme ve beğeni geçmişleri tutulmaktadır.

4.3. Veritabanı Tasarımı

Raddio, herhangi bir veritabanı ürünü kullanmadan verileri json olarak diskte saklamaktadır. Genellikle veritabanı çözümlerinde yapılan ilişkisel veritabanı mantığı yerine ilişkisel olmayan veritabanı mantığı [8] ile yaklaşılarak veritabanı işlemlerinde olabildiğince hız kazanmayı hedeflemiştir. Bu hızı ve verimi sağlayabilmek adına her öneri hesaplaması için belirli bir frekans belirlemiş ve bu frekans boyunca yalnızca bir kez hesaplama yapmıştır. Ayrıca Redis [9] ürününü kullanarak sık kullanılan verileri hafızaya alarak sistemin daha da hızlı olması sağlanmıştır.

4.4. Nasıl Çalışır?

Raddio, ilk olarak kullanıcıdan radyo başlatması için bir sanatçı, şarkı, tür, ruh hali veya aktivite seçmesini istemektedir. Seçtiğı seçenek doğrultusunda bir radyo başlatılır ve

kullanıcının davranışları gözlemlenir. Kullanıcı bir şarkıyı beğendiğinde veya beğenmediğinde, o şarkıdan sonra gelecek şarkıları o anki ve geçmiş tercihlerine göre gerçek zamanda hesaplar ve öneriler sunar. Anlık beğenileri de hesaba kattığı için kullanıcıların geçmiş ve o an arasında ki beğeni farklılıklarından doğan problemler nispeten daha aşağı çekilir. Bu sayede anlık olarak tanımlayabileceğimiz ruh hali ve aktivite radyolarında daha doğru öneriler kullanıcıları mutlu edecektir.

4.5. Önyüz Tasarımı

Raddio'nun kullanıcıyla etkileşimde olacağı önyüzü oldukça kolay ve kullanıcı dostu olarak tasarlanmıştır. Birçok servis gibi Raddio da kullanıcılarına daha etkin ve verimli bir deneyim yaşatmak adına bu iki önemli tasarım kalıbını benimsemiştir.

5. Deneyisel Değerlendirmeler

Değerlendirmeler için farklı büyüklükteki veri setleri ile sistem test edilmiştir. Farklı kullanıcı sayıları ve bu kullanıcıların 1 aylık, 3 aylık ve 6 aylık dinleme geçmişleri baz alınarak hazırlanan veri setlerine göre öneri başarı oranları Şekil 5'te verilmiştir.

Kullanıcı Sayısı	1 Ay	3 Ay	6 Ay
100	%13	%26	%40
1.000	%38	%49	%60
10.000	%63	%75	%87

Şekil 5 – Veri setlerine göre öneri başarı oranları

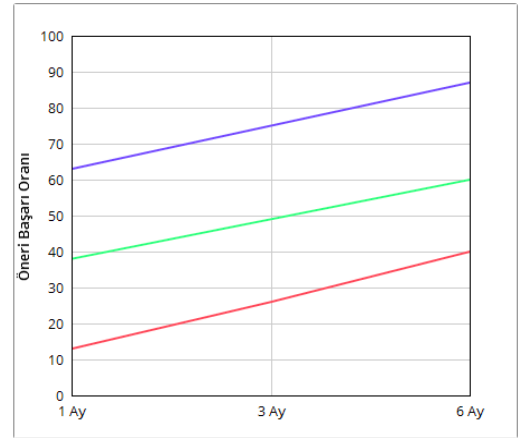
Başarı oranını hesaplarırken tüm kullanıcılar için ayrı ayrı yapılan önerilerin başarı oranları hesaplanmış ve genel sonuç için bu oranların ortalaması alınmıştır. Şekil 6'daki başarı oranı formülünde de görüleceği gibi bir kullanıcı için öneri başarı oranını (BO) bulmak için önerilen şarkılarda beğendiği şarkı sayısını (BŞS), önerilen şarkı sayısına bölmek ve 100

ile çarpmak gerekir. Raddio her zaman sonraki 4 şarkıyı önereceğinden ÖSS hep 4'dür diyebiliriz.

$$BO\% = BŞS / ÖSS * 100$$

Şekil 6 – Öneri başarı oranı formülü

Örneğin; önerilen şarkılarda beğeni sayısı 3 ise önerinin başarı oranı %75 hesaplanır.



K : 100 | Y : 1.000 | M : 10.000
kullanıcı

Şekil 7 – Veri setlerine göre öneri başarı oranı grafiği

6. Sonuçlar

Bu çalışmada, bir müzik öneri sistemi olan Raddio uygulamasının ilk aşamaları başarıyla tamamlanmıştır. Kullandığı melez yaklaşım ile birçok problemi geride bırakmıştır. Ayrıca bu melez yaklaşım özellikle müzik gibi fazla sayıda kullanıcı ve öğe olan öneri sistemlerinde en iyi yöntemdir. Kullanıcı sayısı ve kullanım oranı arttıkça önerilerin başarı oranı da artacaktır.

Graf veritabanlarının öneri sistemleri için daha hızlı ve verimli olduğu günümüzde araştırma konusudur [12]. Gelecek çalışmalarda öncelikli olarak bir graf veritabanı kullanarak hız ve verimliliği deneyimlemek hedeflenmiştir.

7. Kaynaklar

[1] Uitdenbogerd, Alexandra L., and Ron G. van Schyndel. "A Review of Factors Affecting Music Recommender Success." *ISMIR*. Vol. 2. 2002.

[2] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *Knowledge and Data Engineering, IEEE Transactions on* 17.6 (2005): 734-749.

[3] Celma, Oscar. *Music Recommendation and Discovery: The Long Tail, Long Tail, and Long Play in the Digital Music Space*. Springer, 2010.

[4] Maneeroj, Saranya, Hideaki Kanai, and Katsuya Hakozaiki. "Combining Dynamic Agents and Collaborative Filtering without Sparsity Rating Problem for Better Recommendation Quality." *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*. 2001.

[5] Yoshii, Kazuyoshi, et al. "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model." *Audio, Speech, and Language Processing, IEEE Transactions on* 16.2 (2008): 435-447.

[6] G. Demir, T. Kaya, C. Ogut and A.C. Sag "CORS: A Hybrid Music Recommender System" 2014.

[7] Last.fm : <http://www.last.fm>

[8] <http://en.wikipedia.org/wiki/NoSQL>

[9] Redis : <http://redis.io>

[10] NodeJS : <http://nodejs.org>

[11] Sails : <http://sailsjs.org>

[12] Mirza, Batul J. *Jumping connections: A graph-theoretic model for recommender systems*. Diss. Virginia Polytechnic Institute and State University, 2001.