TAVSİYE SİSTEMLERİNDE NODE2VEC GÖMME VE SİNİRSEL İŞBİRLİKÇİ FİLTRELEME KULLANIMI

Yolande Margaret GOSWELL

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

2020

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

TAVSİYE SİSTEMLERİNDE NODE2VEC GÖMME VE SİNİRSEL İŞBİRLİKÇİ FİLTRELEME KULLANIMI

(Recommender Systems Using Node2vec Embedding And Neural Collaborative Filtering)

Yolande Margaret GOSWELL

Bilgisayar Mühendisliği Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Tolga AYDIN

ERZURUM

Haziran, 2020

KABUL VE ONAY TUTANAĞI

Yolande Margaret GOSWELL tarafından hazırlanan "TAVSİYE SİSTEMLERİNDE NODE2VEC GÖMME VE SİNİRSEL İŞBİRLİKÇİ FİLTRELEME KULLANIMI" başlıklı çalışması 05 / 06 / 2020 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Bilgisayar Mühendisliği Ana Bilim Dalı, Bilgisayar Bilim Dalı Yüksek Lisans tezi olarak kabul edilmiştir.

Jüri Başkanı:	Prof. Dr. Abdulsamet HAŞILOĞLU *Atatürk Üniversitesi*	Aslı ıslak imzalıdır.
Danışman:	Dr. Öğretim Üyesi Tolga AYDIN *Atatürk Üniversitesi*	Aslı ıslak imzalıdır.
Jüri Üyesi:	Dr. Öğretim Üyesi Saffet Gökçen ŞEN *Erzurum Teknik Üniversitesi*	Aslı ıslak imzalıdır.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Mehmet KARAKAN
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak Danışman: Dr. Öğr. Üyesi Tolga AYDIN danışmanlığında sunulan "**TAVSİYE SİSTEMLERİNDE NODE2VEC GÖMME VE SİNİRSEL İŞBİRLİKÇİ FİLTRELEME KULLANIMI**" başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	*0*	**30**
Kuramsal Temeller	*0*	**30**
Materyal ve Yöntem	*6*	**35**
Bulgular	*6*	**20**
Tartışma	*0*	**20**
Tezin Geneli	*12*	**25**

Not: *Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının* ***%5'den*** *büyük olmaması gerekir.*

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	**Tez Danışmanı**
Yolande Margaret GOSWELL	Danışman: Dr. Öğr. Üyesi Tolga AYDIN
5.6.2020	**5.6.2020**
İmza:	İmza:

***** Tez ile ilgili YÖKTEZ'de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun …./…/…. tarih ve ……. sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐Enstitü Yönetim Kurulunun …./…/…. tarih ve ……. sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir

TEŞEKKÜR

Bu tezin yürütülmesinde yardımlarıyla yol gösteren danışman hocam Sayın Dr. Öğr. Üyesi Tolga AYDIN'a teşekkürlerimi sunuyorum.

Ayrıca tez çalışmam süresince maddi ve manevi desteklerini esirgemeyen aileme ve arkadaşlarıma teşekkürü bir borç bilirim.

Yolande Margaret GOSWELL

ÖZET

YÜKSEK LİSANS TEZİ

TAVSİYE SİSTEMLERİNDE NODE2VEC GÖMME VE SİNİRSEL İŞBİRLİKÇİ FİLTRELEME KULLANIMI

Yolande Margaret GOSWELL

Danışman: Dr. Öğr. Üyesi Tolga AYDIN

Amaç: Bu tezde, tavsiye sistemlerinin geliştirilmesinde kullanılan farklı tekniklerin kapsamlı bir çalışması sunulmaktadır. Çalışmamızda, farklı veri tiplerinin tavsiye sistemlerine nasıl dâhil edildiği konusu da araştırılmıştır. Bu tezde, sinirsel işbirlikçi filtrelemeyle birleştirilmiş grafsal bir veri temsili sunularak node2vec-NCF adlı bir model sunulmuş ve tavsiye etme süreçleri gerçekleştirilmiştir.

Yöntem: Node2vec olarak bilinen bir graf gömme yöntemi kullanarak, kullanıcı ve öğe varlıklarını verilerin yapısal bütünlüğünü yakalayacak bir şekilde temsil ederiz. Bu, daha sonra, verilen bir kullanıcı-öğe çiftinin derecesini tahmin eden bir sinirsel işbirlikçi filtreleme çerçevesine girdi olarak kullanılmaktadır.

Bulgular: Modelimiz doğruluk metriklerinin yanı sıra hassasiyet ve hatırlama metriklerinde geleneksel yöntemlerden daha iyi performans göstermiştir. Modelimizin performansı, node2vec gömme yönteminin veri setinde bulunan gizli özellikleri doğru yakalama yeteneğiyle ilişkilendirilebilir.

Sonuç: Modelimiz geleneksel tavsiye sistemi algoritmalarından daha iyi performans göstermiştir. Model, işbirlikçi filtreleme için derecelendirme değerlerinden yararlansa da, metin ve görüntü gibi çeşitli veri türlerini içerecek veya karma (hibrit) bir sistemin bir parçası olacak şekilde genişletilme potansiyeline sahiptir.

Anahtar Kelimeler: tavsiye sistemi, sinirsel işbirlikçi filtreleme, node2vec, graf gömme

Haziran 2020, 54 sayfa

ABSTRACT

MASTER THESİS

RECOMMENDER SYSTEMS USING NODE2VEC EMBEDDING AND NEURAL COLLABORATIVE FILTERING

Yolande Margaret Goswell

Supervisor: Asst. Prof. Dr. Tolga AYDIN

Purpose: In this thesis, we present a comprehensive study of the different techniques used in recommender systems development. We also looked at how different datatypes are incorporated into recommender systems. We present a graph data representation combined with neural collaborative filtering to perform recommendations, called the node2vec-NCF model.
Method: Using a graph embedding method known as node2vec we represent user and item entities in a manner that captures structural integrity of the data. This is then used as input to a neural collaborative filtering framework that predicts the rating for a given user-item pair.
Finding: Based on the evaluation of our model using accuracy and performance metrices (precision and recall), our model performed better than traditional methods. The performance of our model can be attributed to the ability of the node2vec embedding method to truly capture hidden features found in the dataset.

Results: Our model performs better than traditional recommender system algorithms. Even though the model makes use of rating values for collaborative filtering, it has the potential to be extended to include varied data types such as images, text or included as part of a hybrid system.

Keywords: recommender systems, neural collaborative filtering, node2vec, graph embedding

June 2020, 54 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU ii
TEŞEKKÜR iii
ÖZET iv
ABSTRACT v
İÇİNDEKİLER vi
TABLOLAR DİZİNİ viii
ŞEKİLLER DİZİNİ ix
KISALTMALAR DİZİNİ x
GİRİŞ 1
Amaç 2
Katkı 3
Tezin Akışı 3
KAYNAK ÖZETLERİ 4
Terim Tanımı 4
Genel Tavsiye Sistemleri 6
Literatür Taraması 9
MATERYAL ve YÖNTEM 17
Skip gram Modeli 17
Node2vec Gömmesi 18
Node2vec gömme örnekleme stratejisi. 18
Sinirsel İşbirlikçi Filtreleme (NCF) 20
Yapay sinir ağları. 20
NCF'nin Çok Katmanlı Algılayıcı Örneği 21
Verilerin Grafsal Gösterimi 23
Yöntem Özeti 24
Sözde Kod (Pseudocode) 25
Modelin Tasarımı 27
Modelin Uygulaması 28
Veri Setleri 29
Deneysel Yöntem 34

Değerlendirme 35
Değerlendirme Ölçütleri 35
Bilgisayar Özellikleri 37
Eğitim ve Deney Veri Setleri 37
ARAŞTIRMA BULGULARI ve TARTIŞMA 39
SONUÇ 47
KAYNAKLAR 49
EKLER 52
EK 1 52
ÖZGEÇMİŞ 55

TABLOLAR DİZİNİ

Tablo 1. Gerçek tavsiye sistemleri veri kümesi örneği 30
Tablo 2. Veri kümelerinin temel özellikleri 30
Tablo 3. Veri setlerinin kaynakları 30
Tablo 4. Çeşitli veri kümeleri için optimize edilmiş çağ, batch_size, bırakma ve düzenleme parametre değerlerini gösteren tablo 35
Tablo 5. Tüm veri kümeleri için test veri setlerinin derecelendirmeye dayalı istatistiksel özellikleri 39
Tablo 6. Modelin eğitim ve test sonuçları 40
Tablo 7. Hassasiyet, hatırlama ve isabet oranı metriklerinin üst 5 ve 10 öğe için değerlendirme sonuçları 40
Tablo 8. MovieLens 1m için hassasiyet ve hatırlama oranları sonuçları (Palumbo *et al.* 2018) 44
Tablo 9. Modelin, sürpriz tavsiye sistemi kütüphanesi kullanılarak uygulanan temel modellerle karşılaştırılma sonuçları 44
Tablo 10. Sürpriz tavsiye sistemi kütüphane modellerinin ve modelimizin (node2vec-ncf) hassasiyet ve hatırlama değerlerinin tüm veri setleri için sonuçlarının karşılaştırılması 45

ŞEKİLLER DİZİNİ

Şekil 1. Kullanıcı-öğe matrisi 5
Şekil 2. Skip gram modeli için oluşturulan veri örneği 18
Şekil 3. Node2vec algoritmasında geçiş olasılığının nasıl hesaplanacağını gösteren diyagram (Grover and Leskovec 2016) 19
Şekil 4. Graf ve metinlerde Korpus üretimi 20
Şekil 5. Sinir ağının bileşenlerini gösteren bir diyagram 21
Şekil 6. Sinir işbirlikçi filtreleme çerçevesi (He *et al*. 2017) 23
Şekil 7. Graf gösterimi (veri kümesi dönüşümü) 24
Şekil 8. Önerilen Node2vec-NCF metodolojimizin şematik bir temsili 27
Şekil 9. Node2vec-NCF modelimizin sinir ağı bileşeninin mimarisini gösteren diyagram 29
Şekil 10. MovieLens 100k veri kümesinin tüm işlemlerde derecelendirme dağılımı 31
Şekil 11. MovieLens 1m veri kümesinin tüm işlemlerde derecelendirme dağılımı 31
Şekil 12. Öğe başına derecelendirme sayısı dağılımı (MovieLens 100k veri kümesi) 31
Şekil 13. Öğe başına derecelendirme sayısı dağılımı (MovieLens 1m veri kümesi) 32
Şekil 14. Amazon veri kümesinin tüm işlemlerde derecelendirme dağılımı 32
Şekil 15. Öğe başına derecelendirme sayısı dağılımı (Amazon veri kümesi) 32
Şekil 16. Books veri kümesinin tüm işlemlerde derecelendirme dağılımı 33
Şekil 17. Öğe başına derecelendirme sayısı dağılımı (Books veri kümesi) 33
Şekil 18. Fine Foods veri kümesinin tüm işlemlerde derecelendirme dağılımı 33
Şekil 19. Öğe başına derecelendirme sayısı dağılımı (Fine Foods veri kümesi) 34
Şekil 20. Tüm veri kümelerindeki öğelerin tahmini kapsamı 40
Şekil 21. Tüm veri kümeleri için en iyi önerilen N adet öğe için hatırlama değeri 41
Şekil 22. Tüm veri setleri için en iyi önerilen N adet öğe için isabet oranı. 42
Şekil 23. Tüm veri setleri için en iyi önerilen N adet öğe için hassasiyet oranı 42

KISALTMALAR DİZİNİ

BFS	Breadth First Search (Genişlik İlk Arama)
CF	Collaborative Filtering (İşbirlikçi Filtreleme)
CNN	Convolution Neural Network (Konvolüsyon Sinir Ağı)
DeepCoNN	Deep Cooperative Neural Networks (Derin Kooperatif Sinir Ağları)
DFS	Depth First Search (Derinlik İlk arama)
IOT	Internet of Things (Nesnelerin İnterneti)
JRL	Joint Representation Learning (Ortak Temsil Öğrenme)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
MLP	Multi-Layer Perceptron (Çok Katmanlı Algılayıcı)
MSE	Mean Square Error (Ortalama Kare Hatası)
NCF	Neural Collaborative Filtering (Sinirsel İşbirlikçi Filtreleme)
ReLU	Rectified Linear Unit (Rektifiyeli Doğrusal Ünite)
RMSE	Root Mean Square Error (Kök Ortalama Kare Hatası)
SVD	Singular Value Decomposition (Tekil Değer Ayrışması)
SVM	Support Vector Machine (Destek Vektör Makinesi)
TF / IDF	Term Frequency–Inverse Document Frequency (Terim Sıklığı – Ters Belge Sıklığı)
TransNet	Transformational Neural Networks (Dönüşümsel Sinir Ağları)

GİRİŞ

Günümüzde hayatımızın doğası gereği yaşamımız internet ile iç içedir. İnternet; araştırma, alışveriş ve arkadaş ile bağlantı ve iletişim kurmada, düşüncelerimizi ve görüşlerimizi paylaşmada, bilgi aramada vb. için kullanılmaktadır. Şüphesiz, bir şeyler için arama yapıldığı zaman, birçok bilgi elde edilmektedir. Bizimle ilgili olanı elde etmek için her zaman seçici davranmak gerekmektedir. Zaman içinde toplanan tüm verilerin yüzde 90'ından fazlasının son 2 yılda, yani günde oluşturulan 2,5 kentilyon baytlık bir oranda oluşturulduğu tahmin edilmektedir (Einstein 2019). Bu artışın, bilgi işlem aygıtlarının, internetin ve onunla çalışan uygulamaların erişilebilirliğindeki yükselişin bir sonucu olarak ortaya çıktığı anlaşılmaktadır. Toplumun ilgi odağı olan bu tür uygulamalar arasında, sosyal medya platformları, e-ticaret siteleri, haber siteleri ve internete bağlı cihazlar (IOT) vb. sayılabilir. Bu tür uygulamaların sürekli kullanımı, büyük miktarda veriye dönüşen dijital bir ayak izini oluşturmaktadır. Bu izlerin kullanılmasıyla, tavsiye sistemleri ilgili konuda, kullanıcıya kendileriyle ilgili olabilecek muhtemel senaryoları ve bilgileri (öğeleri) sunmayı amaçlamaktadır.

Tavsiye sistemleri, belirli bir kullanıcı için faydalı olan öğelerin önerilerini sağlayan yazılım araçları ve teknikleri olarak tanımlanmaktadır (Ricci, Rokach and Shapira, 2011). Aynı zamanda aktif kullanıcıya anlamlı ve etkili ürün önerileri sağlamak amacıyla bilgi toplama ve filtreleme için yazılım araçları ve teknikleri olarak ifade edilmektedir (Gatzioura and Sànchez-Marrè, 2014; Patel, Desai and Panchal (2017)). Bu tanımlardan, tavsiye sistemlerinin bir dizi yazılım aracı olduğu ve aynı zamanda formül, algoritmalar, problem formülasyonu ve tavsiye sistemleri probleminin çözümü ile ilgili yöntemlerin çözülmesiyle alakalı teknikler olduğu anlaşılabilir. Önemli olan, sistem tarafından tahmin edilenin, kullanıcıyla ilgili olabilecek şeyleri tavsiye edebilmesidir.

Bu sistemlerde kullanılan teknikler 1990'lardan beri kullanılmaktadır. O zamandan beri, tavsiye sistemlerinin, başta e-ticaret, sosyal medya siteleri, müzik ve televizyon yayını, haber ve bilgi paylaşımı uygulamaları olmak üzere e-öğrenme ve akademik yayın platformları, çevrimiçi seyahat planlayıcıları gibi birçok alanda uygulamaları bulunmaktadır.

E-ticaret sitelerinde (örneğin, amazon, hepsiburada) tavsiye sistemleri, kullanıcıya satın almakla ilgilenebileceği şeyleri önermekle sorumludur. Film ve müzik akış sitelerinde (ör. Last.fm, Netflix), tavsiye sistemleri, izleyicilere hangi filmlerin izleneceğini veya hangi

müziğin dinleneceğini önerme konusunda büyük rol oynamaktadır. Arama motorlarında (örneğin Google), tavsiye sistemleri alakalı olabilecek sayfaları seçer ve kullanıcıya bu seçilenleri yine algılanan önem sırasına göre kullanıcıya sunar. Gezi planlama veya tatil sitelerinde, tavsiye sistemleri, kullanıcının ilginç bulabileceği seyahat turlarını önermeye yardımcı olur. Öte yandan tanışma sitelerinde, bir kullanıcıya potansiyel eşleşmeler önerir. Bir diğer aktif kullanım alanı ise, kullanıcının takip edebileceği arkadaş ve sayfaları önermek amacıyla kullanıldığı sosyal medya siteleridir (örneğin, Instagram, Facebook).

Tavsiye sisteminin amacı iki perspektiften gözlemlenebilir. Birincisi, kullanıcının sistemi maksimum düzeyde kullanmasını sağlamak, ikincisi ise işletme sistemleri bağlamında, kârı maksimize etmektir. Tavsiye sistemlerinin e-ticaret sitelerine sağladığı avantajlar, tarayıcıları bir nevi alıcılara dönüştürmeyi, kullanıcıların amaçlanandan daha fazla satın almalarını sağlayan çapraz satışları ve ayrıca kullanıcı deneyimlerini kişiselleştirerek müşteri sadakati sağlamayı içermesidir (Schafer, Konstan and Riedl, 1999).

Kullanıcıyla ilgili öneride bulunarak kullanıcı memnuniyetini sağlamak bu sistemin amaçları arasındadır. Ayrıca kullanıcı, kendisine gerekli olanı elde etmek için lazım olmayan çok fazla bilgi içerisinde dolaşmanın zamanından ve enerjisinden kurtarılır.

Tavsiye sistemleri, oluşturulan çok miktarda veriden faydalanır. Bu veriler, kullanıcı tercihlerimizi, internet üzerinde harcanan zamanı ve web platformları kullanma etkinliklerimizi vb. şeyleri içerir. Tavsiye sistemleri, kesinlik kalitesini yükseltmek için, zamanla farklı veri türlerini ve kullanıcı verisindeki bolluğu kullanarak gelişmiştir.

Amaç

Bu çalışmanın amacı, bir tavsiye sisteminin ne olduğu ve nasıl çalıştığı hakkında bir bilgi tabanı sağlamaktır. Tavsiye veren kurumlar arasındaki yapısal ilişkiyi araştıran bir tavsiye sistemi modeli önermekte ve daha ileri boyutta fikir vermektedir. İlk olarak, literatürdeki tavsiye sistemlerinin farklı yöntemleri incelenmektedir. Bu, tavsiye sistemlerinin ne olduğunu ve kullanılan algoritmaların niteliğinin doğruluğunu bir şekilde anlamak için önemlidir. İkinci olarak, yapısal yaklaşımla diğer gömülü tekniklerden farklı olan alternatif bir veri gösteriminin kullanımı incelenmektedir. Bu gösterim kullanılarak, sinirsel işbirlikçi filtreleme yöntemini barındıran bir tavsiye sistemi modeli uygulanmaktadır.

Son olarak, bu tavsiye sistemi uygulama biçiminin sonuçları analiz edilerek faydaları vurgulanmaktadır.

Katkı

Bu çalışma;

- Tavsiye sistemlerinin çeşitli tiplerinin gelişiminin bir özetini sunmaktadır.
- Grafsel veri gömme ve sinirsel işbirliği filtrelemesi kullanarak bir tavsiye sistemi modeli oluşturmaktadır.
- Diğer yöntemlere göre modelin avantajları sunulmaktadır.

Tezin Akışı

Tezin akışı incelendiğinde süreç şu şekilde ilerlemektedir. İlk olarak aşağıdaki bölümlerde, tavsiye sistemlerini anlamaya yönelik olan kavram ve anahtar terimlerin bir listesi açıklanmaktadır. Ayrıca, tavsiye sistemlerinin geleneksel kavramlar üzerine gelişimini anlamak için iç görü sağlanmaktadır. Sonra, tavsiye sistemleri alanında yapılan çalışmaları anlamaya yönelik kaynaklar incelenmektedir. Bu kaynaklar, geleneksel yöntemlerden, günümüze kadar kullanılan veri setlerinde yer alan farklı multimedyaları da içeren mevcut tekniklere kadar kapsamlı çalışmaları sunmaktadır. Ayrıca, node2vec algoritmalarını içeren tavsiye sistemleri alanında yapılmış çalışmalardan da bahsedilmektedir.

İlerleyen kısımlarda, node2vec algoritmasının ne olduğu ve tavsiye sistemleri için nasıl kullanılabileceğinin bir açıklaması yapılmıştır. Node2vec bir graf düğümü gömme tekniği olduğundan, tavsiye sistemi sorununun node2vec algoritması için uygun olan uyarlanabilir bir graf gösterime nasıl dönüştürüleceği incelenmektedir. Bir sonraki bölüm algoritmanın ve uygulamanın nasıl yapıldığına odaklanan yöntem bilgisini sunmaktadır. Kullanılan veri setlerinin yapıları ve veriler üzerinde gerçekleştirilen ön işleme adımları bu bölüm içerisinde anlatılmaktadır.

Son olarak, farklı değerlendirme ölçütleri kullanan algoritmalarımızın davranış ve performans yönleri ile literatürdeki en son algoritmalarla nasıl bir araya geldiklerine dair bir analiz yapılmaktadır.

KAYNAK ÖZETLERİ

Tezin özetini daha iyi anlamak için, özellikle tavsiye sistemleri ve node2vec ile yapılan işlere ek olarak, geri bildirimde bulunmak ve tavsiye sistemleri alanında yapılan çalışmaları incelemek önemlidir. Öncelikle geçmişte kullanılan farklı geleneksel teknikler araştırılmıştır. Tavsiye sistemleri alanında farklı veri türleri ve sektörlerin kullanımı ile ilgili son yıllardaki literatür kaynaklarına ve alandaki diğer çalışmalara bakılarak nasıl gelişmeler olduğu anlaşılmaya çalışılmıştır.

Terim Tanımı

Öğe (Item): Kullanıcının aradığı, ilgisini çeken veya kullanıcının profili ile ilgili bilgi ya da ürünü ifade eder. Örneğin, kullanıcının satın alabileceği ürün, aradığı bilgi ya da kaynakça, sevdiği müzik veya dizi vb.

Kullanıcı (User): Siteyi kullanan veya bilgi arayan kişi veya aracıdır.

Geri Bildirim (Feedback): Gerçekleştirilen bir eyleme karşılık olarak verilen veya toplanan bir reaksiyon veya bilgi formudur. Bu bağlamda, geribildirim, kullanıcının memnuniyet düzeyini belirten bir sistemde gerçekleştirdiği bir tepki veya bir eylemdir**.**

İmalı Geribildirim (Implicit Feedback): Kullanıcıların sistemle olan etkileşiminden düşülen bir geri bildirim şeklidir. Kullanıcının bir öğeyi görüntüleme sayısı, söz konusu öğeyi görüntüleme için harcanan süre, tıklama sayısı vb. gibi birkaç form alabilir.

Belirgin Geribildirim (Explicit Feedback): Kullanıcı tarafından belirli bir öğe için özel olarak verildiğinde geri bildirim açıklanmaktadır. Belirgin geribildirimin en yaygın şekli kullanıcı tarafından değerlendirilmesidir.

Kullanıcı Değerlendirmesi (User Rating): Derecelendirme, bir kullanıcının belirli bir öğeyi ne kadar sevip sevmediğinin bir ölçüsüdür. Genellikle 1-5 veya 1-10 arasında bir aralıkta ölçülür.

Fayda Matrisi (Utility Matrix): Kullanıcı-öğe matrisi olarak da adlandırılan fayda matrisi, özellikle işbirlikçi filtreleme yaklaşımı olan tavsiye sistemlerinin çok önemli bir bileşenidir. Kullanıcı-öğe matrisi (Şekil 2.1.), kullanıcılar ve öğeler arasındaki etkileşimleri haritalayan bir matristir. u (kullanıcı sayısı) ve v (öğe sayısı) boyutunda bir M matrisi göz

önüne alındığında, matrisin $M_{i,j}$ numaralı elemanı, i numaralı kullanıcı tarafından j öğesi hakkında verilen geri bildirim sayısını belirtir.

	Öğe 1	**Öğe 2**	**Öğe 3**	**Öğe 4**	**Öğe 5**	**Öğe 5**
Kullanıcı 1	4					2
Kullanıcı 2		5			1	
Kullanıcı 3			3			
Kullanıcı 4		3				
Kullanıcı 5				1	2	

Şekil 1. Kullanıcı-öğe matrisi

Komşuluk: Tavsiye sistemindeki komşuluk, belirli bir kullanıcı veya bir öğeyle benzerlik içinde olan (atanmış bir ölçü ile) yakın olan öğeler veya kullanıcılar anlamına gelir.

İşbirlikçi Filtreleme (Collaborative Filtering): Kullanıcının geçmiş tercihlerine odaklanan bir tavsiye sistemi yöntemidir. En popüler modellerden birisidir.

Benzerlik Ölçüsü (Similarity Measure): Bir benzerlik ölçüsü genellikle bir öğenin veya kullanıcının bir başkasına ne kadar yakın olduğunu belirlemek için kullanılır. Benzerlik ölçüleri genellikle vektör uzaklık ölçüleri şeklini alır. Kullanılan ölçülerden bazıları Pearson korelasyon katsayısı ve kosinüs benzerlik (cosine) ölçüleridir.

İçerik Esaslı Filtreleme (Content Based Filtering): İçerik esaslı teknikler, alana bağlı algoritmalardır ve tahmin üretmek için öğelerin özellikleri üzerinde daha fazla dururlar. Bu teknik, zengin bir öğe özellik tanımının kullanılabilirliğini gerektirir, çünkü teknikler, bir kullanıcının beğendiği öğelerin kalitesini veya doğasını tanımlamak için bu özelliklerin değerine bağlıdır.

Hibrit Tavsiye Sistemleri: Hibrit tavsiye sistemleri, daha iyi öneriler sunmak için farklı tavsiye sistem yöntemleri ile birlikte kullanılan modellerdir.

Bağlam Bilinci (Context Awareness): Bağlam bilinci tavsiye sistemleri, kullanıcıya önerilerde bulunurken bağlam fikrini dikkate almaktadır. Bir işletmenin bağlamı; yer, tercihler, sağlık, zaman, mevsimler, akıl durumu gibi şeyler olabilir.

Yorumlar (Reviews): Belirli bir maddeye ilişkin duygularını ve görüşlerini ifade eden, kullanıcı tarafından yazılmış metinlerdir.

Etiketler (Tags): Kullanıcı tarafından atanan bir öğeyi tanımlayan başlıktır. Bu, öğe niteliklerle karıştırılmamalıdır. Doğru gruplama veya ilişkilendirme için kullanıcılara veya

öğelere atanan tanımlayıcılardır. Etiketler, öğelerin etiketlenmesinde de kullanılır. Ayrıca tavsiye yapmak için de faydalıdır.

Gizli Özellikler (Latent Features): Gizli özellikler verilerin sayısal vektör gösterimleridir. Bir varlığı tanımlayabilen gizli özellikler içerdiği bilinmektedir.

Soğuk Başlangıç (Cold Start): Soğuk başlangıç sorunu, tavsiye sistemlerinin karşılaştığı bir sorundur. Yeni kullanıcılara önerilerde bulunma veya yeni ürünler önerme sorunudur. Bir ürün, sistemde derecelendirilmemişse tavsiye için dikkate alınmaz. Aynısı, sistemin tercihlerini belirlemesini sağlamak için herhangi bir öğe hakkında herhangi bir geri bildirim vermemiş olan yeni bir kullanıcı için de geçerlidir. Aynı zamanda bu, işbirlikçi filtreleme tekniklerinde de yaşanan bir problemdir.

Rektifiyeli Doğrusal Birim Aktivasyon Fonksiyonu (Rectified Linear Unit, (ReLU)): Yapay sinir ağlarında kullanılan bir aktivasyon işlevidir. ReLU, 0'dan büyükse giriş değerini döndürür. Aksi halde, giriş 0 ise veya 0'dan küçükse 0 döndürür.

Regularization Rate: Bir sinir ağı parametresidir. Eğitim setine aşırı bağlılığı (overfitting) önlemeyi sağlamak için modelin karmaşıklığını yönetmeye yardımcı olur.

Yerk Oranı (Dropout Rate): Başka bir sinir ağı parametresidir. Yerk oranı, modelin aşırı bağlılığını (overfitting) önlemeyi sağlamak için sinir ağ eğitimi sırasında rastgele bırakılan düğüm sayısını yönetir.

Veri Gömme (Embedding): Ayrık/kategorik bir değişkenin hesaplamalarda kullanılabilecek sürekli sayı vektörüyle eşleştirilmesi olayıdır.

Genel Tavsiye Sistemleri

Tavsiye sistemlerinin, kaynaklara göre 1990'lara kadar geçmişi bulunmaktadır. Tavsiye sistemleri için kullanılan teknikler literatürde bilgi toplama ve filtreleme başlığı altında bulunabilir (Resnick, Iacovou, Suchak, Bergstrom and Riedl, 1994). Shah, Salunke, Dongare and Antala (2017), çok basit terimlerle tavsiye veren sistemlerin farklı geleneksel tekniklerini ilgili çalışmada özetlemektedir.

İşbirlikçi filtreleme, tavsiye sistemleri oluşturmak için en eski ve en popüler yaklaşımlardan biridir. Birincil bilgi kaynağı kullanıcı verileri olan alandan bağımsız bir yaklaşım olarak tanımlanmaktadır. Böyle bir modelde, tavsiye sistemleri, kullanıcıların önceki alımları veya davranışları ile benzer kullanıcıların davranışları hakkındaki bilgileri analiz eder. Bu yöntem sayesinde, kullanıcıya, beğendiği öğelere benzer öğeler önerilmektedir.

İşbirlikçi filtrelemenin merkezinde fayda / kullanıcı-öğe matrisleri bulunur. Bellek esaslı (Memory-based) ve Model-temelli (model based) işbirlikçi filtreleme olmak üzere iki biçimi vardır. Komşuluk ya da sezgisel tabanlı yöntemler olarak da bilinen bellek tabanlı işbirlikçi filtreleme, kullanıcıların işbirlikçi filtreleme için çok önemli olan öğelerin derecelendirmelerini kaydeder. Sistem, benzer kullanıcıların bir komşuluğunu tespit eder ve bir kullanıcı-öğe matrisi oluşturur. Belirli bir kullanıcı önerisi vermek için, 2 farklı yönü olabilir, bunlar kullanıcı odaklı ve öğe odaklı olarak sıralanabilir. Komşuluk ya da sezgisel tabanlı yöntemler olarak da bilinen bellek tabanlı işbirlikçi filtreleme, kullanıcıların işbirlikçi filtreleme için çok önemli olan öğelerin derecelendirmelerini kaydeder. Sistem, kullanıcı-öğe matrisi ile oluşturulan benzer kullanıcıların bulunduğu bir komşuluk haritasını çıkarır. Belirli bir kullanıcının tavsiye alması adına, bellek esaslı işbirlikçi filtreleme için 2 farklı yaklaşım esastır. Bunlar kullanıcı odaklı ve öğe odaklı yaklaşımlardır. Kullanıcı odaklı yöntemde, sistem, bir kullanıcının bir öğesine olan ilgisini değerlendirir. Bu öğenin puanlarını, kullanıcının komşuluğunda bulunan diğer kullanıcıları dikkate alarak hesaplar. Kullanıcının komşuluğu benzer puan alan öğelere sahip diğer kullanıcıları da içerir (Shah, Salunke, Dongare and Antala. 2017).

Öğe odaklı yöntemlerde, sistem, bir kullanıcının i. öğesinin derecelendirmesini, i'ye benzer öğeler için u derecelendirmelerine dayanarak tahmin eder. Öğeye dayalı yaklaşımlarda, iki madde arasındaki benzerlik, sistemin diğer kullanıcıları tarafından benzer şekilde verilen derecelendirmelerle hesaplanmaktadır. Literatürde kullanılan en belirgin benzerlik ölçütlerinden bazıları Pearson korelasyon katsayısı ve Kosinüs benzerlik ölçüsüdür. Kosinüs benzerliği, Pearson'a dayalı ölçekten farklıdır, çünkü istatistiksel yaklaşımdan ziyade lineer cebire dayanan bir vektör uzayı modelidir. İki n-boyutlu vektör arasındaki benzerliği aralarındaki açıya göre ölçer. Kosinüsüne dayalı ölçüm, iki metin belgesini karşılaştırmak için bilgi almada ve metin madenciliği alanlarında yaygın olarak kullanılır. Bu durumda belgeler terimler vektörü olarak gösterilir (Isinkaye, Folajimi and Ojokoh, 2015). Model-temelli yaklaşım, bellek-temelli yaklaşımdan farklıdır. Çünkü sistem tarafından depolanan kullanıcı-öğe derecelendirmelerini hesaplamak yerine, bu verileri kullanır ve veriler üzerinde ilginç modelleri keşfetmeye ve tahmin modellerini öğrenmeye çalışır. Bu öğrenme sürecinde kullanılan tekniklerden bazıları şu şekilde sıralanabilir. Matris ayrıştırma teknikleri, Tekil değer ayrıştırma (SVD), Matris tamamlama teknikleri, Gizli anlam yöntemleri, Regresyon, Kümeleme teknikleri, Birliktelik kuralı çıkarımı, Destek vektör makinesi (SVM), Karar ağacı algoritmaları, Yapay Sinir Ağı (Derin öğrenme ağları) ve Bayesian sınıflandırma modeli. İşbirlikçi filtreleme tekniklerinin kullanıcı etkileşimi ve geri bildirime daha fazla odaklanması, bu tekniğin dezavantajlarından bir tanesini oluşturmaktadır. Çünkü kullanıcının

tercihlerinin geçmişi olmadığında, sistem bir tavsiye veremez. Tavsiye sisteminin diğer bir teknik kategorisi de içerik esaslı olmasıdır.

Bilişsel teknikler olarak da bilinen içeriğe dayalı teknikler, temel olarak önerilen öğelerden elde edilen bilgilere dayanır. İçeriğe dayalı teknikler, etki alanına bağlı bir algoritmadır ve tahmin üretmek için öğelerin nitelikleri üzerinde daha çok durmaktadır (Isinkaye, Folajimi and Ojokoh, 2015). Bu teknik, çok sayıda öğe özellik tanımının kullanılmasını gerektirir. Çünkü teknikler, bir kullanıcının beğendiği öğelerin kalitesini veya doğasını tanımlarken bu özelliklerin değerine bağlı kalır. Daha sonra algoritma bu öğelerin özelliklerini diğerleriyle karşılaştırarak ilerler ve kullanıcıya benzer nitelikteki öğeleri önerir.

İçerik esaslı filtreleme, genellikle öğenin değerli tanımlayıcılarını toplamanın mümkün olduğu durumlarda kullanılır. Bu tür durumlar arasında, önerilen öğenin yayınlar, haberler ve web sayfası gibi belgeler olduğu sistemler yer almaktadır. İçerik tabanlı tavsiye sistemlerinde kullanılan tekniklerden bazıları arasında Terim Frekansı Ters Belge Frekansı (TF / IDF) ve Yapay Sinir Ağı modelleri yer almaktadır.

Daha önce tartışılan iki teknik de, tavsiye sistemlerinin kullanıldığı her türlü sistemde %100 verimlilik sağlayan, hepsine uyabilecek çözümleri içermez. Hibrit tavsiye sistemleri, daha iyi sonuçlar elde etmek için farklı yaklaşım türlerinin güçlerini birleştiren tekniklerdir. Hibrit teknikleri geliştirmek için farklı teknikleri birleştirmenin çeşitli yolları vardır (Shah *et al*. 2017).

Bir diğer tavsiye sistemi kategorisi, Bağlama-duyarlı tavsiye sistemleridir. Bağlama-duyarlı sistemler, tavsiye sistemlerinin oluşturulmasında çok gerekli bir yöntemidir. Bağlam, Cambridge sözlüğü tarafından "bir şeyin var olduğu durum ya da meydana gelme ve bunu açıklamaya yardımcı olabilecek durum" olarak tanımlanır. Bağlama- duyarlı tavsiye sistemleri teknik açıdan, kullanıcıya tavsiyelerde bulunmanın bağlam düşüncesini dikkate almaktadır. "Bağlam, bilgisayar bilimi (öncelikle yapay zekâ ve birçok yerdeki bilgi işlemde), bilişsel bilim, dil bilimi, felsefe, psikoloji ve örgütsel bilimler de dahil olmak üzere farklı araştırma disiplinlerinde incelenen çok yönlü bir kavramdır", (Ricci, Rokach and Shapira, pp.219, 2011). Herhangi bir bağlam (context) göz önünde bulundurulmaksızın, verilen bir tavsiye kullanıcıyla ilgisiz olabilir. Bir işletmenin bağlamı (context), yer, tercihler, zaman, mevsimler, aklın durumu gibi şeyler olabilir. Ya da bir işletmenin bağlamı; konum, tercihler, zaman, mevsimler, aklın durumu gibi şeyler olabilir. Örneğin, bir kullanıcının ilgisi belirli bir zaman diliminde farklılık gösterebilir; bir tatil seyahat paketi web sitesi mevsime bağlı olarak farklı önerilerde bulunabilir. Tavsiye sistemde bağlamı (context) yakalamanın üç tane

belirtilen yöntemi vardır, bunlar Bağlam ön filtreleme, Bağlam farkında filtreleme ve Bağlam modellemedir. (Ashley-Dejo, Ngwira and Zuva, 2015).

Bazen önerilen öğenin niteliğine bağlı olarak, yukarıda tartışılan geleneksel teknikleri kullanmak mümkün değildir, bu nedenle yararlı olabilecek teknikler geliştirilmiştir. Tavsiye sistem teknikleri, uygulama alanına ve öğe türüne göre değişiklik göstermektedir. Otomobil, emlak, finansal hizmetler gibi uygulamalarında daha belirleyici ve bilgili bir yaklaşım gerektiren karmaşık ürünler söz konusu olduğunda, farklı tavsiye teknikleri gerekmektedir. Bilgiye dayalı (knowledge-based) ve kısıtlamaya dayalı (constraint-based) tavsiye sistemleri bu kategoriye girmektedir. Bilgiye dayalı ve kısıtlamaya dayalı bir tavsiye sisteminin kullanıcıya değerli bir öneride bulunabilmesi belirli bir etki alanına sahip olmasına bağlıdır (Kumar and Sharma, 2016). Bilgiye dayalı tavsiye sistemleri, belirgin kullanıcı gereksinimlerini ve önerilerin hesaplanması için temel ürün alanı hakkındaki derin bilgileri kullanır (Ricci *et al*. S 161, 2011). Bilgi tabanlı sistemler, anlamlı bir öneride bulunabilmek için kullanıcı gereksinimlerinin ve tercihlerinin açıkça belirtilmesini gerektirir.

Literatür Taraması

Bu başlık altında tavsiye sistemlerinin gelişimi daha detaylı bir şekilde incelenmeye devam edilecektir. Tavsiye alanlarının gelişimini doğru bir şekilde gözlemlemek için farklı alanlardaki çeşitli yaklaşımların literatür bilgisinde daha derinlemesine bir analiz yapılmasında fayda vardır. Tavsiye sistemleri, teknolojinin büyümesiyle birlikte gelişmiştir. Sayısal derecelendirmelerin kullanılmasından ve basit eksik değer yaklaşımlarının gerçekleştirilmesine kadar kullanılan süreçler bu gelişime önemli katkıda bulunmuştur. Ayrıca matrislerin ayrıştırılması, kullanıcı bağlamı ve metinler gibi diğer veri öğelerinin daha karmaşık modellere ayrıştırılması ve bu veri formlarında gömülü olan doğrusal olmayan kullanıcı/öğe ilişkilerinin görüntülenmesi ve öğrenilmesi bu gelişime katkı sunan diğer etkenlerdir. Geleneksel olarak, işbirlikçi filtreleme algoritmaları, öngörülen derecelendirmeyi hesaplamak için algoritma türüne bağlı olarak, kullanıcının veya öğenin komşuluk ağırlıklı ortalamalarını kullanır. Öğenin / kullanıcının komşuluğunu oluşturmak için vektör alanındaki benzerlik veya mesafe ölçüsünü kullanır. Benzerlik önlemlerinin iyileştirilmesinden kullanıcıların veya öğelerin bölgesini daha iyi belirleme yöntemlerine kadar geleneksel yaklaşımı iyileştirmenin yolları hakkında birkaç makale yazılmıştır. Li, Zhou, Xiong, Hu and Wei, (2017), kullanıcı benzerliğini hesaplamak için yeni bir yöntem önermektedir. Benzerliğin puanını artıran "Ortalama puan faktörü" ve benzerliğin kredibilitesini arttırmak için "benzerlik derecesi düzeltme faktörü" adlı 2 faktörü ortaya koymaktadırlar. Bu algoritmayı geleneksel kullanıcı tabanlı CF algoritmasıyla karşılaştırarak MAE ve RMSE

ölçümlerine dayanan umut verici sonuçlar üretmişlerdir. Aynısı soğuk başlangıç probleminde yapılan benzetilmiş bir deney için de geçerlidir. Bireysel özelliklerin benzerliğini bir araya getirme yaklaşımı, soğuk başlangıç problemini çözmede kullanılan yöntemlerden birisidir.

Bağlam (context), keşfedilen başka bir alandır. Tavsiye sistemi algoritmaları, bazen kullanıcı ve öğe özelliklerinin vektör temsiline dayanır. Ancak, bağlam oluşturmak için tüm özellikler yararlı değildir. Vargas-Govea, González-Serna and Ponce-Medellın, (2011) bağlamsal bilginin bir tavsiye sisteminin performansındaki etkisini incelemiştir. Metotları, tutarlılık seçici algoritması olarak bilinen öznitelik seçme yaklaşımını kullanır. Seçilmiş sayıda nitelik yerine tüm niteliklerin kullanılmasının, bağlamsız öneriler açısından performansta fark oluşturmadığını tespit etmişlerdir. Bununla birlikte, bağlam özel algoritma yöntemlerine gelince, sadece belirli öznitelikler kullanıldığında performansın daha iyi olduğunu gözlemlemişlerdir. Öznitelik seçimi, bağlam temelli modellerde göz önünde bulundurulmalıdır. Bağlamı özel olarak inceleyen bir diğer makale ise He and Wu'dur, (2009). Bir kişinin tercihlerinin zamanla değiştiğini ve zaman içinde komşusunun tercihlerindeki değişiminin bir sonucu olarak da değişebileceğini savunmaktadırlar. Makalelerinde, seyreklik sorunuyla başa çıkmanın yanı sıra performansı iyileştirmek için bir zaman bağlamı içeren geleneksel işbirlikçi filtreleme algoritmasını da kapsayan bir hibrit model önerilmiştir. Kullandıkları yöntem, belirli bir kullanıcının komşularının belirli bir zaman aralığında eşit aralıklarla ilgilerinin incelenmesini önermiştir. Böylece, incelenecek bir komşuluk kümesi üretilir. Belirli bir öğe için öngörülen değer daha sonra her bir komşuluk kümesinin en ilginç öğelerine dayanarak her komşuluk aralığı için hesaplanır. Sonuç, belirli bir öğe için farklı zaman aralıklarında bir tahmin vektörüdür. Her bir tahminin nihai tahmin değerini belirlemedeki önemi, tahmin değerinin ağırlığı hesaplanarak belirlenir. Nihai tahmin, komşuluk tahmininin toplamının, atanan ağırlıklarıyla, ağırlıkların toplamına bölünmesiyle elde edilen değerin çarpımı olarak hesaplanır. Deney, MovieLens 100k derecelendirme veri kümesi kullanılarak gerçekleştirilir. Hatırlama ve MAE ölçütlerini kullanarak yapılan değerlendirmelerin ardından önerilen yaklaşım geleneksel CF referans çizgisinden daha iyi bir performans sergilemektedir. Komşuluğunun ölçeği arttıkça MAE'nin önemli ölçüde düştüğünü fark etmek önemlidir. Tavsiye sisteminin bir diğer önemli gelişmesi, metin verilerini tavsiye sürecine dâhil etmek için sadece sayısal kullanıcı derecelendirmesini kullanmak üzere gelişmesidir.

Bazı sistemler kullanıcı deneyimlerini öğeleri sayısal olarak derecelendirmek yerine öğeleri etiketleme veya yorum verme seçeneklerine göre oluşturur. Araştırmacılar hem kullanıcı öğesi profillerini zenginleştirmek hem de tavsiye sistemine dahil etmek ve

performansı iyileştirmek için kullanıcı tercihleri hakkında çok şey öğrenmek için verilerin bu tür metinsel bileşenlerinin içeriğini de araştırmışlardır. Makalelerinde, Tso-Sutter, Marinho and Schmidt-Thieme (2008), tavsiye sistemi performansının arttırılmasında etiketlerin kullanımını daha ayrıntılı olarak araştırmışlardır. Tso-Sutter *et al.* etiketleri buna entegre ederek işbirlikçi filtreleme algoritmasının geliştirilmesini önermiştir. Etiket-kullanıcı ile etiket-öğe arasındaki korelasyon, fayda matrisini oluşturan kullanıcı-öğe ilişkisine ek olarak incelenmiştir.

Tavsiye sistemlerinde kullanılmış olan bir başka metinsel veri biçimi yorumlardır. Bir kullanıcının bir öğeye olan yararını, tercihlerini, sevdikleri veya sevmedikleri bir öğenin özelliklerini vb. ifade etmede çok değerli bir bilgi kaynağıdır. Önemli inceleme unsurlarının ve özelliklerinin çıkarılmasında farklı teknikler uygulanmıştır. Bunlardan Terim Frekansı / Ters Belge Frekansı (TF-IDF) (Garcia Esparza, O'Mahony and Smyth, 2011) terim temelli kullanıcı profilleri oluşturmak için kullanılmıştır. Yine, SVM ve CNN (Sharma, Tripathi, Sahu, Mittal and Anand, 2016) gibi makine öğrenme teknikleri ile doğrudan kullanıcı yorumlarından değerlendirmelerini tahmin etmek için yöntemler kullanılmışdır. Ayrıca, bağlamla ilgili gizli faktörleri modelleme işlemi, yorumların duygularını ve aynı zamanda yorumların yardımcılığını belirlemek için kullanılmıştır (Dong, Schaal, O'Mahony and Smyth, 2013). Bu çalışmada uygulanan yöntem, profil ve derecelendirmeyi tahmin eden iki bileşenli bir algoritma önermektedir. Algoritma, yorumlardaki konuları çıkarmak için Latent Dirichlet Allocation (LDA) uygulamasını kullanır. Kullanıcı ve restoran profili içeriğini kapsamaktadır. Örneğin, belirli bir kullanıcının i restoranına ait tüm ürünlere ait yorumlarında normalize edilmiş ortalama bir konu dağılımı olarak bilinen bir terimi türetir. Derecelendirme tahmin bileşeni, kullanıcı derecelendirme ve konu dağılımları arasındaki ilişkiyi yakalayan ve derecelendirmeyi öngören doğrusal ve lojistik modellerden oluşur. Zhang, Ding, Chen, Li and Zhang, (2013), kullanıcı yorumlarından elde edilen sanal derecelendirmeleri kullanarak çevrimiçi tavsiye sistemini geliştirmeyi amaçlayan 2 aşamalı bir algoritma önermiştir. Bu yöntem, kullanıcı derecelendirmelerinin verilmediği, ancak kullanıcının bir inceleme sağladığı sistemlerde yararlı olmaktadır. Sistemin ilk basamağı, kendinden denetimli, sözlüğe dayalı ve corpus tabanlı bir model (SELC modeli) kullanmaktadır. SELC modeli, kullanıcının yorumlarını SVM modeliyle güncellenen bir duyarlılık sözlüğü kullanarak sınıflandırır. Model ayrıca ifadelere (emojiye) dayalı sınıflandırmayı da içermektedir. Algoritmanın ikinci aşaması, önerilerde bulunmak için birinci aşamada üretilen etiketli verileri kullanır. Bu yöntemlerin tümü açık derecelendirmeler olmadan öneride bulunma imkânı sağlamaktadır. Özellikle yapay sinir ağları alanında, teknolojideki ve yapay zekâdaki gelişmelerle, geleneksel makine öğrenmesi algoritmalarının uygulandığı herhangi bir alana derin öğrenme

tekniklerinin uygulanması şaşırtıcı değildir. Tavsiye eden sistemler bu uygulama alanlarından istisna değildir. Derin öğrenme (DL) modelleri, geleneksel matris çarpanlarına ayırma ya da istatistiksel modeller kullanılarak yansıtılmayan ya da gözlemlenemeyen, kullanıcı ve ürün verileri arasında olabilecek karmaşık ve doğrusal olmayan ilişkileri öğrenme yeteneğine sahiptir. Zhang, Yao, Sun and Tay, (2019), tavsiye sisteminin geliştirilmesinde derin öğrenmenin kullanımını ve bunları uygulamada kullanılan farklı türde derin öğrenme modellerini geliştirmeyi araştıran en son araştırmaların detaylı bir incelemesini sunmaktadır. Bunlar arasında çok katmanlı algılayıcılar (MLP), Konvolüsyon Sinir Ağları (CNN), Otomatik Kodlayıcılar (Berg, Kipf and Welling, 2017) ve metinler gibi heterojen veri setlerinin farklı elemanlarını birleştirebilen hibrit modeller verilebilir (Zhang, Ai, Chen and Croft, 2017; Chu and Tsai, 2017; Xu, Lukasiewicz, Chen, Miao and Meng, 2017). Sharma, Tripathi, Sahu, Mittal and Anand (2016), bir doktor derecelendirme sitesindeki doktorlara verilen notu tahmin etmek için yorumlar kullanmıştır. Algoritma, kullanıcı yorumlarını sitedeki bir doktora verilen notu tahmin edebilmek için eğitmek üzere bir konvolüsyon sinir ağı kullanır. Model, her bir yorumdaki tüm kelimelerin vektör gösterimini öğrenmek için önceden eğitilmiş bir word2vec algoritması kullanır. Bu gösterim 5 katmanlı (yani çıktı katmanı hariç) bir CNN modeli için girdi olarak kullanılır. Model her incelemeye karşılık gelen bir derecelendirme değerini eşler. MAE ölçüsüne dayanarak değerlendirilen CNN modeli, yardımseverlik, personel ve bilgi özelliklerine dayanarak verilen puanları tahmin etmek için sırasıyla 0.389, 0.685 ve 0.501 puan değerlerini almıştır. Bu yaklaşım, bir kullanıcının derecelendirme sağlamadığı sanal derecelendirmeleri oluşturmak için kullanıcı yorumlarında bulunan bilgilerin kullanılması konusunda çok faydalı olabilir. Zheng, Noroozi and Yu (2017), Derin Kooperatif Yapay Sinir Ağları (DeepCoNN) adlı bir model önermektedir. Model, çıkış katmanına doğru yaklaşan iki paralel sinir ağı kümesinden oluşmaktadır. Birinci ağ kullanıcının gizli olan faktörlerini yakalar ve kullanıcının yorumlarını girdi olarak alır. Diğer ağ, öğenin gizli olan faktörlerini (hidden latent factors) yakalamak için kullanıcılar tarafından öğe hakkında yazılan yorumları kullanır. Bu iki ağ, paylaşılan bir katman kullanılarak ortak bir özellik alanında eşlenir, bu nedenle paralel ağların yakınsaması ortak bir çıktı üretebilir. Çıktı, verilen öğe için öngörülen derecelendirmedir. DeepCoNN modeli, ağa girdi işlevi gören sözcük vektörü temsillerini oluşturmak için sözcük gömme özelliğini kullanır. Sonuç, MSE metrik değerinde, Matris ayrıştırma (MF) gibi standart temel algoritmalara göre %8 civarında bir gelişme olduğunu göstermektedir. Öte yandan Catherine and Cohen, (2017), DeepCoNN algoritmasının, hedef kullanıcının hedef madde için yorum yazdığı örneklere karşı önyargılı olduğunu göstermektedir. Dönüşümsel Sinir Ağları (TransNet) olarak bilinen önerdikleri model, DeepCoNN üzerinde yapılan

çalışmanın bir uzantısı olarak tanımlanmaktadır. İki ağ mimarisi, Hedef ağ ve Kaynak ağdır. DeepCoNN'ye benzer şekilde, kaynak ağı, bir kullanıcı tarafından yazılan yorumları ve bir ürün hakkında yazılmış yorumları işler, ancak söz konusu kullanıcı tarafından belirtilen ürün için yazılan yorumları hariç tutar. Ayrıca, işlevi DeepCoNN'in paralel bileşeninin girişlerini kullanıcının hedef bir öğe için yazmış olduğu bir yorumun yaklaşık değerine eşlemek olan bir dönüşüm katmanı içerir. Hedef ağın işlevi, daha önce belirtildiği gibi dışlanan yorumları, eğitim sırasında kaynak ağın amaçlanan çıktısıyla karşılaştırılabilecek bir gösterime dönüştürmektir. Ağ daha sonra DeepCoNN modelinde olduğu gibi derecelendirmeyi tahmin etmek için kullanılır. TransNet, DeepCoNN ile karşılaştırıldığında gelişmiş sonuçlar vermese de, tavsiye sistemlerinin gerçek dünya uygulamaları açısından daha doğru sonuçlar vermektedir. Zhang, Ai, Chen and Croft, (2017), veri setini heterojen bütünlüğüyle araştıran bir model önermektedir. Ortak Temsil Öğrenme (JRL) olarak bilinen model, genel bir kullanıcı / öğe temsilini vermek üzere, tavsiye sistemleri için ortak olan 3 farklı veri formunu, yani incelemeleri, görüntüleri ve sayısal derecelendirmeleri ortak bir anlamsal gösterim alanına birleştirir. Sonra en üstteki N tavsiyesi nihayet çift taraflı öğrenme kullanılarak veri üzerinde gerçekleştirilir. Her bir yorum için, görüntü ve derecelendirme adına, veriler önce belirli bir kullanıcı ve öğe için ayrı bir gösterimde eşlenir.

Sırasıyla farklı görünümlerden kullanıcı ve öğe gösterimleri ayrı bir gizli (latent space) alana entegre edilir. JRL modeli, inceleme gösterimi için doc2vec (Le and Mikolov, 2014) olarak da bilinen PV-DBOW modelini kullanır. Görüntü gösterimi için bir derin sinir ağı mimarisi kullanılır. Derecelendirme gösterimi için, derecelendirme kullanıcısı ve derecelendirilmiş öğenin bir çift temsili, kullanıcı derecelendirmesini tahmin etmek için iki katmanlı birbirine tam olarak bağlı sinir ağında kullanılır. Bu sinir ağındaki kullanıcı ve öğe için varlık temsili, derecelendirme görünümü için sunum olarak kullanılır. Birleşme katmanındaki entegrasyon için, sırasıyla tüm kullanıcı ve öğe temsillerini bir araya getirmek üzere basit bir birleştirme işlevi kullanılır. Bu yaklaşım, çalışmalarında keşfedilen üç gösterimin ötesine geçme potansiyeline sahip olan heterojen veri kümesiyle çalışmanın bütünleşik bir yaklaşımını sunar. İncelemeden sonra, tavsiye veren sistemlerin farklı veri tipleri, algoritmalar ve tekniklerle geliştiği görülebilir. Ancak, bu tezin merkezinde önerilen yaklaşımlarla ve dayandığı tekniklerle doğrudan ilgili olan bir kaç tanesi ele alınmıştır. O çalışmalardan, Liao, Zhang, Nie, Hu and Chua (2017) kendisinin de kullandığı, sinirsel işbirlikçi filtreleme (NCF) adı verilen sinir ağının kullanılmasıyla işbirlikçi filtrelemeyi uygulamak üzere genelleştirilmiş bir çerçeve (framework) önermektedir. NCF çerçevesi temel olarak, işbirlikçi filtreleme etkisini modelleyen, kullanıcı-öğe etkileşimini yakalayan çok katmanlı bir sinir ağı modelidir. Alt giriş katmanları, kullanıcı ve öğenin özellik vektörlerini

çift (paralel) giriş modunda içerir. Bu katman, seyrek bir vektör girişini, girişin gizli özelliklerini temsil eden yoğun bir katmana eşleyen bir gömme katmanında beslenir. Bir sonraki katman, derecelendirme puanını çıktı olarak tahmin etmek için gizli faktörleri haritalayan çok katmanlı bir sinir ağı mimarisidir (yani sinirsel işbirlikçi filtreleme katmanları). He *et al*. (2017) model çerçevenin üç örneğini göstermiştir. NCF çerçevesi, önerilen algoritmanın temel aldığı çerçeveyi sağlar. Çerçeve, farklı veri temsillerinin onunla uygulanmasına izin verecek şekilde tasarlanmıştır. Literatürde kullanılan farklı veri gösterimi ve özellik öğrenme teknikleri vardır. Bunlar, fayda matrisi veri gösterimi (Li *et al*. 2017), gizli öznitelik vektörleri, bir sıcak kodlama (one-hot-encoding) (He *et al.* 2017) olarak sıralanabilir. Bu makale, bu çalışmanın dayandığı yöntem olduğu için çok önemlidir. Ayrıca, bir sinir ağı kullanarak işbirlikçi filtreleme sürecinin nasıl modelleneceğine dair kılavuz da sunar.

Farklı öznitelik gösterme yöntemlerine bakarak node2vec öznitelik öğrenme algoritması bulunmuştur (Grover and Leskovec, 2016). Nispeten yeni ve daha önce bahsedilen diğer öznitelik gösterme tekniklerinden farklıdır. Palumbo, Rizzo and Troncy, (2017) node2vec'i "rastgele yürüyüşlerle örneklenen düğüm dizileri üzerine word2vec modelinin uygulanmasıyla bir grafteki düğümlerin gösterimini öğrenen bir algoritma" olarak tanımlamaktadır. Node2vec algoritması, düğüm ve kenar kestirimi ve sosyal ağ gibi uygulama alanlarını içeren görevlerle karakterize edilen ağ analizi alanından esinlenmiştir. Grafsel bir temsil yapısındaki düğüm temsillerini öğrenmek için kullanılan algoritmik bir çerçevedir. Başka bir deyişle, node2vec, graf yapıları için gömülü bir nesil algoritmasıdır. Word2vec gömme işleminde kullanılan atlama-gram (skip-gram) modelini kullanır. Amaç, belirli bir hedef kelime için komşu kelimeleri tahmin etmektir. Node2vec durumunda korpus, verilen düğümün çevresindeki bir düğüm kümesiyle değiştirilir. Düğümün komşuluğu, görünüşte rastgele bir yürüyüş prosedüründe hem Genişlik İlk Arama (BFS) hem de Derinlik İlk Arama'yı (DFS) içeren bir örnekleme stratejisi kullanılarak üretilmektedir. Ağ köşeleri için başka bir temsili öğrenme algoritması olan DeepWalk (Perozzi, Al-Rfou and Skiena, 2014) algoritmasından farklıdır. Temel fark, DeepWalk rastgele seçilen düğümlerinin komşularını ziyaret ederken tamamen rastgele bir strateji kullanılırken, node2vec daha rehberli bir arayış içerisindedir. Node2vec arayışı, her adımın bir sonraki hareketini başka bir düğüme yönlendiren bir geçiş olasılığı üzerinden yönlendirilir. Node2vec'in, DeepWalk'a göre önemli bir gelişimi, graf yapıların fiziksel bütünlüğünü koruyabilmesidir. Bu yapısal bütünlükler gerçek dünya ağlarında bulunur ve gerçek dünya veri organizasyonlarının göstergeleridir.

Literatür incelendiğinde, tavsiye veri sistemleri için bu veri sunum tekniğinin kullanılması üzerine az miktarda çalışma yapılmıştır. Palumbo *et al.* (2017), vektörler arasındaki benzerlikleri ölçmek için varlıkların graf gösterimini kullanmıştır. Düğüm varlıkları ve ilişkileri, kullanıcı ve öğe varlıkları ile geri bildirim ilişkilerinin ötesine uzanır. İlk önce node2vec algoritması için girdi geliştirmek üzere bir bilgi grafiği oluşturulur. Varlıkların özellik gösterimlerini oluşturmak için node2vec algoritmasını kullanır. Kullanıcı ve her özellik alt grafiğinin öğeleri arasındaki benzerlikleri ölçülür. Ardından, bireysel özelliklere dayalı benzerlikler kullanılarak genel bir benzerlik ölçüsü hesaplanır. Daha düz bir tavsiye yaklaşımı olan node2vec'in kullanılmasına yönelik bir başka girişim, Palumbo *et al*. (2018) tarafından tarif edilmiştir. Sıralı öğeler için bir öneride bulunmak amacıyla, öznitelik gösterimi için node2vec kullanmışlardır. Graflerin düğümleri, kullanıcılar ve öğeler tarafından temsil edilir ve aralarındaki ilişkiler, mevcut geri bildirimlerin yanı sıra aralarında var olan bir takım öznitelikler tarafından kurulur. Kosinüs benzerlik ölçüsü, kullanıcı öğe çiftleri arasındaki ilişkiyi ölçmek için kullanılır. Wen, Guo, Chen and Ma (2018) ayrıca sosyal ağ alanında yerleşik bir danışma sistemi olan yerleştirme ağı önermiştir. Onların yaklaşımı, kullanıcıları daha sonra geleneksel bir matris ayrıştırma (MF) metoduna dahil edilen düşük boyutlu bir alana yerleştirmek için node2vec kullanılmasına dayanmaktadır.

Özetle, tavsiye sistemi araştırması yaklaşık 20 yıl önce ön plana çıkmış olsa da, sahadaki araştırmalar tüm veri formları ve tekniklerini içeren yıllar içinde büyümüştür ve gelişmiştir.

Tavsiye sistemleri, e-ticaret sitelerinden film ve müzik yayın sitelerine kadar birçok web platformunda kullanılmıştır. Tavsiye veren sistemler için kullanılan teknik, ortalamaların toplamından derin öğrenme gibi daha karmaşık algoritmalara kadar çeşitlilik gösterir. Farklı veri tiplerinin kullanımı çok önemlidir, çünkü tavsiye sistemlerini kullanan platformlar çok farklı ve çok yönlüdür. Kullanıcı mahremiyeti ve kullanıcı verilerinin sömürülmesi ihtiyacı hala çözülememiş önemli bir problemdir. İşbirlikçi filtreleme tekniklerinin halen soğuk başlangıç problemleri vardır. Yorumların, etiketlerin ve niteliklerin kullanımı, kullanıcı derecelendirme verilerinin azlığında birleştirilmek için kullanılır. Ayrıca, tavsiye sistemlerinin pratikte çoğu zaman tavsiye için tek bir strateji kullanmadığını da belirtmek yerinde olacaktır. Bazen kullanıcı ara yüzleri, bir web sitesindeki farklı tavsiye türlerini ayırmak için kullanılmaktadır.

Algoritmaların performansını ölçmek hala kolay değildir, çünkü performans yalnızca çevrimdışı olarak ölçülebilmektedir.

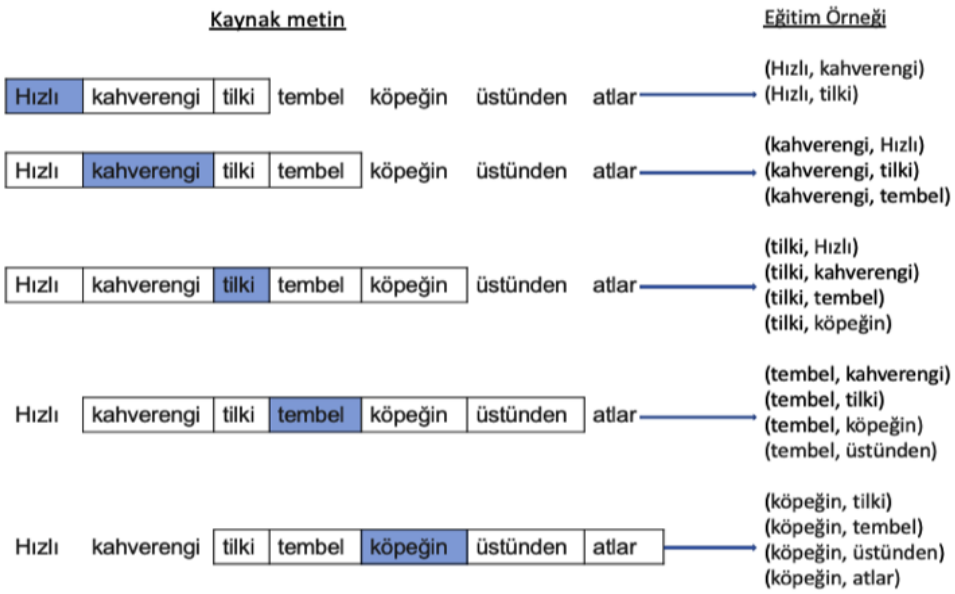
Bu nedenle, çevrimdışı algoritmanın gerçek zamanlı tavsiye sistemlerinde test edilmedikçe nasıl bir performans göstereceğini söylemenin bir yolu yoktur. Bu tür çevrimdışı / teorik deneyler, üretime girmeden önce neyin işe yarayıp neyin işe yaramadığını belirlemek için kullanılır.

MATERYAL ve YÖNTEM

Bu bölümde önerilen algoritmayı oluşturmak için kullanılan yöntem açıklanmıştır. Daha önce de belirtildiği gibi, tavsiye sistem modellemesine grafsel bir temsil yaklaşımı önerilmektedir. Node2vec algoritması olarak bilinen bir graf veri gömme yöntemini kullanarak, grafteki her düğümü temsil eden sayısal vektör temsilleri üretilmektedir. Kullanıcı puanını tahmin etmek için bu temsil formu bir sinir ağı çerçevesi ile birleştirilmektedir. Grafsel bir gösterim, tavsiye sistem varlıklarının fiziksel kompozisyonunu yakalar. Önerilen algoritmanın bileşenlerini daha iyi anlamak için, bu çalışmada kullanılan yaklaşımın arka planını oluşturan çerçeveler açıklanmıştır. Bunlar, node2vec algoritması gömme ve sinirsel işbirlikçi filtreleme çerçevesidir. Ayrıca, veri kümesinin node2vec algoritmasına giriş için nasıl işlendiği de bu bölümde açıklanmaktadır.

Skip gram Modeli

Gömme algoritmamızı daha iyi anlamak için, node2vec ve skip gram modellerini anlamamız gerekir. Skip gram modeli (Mikolov, Chen, Corrado ve Dean, 2013), doğal dil işlemede çok önemli olan word2vec gömme modellerinden biridir. Skip gram modeli, node2vec algoritmasının önemli bir bileşenidir. Esas olarak veriler özellik vektörleri oluşturmak için kullanılır. Skip gram modeli, tek bir gizli katmanı olan bir yapay sinir ağıdır ve tahmin görevini çözmek için kullanılır. Tahmin görevi, bir hedefin komşu (bağlam) kelimelerini tahmin etmektir. Her kelime için oluşturulan ağırlık matrisi, o kelime için özellik vektör gösterimidir. Metin bağlamında işleme, eğitim setinin sonucunda oluşan bir dizi cümle kümesidir (corpus). Eğitim örneğini oluşturmak için, bağlamın boyutu (komşu kelime) belirlenir. [Hedef-bağlam] kelime çiftleri, Şekil 3.2'de özetlenen cümle taranması yoluyla üretilir. Bir cümledeki her bir hedef sözcük için, hedef sözcükten önce ve sonra bir bağlam penceresi belirtilir. Daha sonra, hedef kelime ve bağlam penceresinde birbirinden farklı bir kelime kullanılarak bir eğitim örneği oluşturulur. Modelin uygun şekilde eğitilmesini sağlamak için, hedef kelimeler bağlam kelimeleri olarak rastgele oluşturulmuş / ilişkisiz kelimelerle eşleştirilir. Sağ bağlamdaki kelime çiftleri için 1, ilişkisiz bağlamdaki kelime çiftleri için ise 0 etiketi atanır. Bu, skip gram tahmin modeli için veri kümesi görevi görür.



Şekil 2. Skip gram modeli için oluşturulan veri örneği

Skip-gram modeli node2vec durumunda graf verilere uyarlanmıştır. Kelimeler ve belgeler yerine, node2vec bir dizi bağlı düğüm ve graf kullanılır. 3.2 başlığında korpus üretimi ve eğitim örnek verilerinin bir graften üretimi ayrıntılı olarak açıklanmıştır.

Node2vec Gömmesi

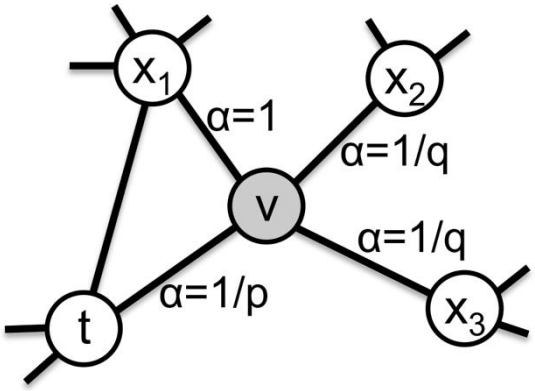
Literatürde daha önce tartışıldığı gibi, node2vec, graf verileri için bir vektör temsili oluşturmak adına kullanılan bir gömme algoritmasıdır. Doğrudan fayda matrisinden veya ayrışma yoluyla oluşturulan gizli bir öznitelik yerine, öğelerin ve kullanıcıların sosyal ağ tavsiyesinde yapıldığı gibi bir graf biçimindeki etkileşimlerine bakılır. Bir girdinin düğümler arasındaki bir kenarı gösteren bir düğüm çifti olduğu bir kenar listesi temsili verildiğinde, node2vec algoritması, her satırın beslendiği grafteki düğümler için bir özelliği temsil ettiği bir tür sürekli veri matrisi verir. Node2vec algoritması, daha önce açıklandığı gibi verilerde bulunan yapısal bütünlükleri yakalayabilir. Ayrıca vektörlerin boyutunu da kontrol edebilir. Node2vec gömme modelli graf düğümleri gömmek için word2vec'in skip-gram modelini kullanılır. Skip-gram, Şekil 3.4'de gösterildiği gibi bir eğitim örneğinin üretildiği bir korpus (bir yazılı metin koleksiyonu) gerektirir. Metin verileri üzerindeki bu korpus üretimi ile graf düğümleri için kullanımı arasındaki ilişki Şekil 3'te açıklanmıştır. Node2vec, düğümün komşusu boyunca yönlendirilmiş bir örnekleme stratejisi yoluyla korpusu oluşturmaktadır.

Node2vec gömme örnekleme stratejisi.

Node2vec'in örnekleme stratejisi, algoritmanın, eğitim kümesini oluşturmak için bir karar verisi oluşturmak amacıyla belirli bir düğümün komşusundan öğeleri seçmesidir. Node2vec'in örnekleme stratejisi 4 adet parametre kabul eder, bunlar:

- **Düğüm başına yürüyüş sayısı (N):** grafteki her düğümden oluşturulacak rastgele yürüyüşlerin sayısıdır.
- **Yürüme uzunluğu (L):** her rastgele yürüyüşte geçilen düğüm sayısıdır.
- **p:** önceden ziyaret edilmiş bir düğüme geri dönme olasılığını kontrol eden dönüş hiper parametresi olarak bilinir.
- **q:** başlangıç düğümünden uzaklaşma olasılığını kontrol eden Inout hiper parametresidir.

İlk iki hiper parametre oldukça açıklayıcıdır. Korpus üretimi için örnekleme stratejisi, **N** (düğüm başına yürüyüş sayısı) grafiğindeki her düğümden başlayacak ve **L** (yürüyüş uzunluğu) düğüm sayısı boyunca rastgele bir yürüyüşe geçecektir. **p** ve **q** değerleri, bir Şekil 3'e bakıldığında daha iyi açıklanmaktadır.



Şekil 3. Node2vec algoritmasında geçiş olasılığının nasıl hesaplanacağını gösteren diyagram (Grover and Leskovec 2016)

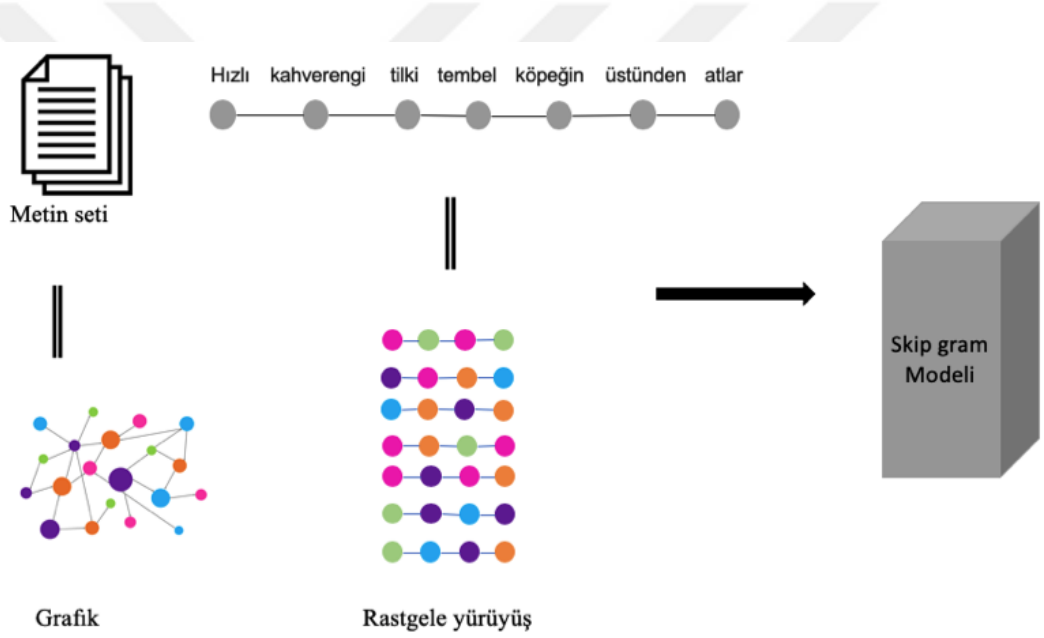
Rastgele yürüyüşte olduğunuzu düşünüp, yukarıdaki diyagramda t düğümünden v düğüme geçtiğinizi var sayalım. ***v***'den komşu ***x*** düğümlerinin herhangi birine geçiş olasılığı, kenar ağırlığı ($w_{v,x}$) × **α** (normalleştirme) dir, Burada **α**, hiper parametrelere bağlıdır. Graf ağırlıksızsa, $w_{v,x}$ = 1 olarak ayarlanır. **p**, ***v*** düğümünü ziyaret ettikten sonra ***t*** düğümüne geri dönme olasılığını kontrol eder. **q**, graflerin keşfedilmemiş kısımlarını keşfetme olasılığını kontrol eder. Mevcut olduğunda kenar ağırlığı da dikkate alınır, bu nedenle son hareket olasılığı fonksiyon tarafından verilir:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & if\ (v,x) \in E \\ 0, & aksi\ takdirde \end{cases} \quad \text{[MOU1]} \qquad (3.1)$$

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p}, & if\ d_{t,x} = 0 \\ 1, & if\ d_{t,x} = 1 \\ \frac{1}{q}, & if\ d_{t,x} = 2 \end{cases} \qquad (3.2)$$

$$\pi_{vx} = \alpha_{pq}(t,x) \times w_{v,x} \qquad (3.3)$$

Denklem 3.1, v düğümünden (mevcut düğüm) x düğümüne (sonraki düğüm) geçme olasılığını belirtir. $\boldsymbol{v}$ ve x arasında bir kenar varsa $\frac{\pi_{vx}}{Z}$ ile verilir, aksi takdirde 2 düğüm arasında kenar yoksa 0 olur. Z normalleştirilmiş bir sabittir. Denklem 3.2, 3 koşula bağlı olan $\alpha_{pq}(t,x)$ değerlerini gösterir. $d_{t,x}$ (yani, t'den x'e en kısa yol) 0'a eşitse, $\alpha_{pq}(t,x)$ $\frac{1}{p}$'ye eşittir. Eğer $d_{t,x}$ 1'e eşitse, $d_{t,x}$ $\frac{1}{p}$'ye eşittir. Son olarak $d_{t,x}$ 2'ye eşitse, $d_{t,x}$ $\frac{1}{q}$ değerindedir. π_{vx} değeri, $\alpha_{pq}(t,x)$ değerini kenar ağırlığı ile çarparak verilir (3.3). Burada $\boldsymbol{d_{t,x}}$, $\boldsymbol{t}$ düğümünden $\boldsymbol{x}$ düğümüne en kısa yolu gösterir ve $\boldsymbol{Z}$ normalleştirilmiş bir sabittir. Bu arama, grafiğin her bir düğümünden bir korpus üretir. Üretilen korpus, skip-gram için veri seti olarak kullanılmaktadır. Node2vec algoritması, maksimum olasılık işlevini en iyi duruma getirmek için skip-gram modelini genişletir. Skip-gram modelinin amacı, bir hedef düğüme en yakın düğümleri tahmin etmektir.



Şekil 4. Graf ve metinlerde Korpus üretimi

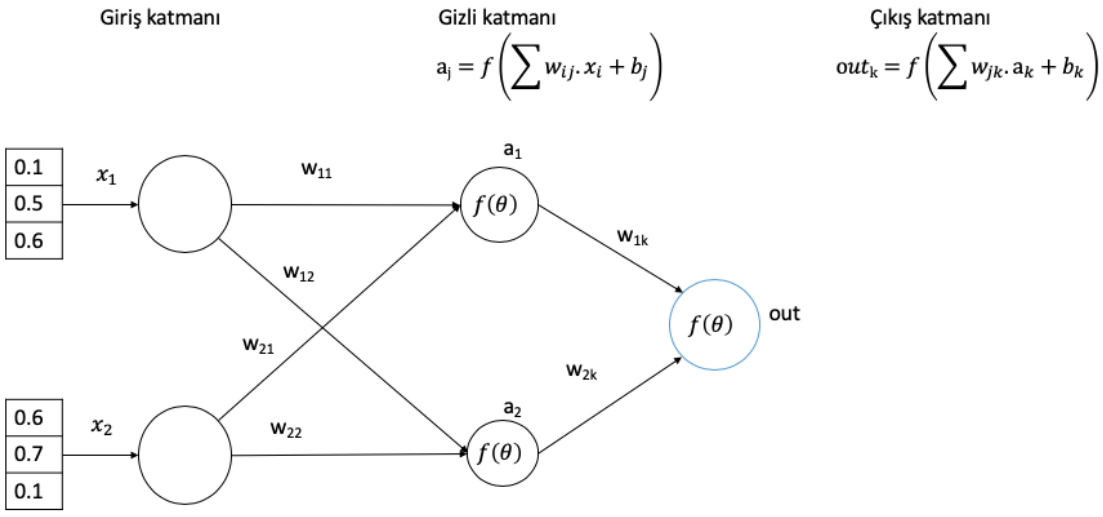
Sinirsel İşbirlikçi Filtreleme (NCF)

Geleneksel olarak, işbirlikçi filtreleme, SVD gibi matris ayrıştırma teknikleriyle gerçekleştirilir. Matris çarpanlara ayırma işlemi, kullanıcı u'nun, gizli özellik vektörlerinin bir iç ürünü ve bir optimizasyon algoritması yoluyla i öğesine verdiği derecelendirmeyi gerçekleştirir. Sinir ağı işbirlikçi filtreleme çerçevesi, tavsiye için derin öğrenme yöntemleri geliştirmek adına basit ve genel bir yönlendirme sağlar. NCF çok katmanlı bir sinir modelidir.

Yapay sinir ağları.

Yapay Sinir ağları, insan beynindeki sinir hücrelerinin etkileşimlerinden modellenmiş, verilerdeki örüntüleri tanıyabilecek şekilde tasarlanmış bir dizi matematiksel işlevdir.

Sinir ağlarını kullanarak kümeleme, sınıflandırma, tahmin vb. makine öğrenimi görevlerini yerine getirebiliriz. Şekil 3.4'ü kullanarak sinir ağlarının nasıl çalıştığı açıklanabilir. Matematiksel olarak, her bir sinir ağı, tamsayı değerler olan ağırlıklar yoluyla birbirine bağlanan nöronlardan oluşur. Her nöron, bir toplama ve aktivasyon fonksiyonunun bir kombinasyonudur. Şekilde görülebileceği gibi, nöron a1'i aktive etmek için nöronun girişleri (x1 ve x2) ilgili ağırlıkları olan w_{ij} ile çarpılır ve sonuçları bir yanlılık değeri ile birlikte eklenir ve bir aktivasyon fonksiyonuna $f(\theta)$ girdi olarak verilir. Aktivasyon fonksiyonu bir nöronun aktif olup olmadığını belirler. Devre dışı bırakılan bir nöron, çıktı olarak 0 üreten bir nöron iken, aktifleştirilmiş bir nöron sıfırdan farklı bir çıktı üretir. Her düğümün çıkışı, bağlı nöronuna ileri yönde bir giriş olarak kullanılır. Çıktı, probleme bağlı olarak ya çalıştığımız bir değerin olasılığını ya da gerçek değerin kendisini gösteren bir değere sahiptir. Belirli bir sorunu modellemek için sinir ağı, bir öznitelik veri kümesi ve beklenen bir çıkış değerleri ile beslenmelidir. Sonra model, verilerimizi sinir ağı üzerinden geçirerek, ağırlık ve önyargı değerlerini tekrar tekrar güncelleyerek eğitir. Her geçişte, beklenen çıktımızı bir kayıp fonksiyonu kullanarak üretilen çıktılarla karşılaştırırız. Kayıp fonksiyonu, tüm eğitim veri kümesinin genel hatasını ortalama olarak hesaplar. Beklenen çıktıya yeterince yakın tatmin edici değerler verebilecek ağırlıklara ulaşana kadar bu olay tekrarlanır. Eğitim süreci, backpropagation olarak bilinen bir işlemle belirli bir kayıp işlevini en aza indirmeyi veya en üst düzeye çıkarmayı amaçlayan bir optimizasyon işlemidir.



Şekil 5. Sinir ağının bileşenlerini gösteren bir diyagram

NCF'nin Çok Katmanlı Algılayıcı Örneği

Sinir ağlarının farklı tipleri ve mimarileri vardır. Modelimiz için kullanılan NCF örneği, çok katmanlı bir algılayıcı (MLP) sinir ağıdır. Çok katmanlı algılayıcı sinir ağı mimarisi, sinyali almak için bir giriş katmanına sahip bir tür ileri beslemeli sinir ağıdır. Girdi

hakkında bir karar veya tahmin yapan bir çıktı katmanı ve bu ikisi arasında MLP'nin gerçek hesaplama motoru olan keyfi sayıda gizli katman yer almaktadır. He *et al*., (2017) NCF çerçevesinin bir MLP örneğini önermektedir. Matematiksel olarak, NCF örneklemesi aşağıdaki formülle tanımlanır.

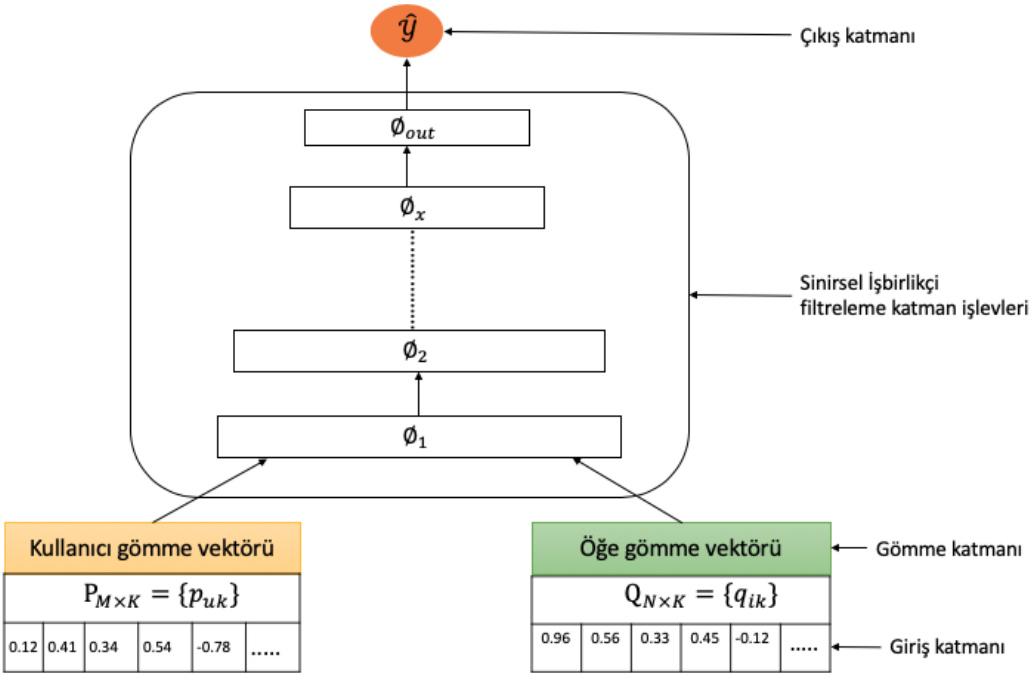
$$\hat{y}_{ui} = f(p_u^U, q_i^{\dot{I}} \mid P, Q, \Theta_f) \tag{3.4}$$

- $\boldsymbol{\Theta_f}$: Ortalama tahmini derecelendirme ile ortalama gerçek derecelendirme arasındaki kaybı en aza indirgemek için optimize edilmesi gereken çok katmanlı sinir ağının parametrelerini temsil eder.
- $\boldsymbol{P}$: $P \in R^{M \times K}$. M satır ve K sütunlu bir matrisidir. Burada her satır tek bir kullanıcının gizli vektörünü temsil eder.
- $\boldsymbol{Q}$: $Q \in R^{N \times K}$. Her satır tek bir öğenin gizli vektörünü temsil eden N satır ve K sütun matrisidir.
- $\boldsymbol{p_u^U}$: Kullanıcı gizli vektörlerini içeren P matrisinin bir satır vektörünü temsil eder.
- $\boldsymbol{q_i^{\dot{I}}}$: Kullanıcı gizli vektörlerini içeren Q matrisinin bir satır vektörünü temsil eder.
- $\boldsymbol{\hat{y}_{ui}}$: sinir ağının çıktısını temsil eder, bu durumda kullanıcı u'nun i öğesine verdiği değeri gösteren sayısal bir değerdir.

$\boldsymbol{P}$ ve $\boldsymbol{Q}$ matrisleri verildiğinde, sinir ağı model parametrelerini $\boldsymbol{\Theta_f}$, her bir kullanıcı öğesi vektörü p_u, q_i çifti için, modelin çıktısının beklenen öngörülen değerle farkı minimum olacak şekilde optimize eder. Bir optimizasyon algoritması kullanarak model parametreleri, gerçek derecelendirme ile öngörülen derecelendirme arasındaki hatayı en aza indirecek şekilde eğitilir. Aşağıdaki kayıp fonksiyonu, eğitim veri kümesinin tüm girişleri adına hatayı hesaplamak için kullanılır.

$$Kayıp = \sum_{(u,i) \in P \cup Q} w_{u,i}(y_{u,i} - \hat{y}_{u,i})^2 \tag{3.5}$$

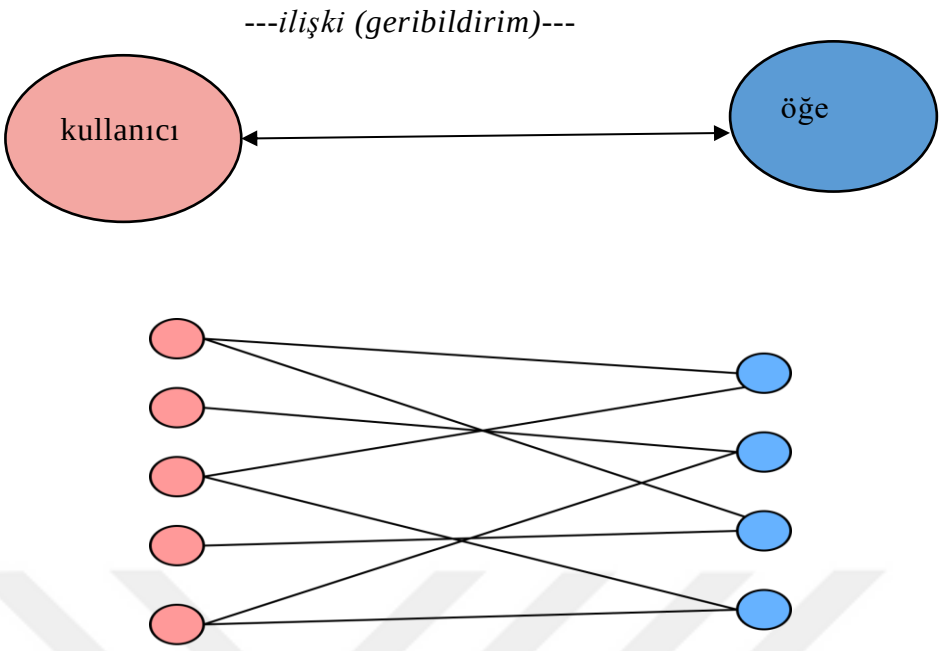
He *et al*. (2017) tarafından tanımlanan NCF modelinin, bir tür geri bildirim veri modeli kullandığına dikkat etmek önemlidir.



Şekil 6. Sinir işbirlikçi filtreleme çerçevesi (He *et al*. 2017)

Verilerin Grafsal Gösterimi

Veriler için graf gösterimi, tavsiye sistemi veri kümelerinin niteliklerini ve bunlar arasındaki ilişkileri bir grafın bileşenlerini yansıtacak şekilde yorumlamanın bir yoludur. Grafın düğümleri kullanıcı kimlikleri ve öğe kimlikleri ile temsil edilir. Bu durumda kenarlar, öğeler ve kullanıcılar arasındaki bir ilişkinin varlığını belirtir. Bu ilişki, tavsiye yöntemlerinde kullanılan iki taraflı bir graf oluşturacaktır (Berg, Kipf ve Welling, 2017). Veri kümesinin doğasına bağlı olarak farklı ilişki türleri çıkarılabilir. Ancak, tüm veri kümeleri için, kullanıcılar ve öğeler arasında uygulanabilecek veya çıkarılabilecek bir ilişki yani "geribildirim" bulunmaktadır. Literatürden anlaşıldığı üzere, geri bildirim örtük veya açık olabilir. Bu tez için, Şekil 3.6'da gösterildiği gibi iki düğüm arasında bir kenar oluşturmak için açık geri bildirim yani kullanıcı öğesi derecelendirmeleri kullanılmıştır.



Şekil 7. Graf gösterimi (veri kümesi dönüşümü)

Yöntem Özeti

Bu çalışmanın metodolojisinin temellerini oluşturan ana bileşenlerin kısa bir açıklamasından sonra, metodolojinin kapsamlı anlatımı üzerinde duralım:

- İlk olarak, veri kümesi grafsal bir formata dönüştürülür. Grafın düğümleri, verilerdeki varlıklar kullanılarak temsil edilir. Grafın düğümleri ana veri varlıklarını temsil eder. Bunlar sırasıyla kullanıcılar ve öğe varlıklarıdır. Varlıklar arasında bir bir ilişki kurmak için, kullanıcının belirli bir kullanıcıya verdiği geri bildirimi kullanırız. Kullanıcılar ve öğeler benzersiz bir tamsayı kimliği ile temsil edilir.
- Node2vec algoritması kullanılarak, kullanıcılar ve öğeler için özellik vektörü temsili oluşturulur. Daha sonra, node2vec algoritmasını kullanarak, giriş verilerinde bulunan her kullanıcı ve öğe varlığı için sürekli bir vektör üretilir. Node2vec algoritmasının parametrelerini kullanarak, vektörün boyutu belirlenebilir. Sürekli vektör, veri setindeki kullanıcılar ve öğeler için öznitelik temsilindedir. Bu daha sonra sinirsel işbirlikçi filtreleme modeli için girdi haline gelir.
- Sinir İşbirlikçi filtreleme çerçevesini arka plan olarak kullanarak, çerçevenin bir örneği geliştirilmiştir. Belirli bir öğe için kullanıcının derecelendirme değerini tahmin etmek için bir tahmin modeli geliştirilmiştir. Model, önceki adımdan oluşturulan kullanıcı öğesi çiftlerinin node2vec temsili kullanılarak eğitilir.

Sözde Kod (Pseudocode)

RVeriSeti: (Tablo 3.1)' de gösterilen ham tavsiye sistemi veri kümesidir.

GVeriSeti: Node2vec algoritmasına girdi olarak kullanılan veri kümesinin graf temsilidir.

WVeriSeti: Node2vec gömme ve her kullanıcı, öğe çiftindeki ham veri kümesine katılarak oluşturulan yeniden yapılandırılmış veri kümesidir. Bu daha sonra Node2vec_NCF sinir ağı için girdi olarak kullanılır.

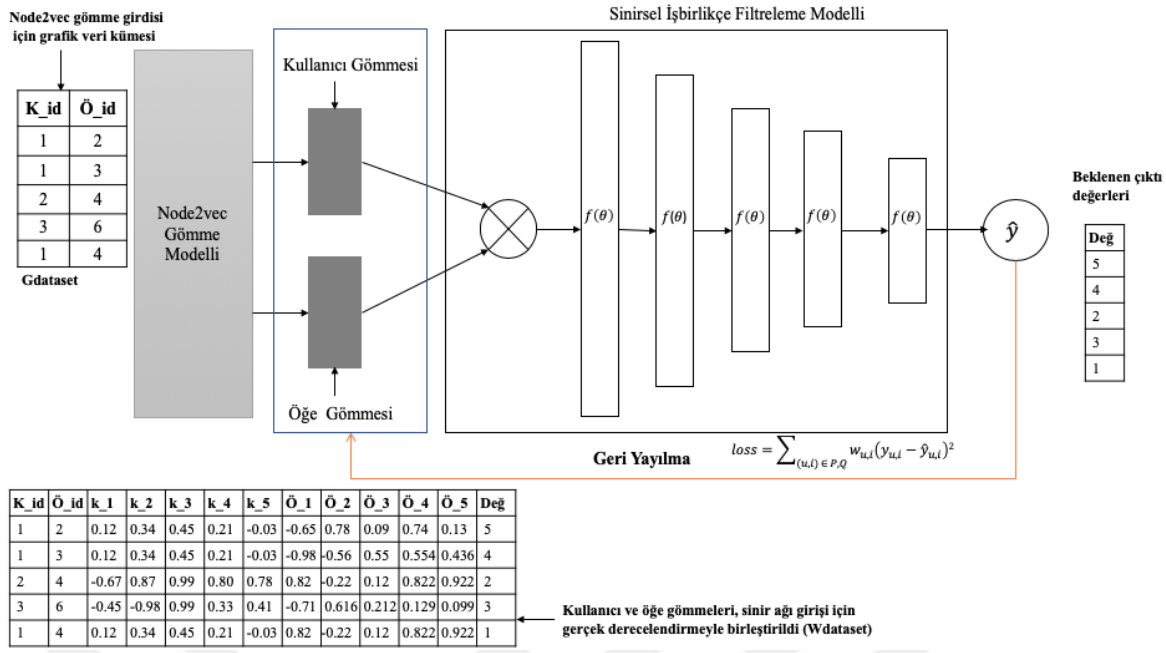
```
Graf_Olustur (RVeriSeti) {
   for each kullanıcı u
      for each öğe i
         if kullanıcı nesneyi değerlendirdi
            demet = {u, i}
            ekle demet to GVeriSeti
   return GVeriSeti
}

Graf_Yeniden_Olusturma (RVeriSeti, W) {
   WVeriSeti = inner_join (RVeriSeti, W ON Kid and Öid)
   return WVeriSeti
}

node2vec_Walk (Graf G` = (V, E, π), Başlangıc düğümü u, Uzunluk l) {
   [u]'yu yürütmeListesi başlangıcı yap
   for yürüt_iterasyonu = 1 to l do
      curr = yürütmeListesi[-1]
      Vcurr = KomşularınıGetir (curr, G`)
      s = TakmaAdÖrneği (Vcurr, π)
      yürütmeListesine s'yi ekle
   return yürütmeListesi
}

Node2vec_Embedding (Graf G = (V, E, W), Boyutlar d, Düğüm başına yürüyüş sayısı r,
Yürüme uzunluğu l, Kaynak genişliği k, Dönüş p, Giriş-Çıkış q) {
   π = ÖncedenDeğiştirilmişAğırlıklar (G, p, q)
   G` = (V, E, π)
```

```
   yürütmeListesi'ni boş olarak başlat
   for iterasyon = 1 to r do
      for düğümler u ∈ V do
         yürütmeListesi = node2vecWalk (G`, u, l)
         yürütmeListelerine yürütmeListesi'ni ekle()
         f = StokastikDereceİndirge (k, d, yürütmeListeleri)
   return f
}
```

Node2vec_NCF (RVeriSeti, Yürüme uzunluğu l, Düğüm başına yürüyüş sayısı r, Dönüş p, Giriş-Çıkış q, Boyut d, Kaynak genişliği k) {

```
      GVeriSeti = GrafOlustur (RVeriSeti)
      W = GömülüOluştur (GVeriSeti, l, r, p, q, d, k)
      MVeriSeti=GrafıYenidenYapılandır (W, RVeriSeti)
      NCF_NN = Kur_NCF_YapaySinirAğı (Katmanların Sayısı, katman aktivasyon
      fonksiyonu, maliyet fonksiyonu)
      RS_model = Eğit_NCF_ YapaySinirAğı (MVeriSeti, regularization_rate rr, dropout
      dr, veri işleme boyut grubu, tur sayısı)
   return RS_model
}
```

Yukarıdaki sözde kod Node2vec_NCF algoritmasının 3 ana bileşenini temsil eder. İlk olarak graf, öneri sistemi veri kümesinden oluşturulur (yani bir kullanıcı, öğe çifti = kullanıcının öğeyi derecelendirdiği graf kenarı). Daha sonra veri kümesi, tüm veriler ve giriş verilerindeki öğeler için sürekli bir vektör veren node2vec algoritmasına giriş olarak kullanılır. Bu adım, giriş olarak adım 1'deki kullanıcı öğesi çiftleri için gömme vektörlerini kullanır. Sinir ağı, gerçek derecelendirmeyi tahmin etmek için eğitilmiştir. Aşağıdaki şekil, sahte kodun görsel bir temsilini vermektedir.



Şekil 8. Önerilen Node2vec-NCF metodolojimizin şematik bir temsili[T2]

Modelin Tasarımı

Python'un Keras işlevsel API'sı ve geliştirilen çift / paralel giriş tasarım yapısı kullanılarak, belirli bir kullanıcının derecelendirmesini tahmin etmek için bir sinir ağı modeli oluşturulmaktadır. He *et al*. (2017), tarafından önerilen sinirsel işbirlikçi filtreleme (NCF) çerçevesine dayalı olarak matris faktörü NCF kullanılarak ifade edilebilir. Çok katmanlı bir algılayıcı (MLP) kullanarak, derin sinir ağları da çerçeve aracılığı ile araştırılabilir. Bu çalışma için ağımızın girdisi, node2vec algoritması kullanılarak oluşturulan gömülü vektörlerdir. Aşağıdaki denklemde NCF çerçevesi altında açıklanan MLP modeli kullanılmaktadır:

$$
\begin{aligned}
x_1 &= \emptyset_1(p_u, q_i) = [p_u\ q_i], \\
\emptyset_2(x_1) &= a_2(W_2^T x_1 + b_2), \\
&\vdots \\
\emptyset_L(x_{L-1}) &= a_L(W_L^T x_{L-1} + b_L), \\
\hat{y}_{ui} &= \sigma\left(h^T \emptyset_L(x_{L-1})\right)
\end{aligned}
\tag{3.6}
$$

Denklem (3.6) şu şekilde açıklanabilir:

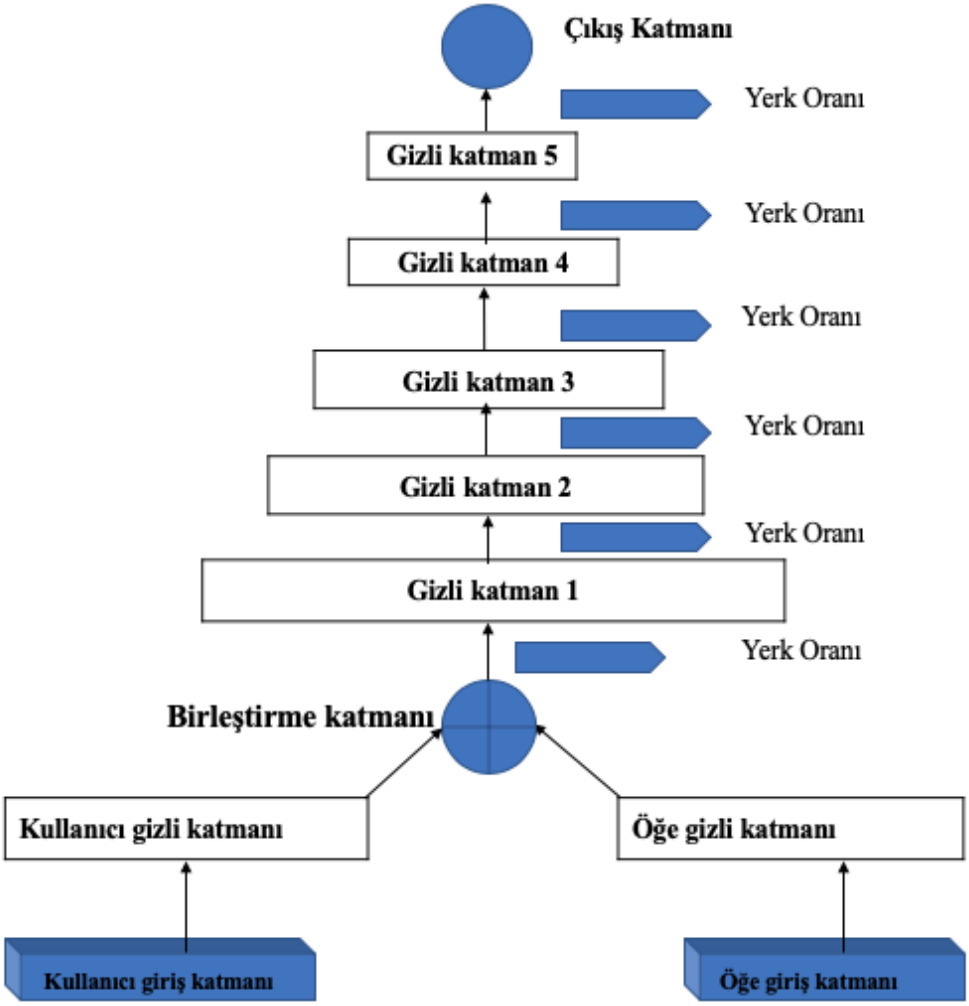
- $\boldsymbol{p_u}$: *u* kullanıcısını temsil eden sürekli vektördür. Sinir ağına girdi olarak kullanılır.
- $\boldsymbol{q_i}$: i öğesini temsil eden sürekli vektördür. Sinir ağına girdi görevi görür.

- $\boldsymbol{x_L}$: sinir ağındaki her nöron için giriş değeridir. Bu değer, önceden bağlanmış bir nöronun aktivasyonunun çıktısıdır. Bu nöronlar, ağ mimarisine ve göreve bağlı olarak katmanlanır.
- $\boldsymbol{\emptyset_L}$: Belirli bir girişi tek bir sayısal çıkışa dönüştüren bir işlevdir. Sinir ağı durumunda, aktivasyon fonksiyonu olarak bilinir.
- $\boldsymbol{W_L^T}$: Tüm nöronların hedef nörona bağlandığı ağırlık vektörleridir.
- $\boldsymbol{b_L}$: her nörona eklenen yanlılık değeridir.
- $\boldsymbol{\hat{y}_{ui}}$: tahmini derecelendirme, yani sinir ağının çıkışıdır.

Modelin Uygulaması

Keras, sinir ağlarını uygulamak için kullanılan bir python kütüphanesidir. Keras'ın iki tip modeli vardır. Sıralı model ve işlevsel API modeli. Sıralı modeller, doğrusal olarak istiflenmiş sinir ağı mimarileri için kullanılır. Diğer yandan fonksiyonel API, kullanıcıya ağ topolojisini oluşturmada daha fazla esneklik sağlar, bu da daha karmaşık modeller oluşturmada kullanışlı bir yöntemdir. Model sinir ağı çift paralel giriş ağ yapısına sahip olmasından dolayı, işlevsel API modelini kullanarak uygulanmaktadır. Paket olarak bulunan node2vec python uygulaması da kullanılmıştır. Belirtilen graf yapısı, veri kümesi manipülasyonu ve değerlendirme fonksiyonu gibi süreçler python programlama dili ile uygulanmıştır. Modelin sinir ağı uygulaması Ek 1'de bulunabilir. Kodun tamamı github deposunda bulunmaktadır(Anonim,2020).

Kullanıcı ve öğe için sırasıyla ağ girişi iki paralel giriş katmanından oluşur. Giriş katmanları bir birleştirme katmanı ile birleştirilir. Birleştirme katmanı, 5 katmandan oluşan gizli bir katman ile MLP'ye bağlanır. Çıkış katmanı, öngörülen derecelendirmeyi veren tek bir nörondur. ReLU aktivasyon fonksiyonu tüm katmanlar için kullanılır. Hiper parametreler epoch, batch_size, regularization_rate ve dropout oranını manuel olarak ayarlanır ve ortalama karesel hata kaybı işlevi kullanılarak öngörülen derecelendirme ile gerçek değer arasındaki hatanın en aza indirilmesi amaçlanır. Model, en iyi sonuçları elde etmek için düzenlenme ve bırakma oranları optimize edilerek eğitilir. Bu denemeler için farklı boyutlarda 5 veri kümesi kullanılır.



Şekil 9. Node2vec-NCF modelimizin sinir ağı bileşeninin mimarisini gösteren diyagram

Veri Setleri

Gerçek tavsiye sistemlerinden toplanan veri setleri çalışmamızda kullanılmıştır. Veri setleri; kullanıcılar, öğeler, kullanıcıların öğelere verdiği dereceler ve derecelendirme zaman damgalarından oluşur. Farklı veri setlerinin özellikleri Tablo 1'de verilmiştir. MovieLens veri setleri, tavsiye sistemleri araştırmalarında en yaygın kullanılan veri setleridir. Fine Foods veri kümesi, bir kullanıcının öğeyi iki kez derecelendirdiğini belirten yinelenen girişler içerir. Sapmayı önlemek için tüm tekrar eden girişler silinerek tek girişe düşürülmüştür. İşlem sayısını azaltmak için ise amazon müzik aletleri veri setleri de örneklendirilmiştir. Books veri seti, MovieLens veri setleri gibi en az 20 öğeyi derecelendiren bir kullanıcı seçilerek güncellenmiştir.

Tablo 1. Gerçek tavsiye sistemleri veri kümesi örneği

Kul_id	Öğe_id	Değerlendirme	Zaman Damgası
1	3	4	964982176
2	5	5	964982150
3	3	2	964982178
1	6	3	964982576
4	1	5	964980176

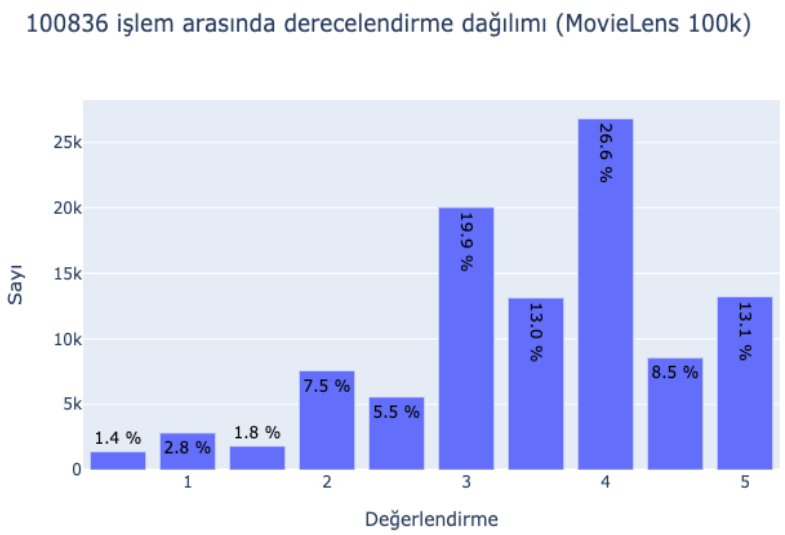
Tablo 2. Veri kümelerinin temel özellikleri

Veri Kümesi	Books	MovieLens 100k	Amazon Musical Instrument	Fine Foods	MovieLens 1m
Kullanıcı Sayısı	1295	610	191633	256059	6040
Öğe Sayısı	14684	9724	59433	74258	3706
İşlem Sayısı	62656	100836	250088	560804	1000208
Derecelendirme Aralığı	1,0 -10,0	0,5 – 5,0	1,0 – 5,0	1,0 – 5,0	1,0 – 5,0
Derecelendirme Ortalaması	7,9	3,5	4,2	4,18	3,58
Minimum Kullanıcı Oyu	20	20	1	1	20

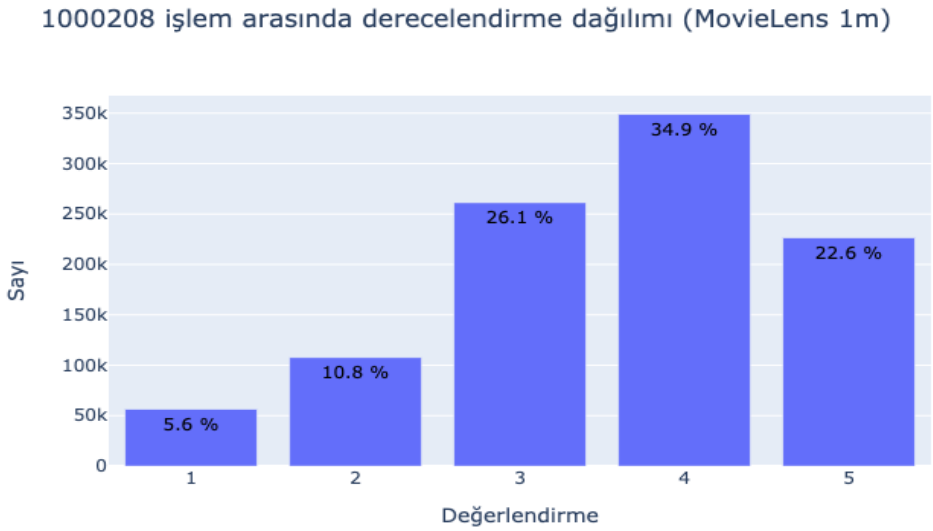
Tablo 3. Veri setlerinin kaynakları

Veri Kümesi	Kaynak
MovieLens 100k	https://grouplens.org/datasets/MovieLens/
MovieLens 1m	https://grouplens.org/datasets/MovieLens/
Amazon Musical Instrument	http://jmcauley.ucsd.edu/data/amazon/index.html
Fine Foods	https://www.kaggle.com/snap/amazon-fine-food-reviews
Books	https://github.com/caserec/Datasets-for-Recommneder-Systems/tree/master/Processed%20Datasets/BookCrossing

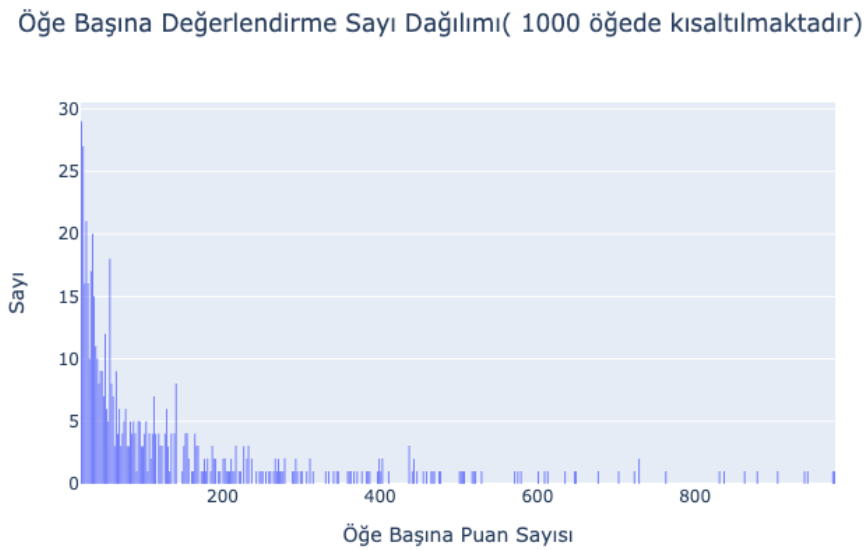
Tavsiye sistemleri veri setlerinin ortak bir özelliği, derecelendirmelerin tüm öğelere eşit dağılmamasıdır. Öğelerin küçük bir alt kümesi genellikle derecelendirmelerin çoğunu içerir. Şekil 3.8., Şekil 3.10., Şekil 3.12., Şekil 3.14. ve Şekil 3.16., her bir veri kümesinin tüm işlemlerde derecelendirme dağılımını; Şekil 3.9., Şekil 3.11., Şekil 3.13., Şekil 3.15. ve Şekil 3.17. ise her bir veri kümesi için öğe başına derecelendirme sayısı dağılımını göstermektedir. Görüldüğü üzere, maksimum kullanıcı sayısı az sayıda kullanıcı arasında yoğunlaşmıştır. Kullanıcıların çoğunluğu, çok az sayıda kullanıcıya oy vermiş ve 50'nin altında bir derecelendirme değerine sahiptir. Bu, tavsiye sistemlerinin doğasının gerçek bir yansımasıdır.



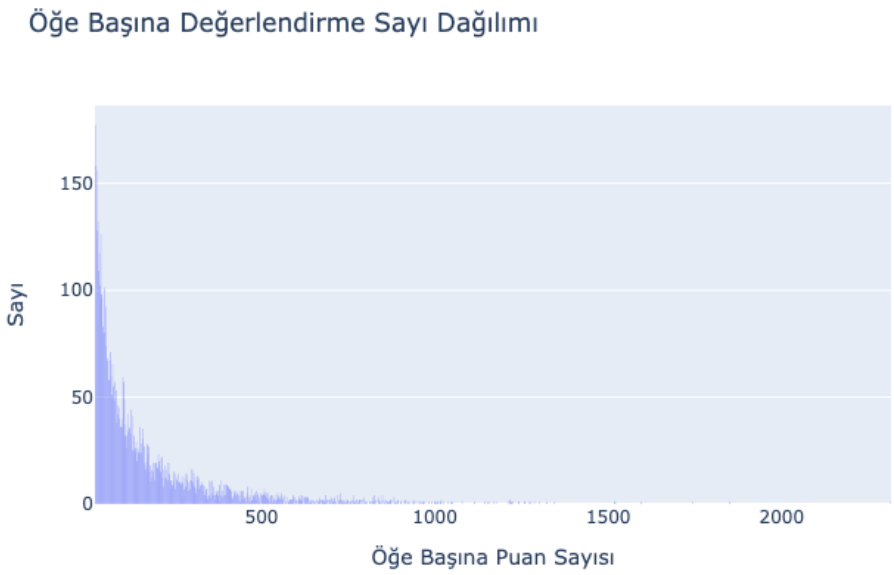
Şekil 10. MovieLens 100k veri kümesinin tüm işlemlerde derecelendirme dağılımı



Şekil 11. MovieLens 1m veri kümesinin tüm işlemlerde derecelendirme dağılımı



Şekil 12. Öğe başına derecelendirme sayısı dağılımı (MovieLens 100k veri kümesi)



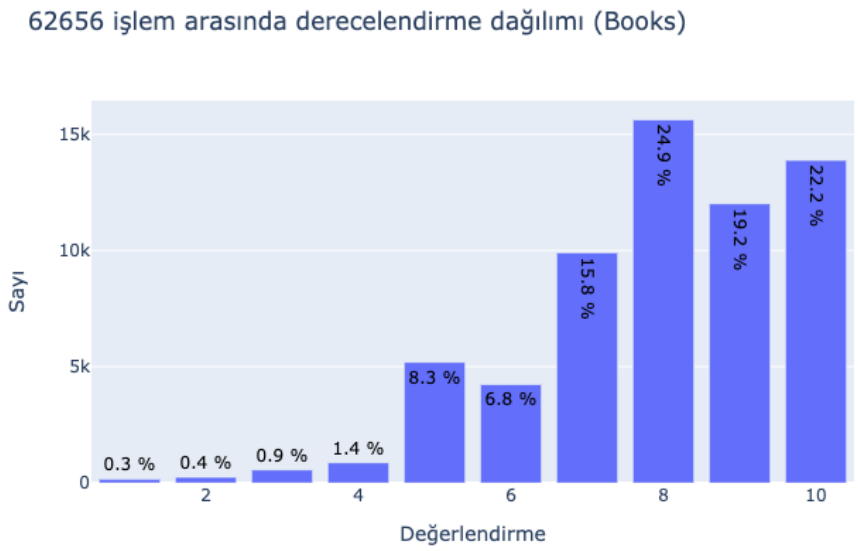
Şekil 13. Öğe başına derecelendirme sayısı dağılımı (MovieLens 1m veri kümesi)



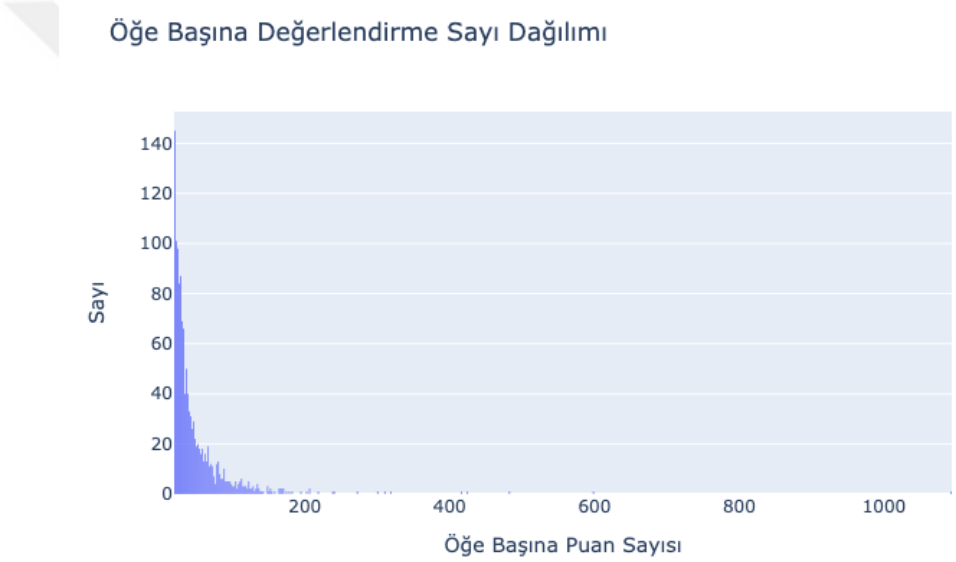
Şekil 14. Amazon veri kümesinin tüm işlemlerde derecelendirme dağılımı



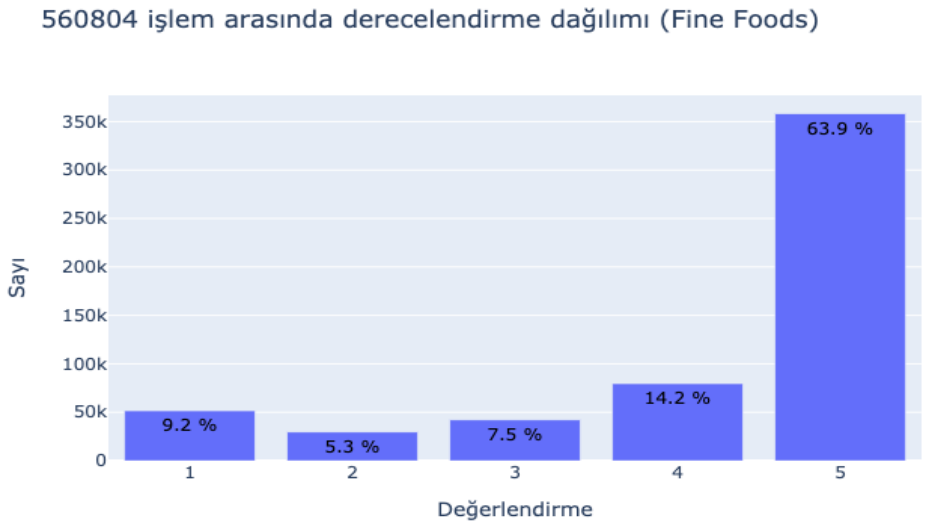
Şekil 15. Öğe başına derecelendirme sayısı dağılımı (Amazon veri kümesi)



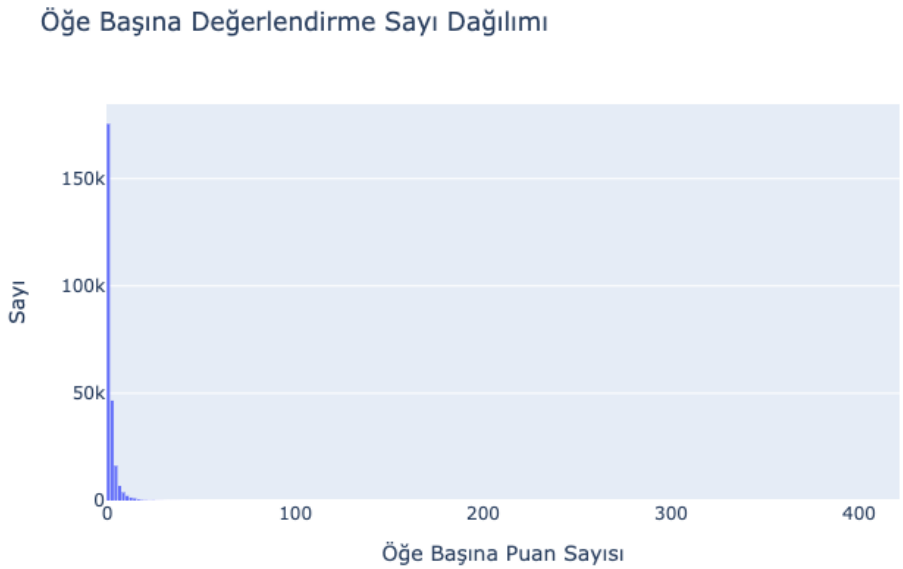
Şekil 16. Books veri kümesinin tüm işlemlerde derecelendirme dağılımı



Şekil 17. Öğe başına derecelendirme sayısı dağılımı (Books veri kümesi)



Şekil 18. Fine Foods veri kümesinin tüm işlemlerde derecelendirme dağılımı



Şekil 19. Öğe başına derecelendirme sayısı dağılımı (Fine Foods veri kümesi)

Deneysel Yöntem

Kullanıcı / öğe düğümü gösterimi için node2vec algoritması kullanılarak, gömülü bir vektör oluşturulmakta ve sinir ağı için giriş vektörü olarak kullanılmaktadır. Çalışmada, vektör gösterimi için şu konfigürasyonlar ile {p = 4; q = 1; boyut = 256; yürüyüş_sayısı = 10; yürüme_uzunluğu = 80}, node2vec algoritması kullanılmıştır. Grover and Leskovec (2016), bağlantı tahmininde düğüm gösterimi için genel bir yapı belirtmemektedir. Bununla birlikte, Palumbo *et al*. (2018), varsayılan ayarların (p = 1; q = 1; yürüyüş_sayısı = 10; yürüme_uzunluğu = 80; pencere boyutu = 10; iter = 1; boyutlar = 128) SVD ve ItemKNN'de daha iyi performans gösterdiğini ama, p = 1,

q = 1 ile node2vec'in temelde derin bir yürüme algoritması olduğunun unutulmaması gerektiğini belirtmektedirler.

Sonra, Şekil 19'daki tavsiye sistemi modelini kullanarak, kullanıcı öğesi çiftlerini node2vec algoritmasından oluşturulan değerlerle değiştirilmiştir. Hedef çıktı olarak karşılık gelen derecelendirmeyle, hiper parametreler, epoch, batch_size, düzenleme oranı ve bırakma değerleri seçilmiştir. Her bir veri kümesi için yukarıdaki hiper parametrelerin optimize edilmiş değerleri Tablo 4'te verilmiştir.

Tablo 4. Çeşitli veri kümeleri için optimize edilmiş çağ, batch_size, bırakma ve düzenleme parametre değerlerini gösteren tablo

Veri Kümesi	Epoch	Batch	Reg	Dropout
Movielens 100k	6	100	0,6	0,001
Movielens 1m	10	500	0,0009	0,1
Fine Foods	10	500	0,5	0,1
Musical Instrument	10	100	0,6	0,08
Books	10	100	0,9	0,8

Değerlendirme

Tavsiye sistemleri, doğru bir şekilde değerlendirme yapma sürecinde basit bir yapıya sahip bir sistem değildir, çünkü performansı gerçek uygulamalarda çevrimiçi işleyişe bağlıdır. Çevrimiçi ve çevrimdışı olmak üzere iki ana tavsiye sistemi değerlendirme türü vardır. Çevrimiçi değerlendirmeler, tavsiye sistemi gerçek zamanlı kullanıcılar tarafından aktif kullanımdayken gerçekleştirilmektedir. Gerçek kullanıcılardan gerçek geri bildirim; A/B testi, Tıklama Oranı (TO), Dönüşüm Oranı (Oran) veya anketler şeklinde sıralanabilir. Ancak çevrimdışı sistemlerde değerlendirme, farklı metrikler kullanılarak modellerdeki mevcut veri kümelerinin test edilmesiyle gerçekleştirilmektedir. Literatürde tanımlanmış farklı değerlendirme ölçütleri vardır. Modelin performansını ölçmek için sayısal değerlendirme metrikleri kullanılmıştır.

Değerlendirme Ölçütleri

Hata Karelerinin Ortalamasının Karekökü (Root Mean Squared Error (RMSE)): Literatürde bulunan en yaygın tahmin değerlendirme ölçütlerinden birisidir. Aşağıdaki formül, RMSE'yi hesaplamak için kullanılmaktadır, burada $\hat{r}_{ui}$, öngörülen puandır, r_{ui}, gerçek puandır ve N, örneklerin sayısıdır. Tahmin edilen değer ile gerçek değer arasındaki hata farkını ölçmektedir. RMSE değeri, büyük hatalara nispeten yüksek ağırlık verir.

$$RMSE_{loss} = \sqrt{\frac{1}{N}\sum_{(u,i)\in N}(\hat{r}_{ui} - r_{ui})^2} \tag{3.7}$$

Ortalama Mutlak Hata (Mean Absolute Error (MAE)): Aynı zamanda RMSE'ye başka bir popüler alternatiftir. Bir tahmin örneğindeki hataların ortalama büyüklüğünü yönlerini dikkate almadan ölçmektedir.

$$MAE_{loss} = \frac{1}{N}\sum_{(u,i)\in N}|\hat{r}_{ui} - r_{ui}| \tag{3.8}$$

MAE, RMSE'ye benzer, ancak ikisi arasındaki temel fark, RMSE'nin Ricci *et al.* (2011)'nın belirttiği gibi büyük hataları abartılı şekilde cezalandırmasıdır (Ricci *et al.*, 2011). MAE ve RMSE değeri arasındaki farkın büyüklüğü, veri setindeki bireysel hatalarda daha büyük bir varyansın göstergesi olmaktadır. MAE'nin her zaman RMSE'den küçük veya eşit olduğuna dikkat edilmelidir.

İsabet Oranı (HitRate)_(HR@K): İsabet oranı (hit rate), listede önerilen öğelerin alaka düzeyinin bir ölçüsüdür. Kullanıcı tarafından derecelendirilen üst düzey tavsiye listelerinde bulunan ve kullanıcının değer verdiği öğelerin kesirlerini ölçmektedir. Her kullanıcı için, isabet oranı aşağıdaki formül kullanılarak hesaplanmaktadır.

$$\text{İ}sabet\ Oranı\ \ (i) = \frac{\sum_{u \in U_T} H_{u,i}}{|U_T|} \tag{3.9}$$

$$Ortalama\ \text{İ}sabet\ Oranı = \frac{\sum_{i \in I_T} \text{İ}sabet\ Oranı(i)}{|I_T|} \tag{3.10}$$

U bir kullanıcıyı temsil ettiğinde, U_T kullanıcı kümesidir, I_T öğe kümesidir ve öğenin üst-k listesinde olup olmamasına bağlı olarak $H_{u,i}$, [1, 0]'dır. Tavsiye sistemleri bağlamında, böyle bir hesaplama zahmetli olabilir. He *et al.* (2017). isabet oranını hesaplamak için bu çalışmada da kullanılan daha basit bir yöntem kullanmışlardır. Öneride bulunmak için kümenin tamamını kullanmak yerine, kullanıcının görmediği 99 öğe rastgele seçilmektedir. Burada kullanıcı seti örneklendirilmektedir. Toplam kullanıcı sayısı 1000'den az olduğu durumda, tüm kullanıcılar başka durumda kullanılır ve 1000 kullanıcı örneklenir.

Hassasiyet (Precision)_P@K: Hassasiyet, üst kümedeki önerilen ilgili öğelerin yüzdesidir. Modelin ilgili tahminlerde bulunma yeteneğini ölçmektedir. İkili bir sınıflandırma modelinde, tanımlanması çok kolaydır. Ancak, tavsiye sistemlerinde, göreceli olduğu için ilgili bir öğeyi tanımlamak çok zor olabilir. Burada, ilgili öğe, kullanıcının gerçekten r>= 3,5 olarak derecelendirdiği ve derecelendirmenin 1-5 arasında değiştiği bir öğe olarak tanımlanır ve r>=6,5 için, 1-10 arasında değişir. Hassasiyet aşağıdaki formülle hesaplanır:

$$\text{P@k} = \frac{(ilgili\ \text{ö}nerilen\ \text{ö}\breve{g}elerin\ sayısı)}{(\text{Ö}nerilen\ \text{ö}\breve{g}e\ sayısı)} \tag{3.11}$$

Hatırlama (Recall)_R@K): Hatırlama, top-k listesinde bulunan zaten derecelendirilmiş öğelerin oranıdır. Hatırlama aşağıdaki formülle hesaplanmaktadır. İlgili maddeler yukarıda açıklandığı gibi belirlenmektedir. Not: Önerilen ve / veya ilgili maddenin sayısı 0 olduğunda, 0'a bölünmeyi önlemek için 1'e bölme işlemi yapılmaktadır.

$$\text{R@k} = \frac{(\text{İ}lgili\ \text{ö}nerilen\ \text{ö}\breve{g}elerin\ sayısı)}{(\text{T}\ddot{u}m\ kullanıcıya\ ilgili\ olan\ \text{ö}\breve{g}elerin\ sayısı)} \tag{3.12}$$

Tahmini kapsam (Predictive Coverage): Tahmini kapsam, sistemin tahminlerde bulunabileceği toplam öğelerin yüzdesini ölçmektedir. (Ge, Delgado-Battenfeld and Jannach, 2010). Bu tez için, I toplam öğe kümesi olarak belirlenirken, Ip, modelin sırasıyla 1-5 veya 1-10 arasında değiştiği minimum 3.5 veya 6.5 derecelendirme aldığı öğeler kümesi olarak belirlenmektedir,

$$\textbf{Tahmini kapsam} = \frac{|I_p|}{|I|} \tag{3.13}$$

Bu ölçümler kullanılarak, modelin performansı değerlendirilmektedir ve literatürdeki diğer modellerle karşılaştırılmaktadır.

Bilgisayar Özellikleri

Bu tez üzerindeki deney, aşağıdaki özelliklere sahip bilgisayarda yapılmaktadır;

- Bilgisayar: Macbook Pro 2012
- RAM: 4 GB
- SSD Drive: 500GB
- İşlemci: 2,5 GHz Intel Core i5

Ayrıca, yoğun görevler gerçekleştirmek için GPU'ya erişim sağlayan Google Colab bulut tabanlı makine öğrenme platformu kaynak olarak kullanılmıştır. Aşağıdaki özelliklere sahiptir;

- RAM: 12 GB
- Hard Disk: 34 GB
- GPU

Eğitim ve Deney Veri Setleri

Modeli eğitmek ve test etmek için, sistemi uygun şekilde değerlendirmek üzere veri seti literatürde uygulanan iki farklı yönteme göre ayrılmıştır. He *et al.* (2017) göre, tüm kullanıcı grubundaki her kullanıcı için, zaman etiketine göre en son oy alan kullanıcı dışarıda bırakılmaktadır. Bu yönteme, literatürde yaygın olarak kullanılan bırakma veya tutma yöntemi denilmektedir. (Ricci, 2011).

Kalan veri seti, 5 defa %80:20 oranında eğitim ve test setlerine ayrılmıştır. Nihai modelin performansı MSE, RMSE, hatırlama ve hassasiyet ölçümleri ile değerlendirilmektedir. İsabet (hit ratio) oranını hesaplamak için, önerilen listede her kullanıcı adına 1 öğe seçerek bırakma yöntemi kullanılmıştır. Zaman damgasının kullanılamadığı Books veri seti için, her kullanıcı adına rastgele bir öğe seçilerek işlemler gerçekleştirilmiştir.

Temel olarak, model literatürde tanımlanan diğer modeler ile karşılaştırılmıştır. Ayrıca, herhangi bir iyileştirme olup olmadığını görmek için farklı veri seti bölümleri kullanılarak modelin performansı da değerlendirilmiştir.

ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu bölümde, modelin sonuçları tartışılmakta ve diğer kıyaslama yöntemlerine göre analiz edilmektedir. Literatürde farklı veri kümelerinde farklı yöntemler kullanılmış ve değerlendirilmiştir. Önerilen algoritmanın performansını anlamak için çevrimdışı değerlendirme kullanıldığını belirtmek önemlidir. Farklı değerlendirme ölçümleri kullanan modelin sonuçları ana hatlarıyla belirtilmiştir. Daha sonra, modelin sonuçları literatürdeki diğer modeller ile karşılaştırılmıştır.

Tavsiye sistemi modellemesinde node2vec gömme tekniği kullanıldığından dolayı, model tavsiye sürecinde node2vec gömme yöntemi kullanan başka bir modelle karşılaştırılmaktadır. Ayrıca modellerin ölçümlerinin gerçek zamanlı olarak performansındaki etkisi de tartışılmıştır. Tablo 4.2., kullanılan tüm veri kümeleri için MAE ve RMSE değerlendirme ölçütleri kullanılarak ölçülen eğitim ve test sonuçlarını göstermektedir. Test setindeki sonuçlara göre, MovieLens 100k ve MovieLens 1m veri setleri için daha iyi sonuçlar elde edilmiştir. Tablo 4.3., önerilen en üst 5 ve 10 öğe için hassasiyet, hatırlama ve isabet oranı değerlerini göstermektedir. Şekil 4.1., farklı veri kümeleri için öğelerin tahmini kapsam yüzdesini göstermektedir. Tahmini kapsam, iki MovieLens veri kümesi ve şaşırtıcı bir şekilde Books veri kümesi ile en iyi performansı göstermektedir. Yine de dikkat edilmesi gereken önemli bir nokta, bu üç veri kümesinin en az 20 puanla derecelendirilmiş kullanıcı içermesidir.

Modelimizin sonuçlarına daha iyi bir bakış açısı sağlamak için Tablo 5'de veri setlerinin derecelendirme değerlerinin istatistiksel analizi verilmiştir.

Tablo 5. Tüm veri kümeleri için test veri setlerinin derecelendirmeye dayalı istatistiksel özellikleri

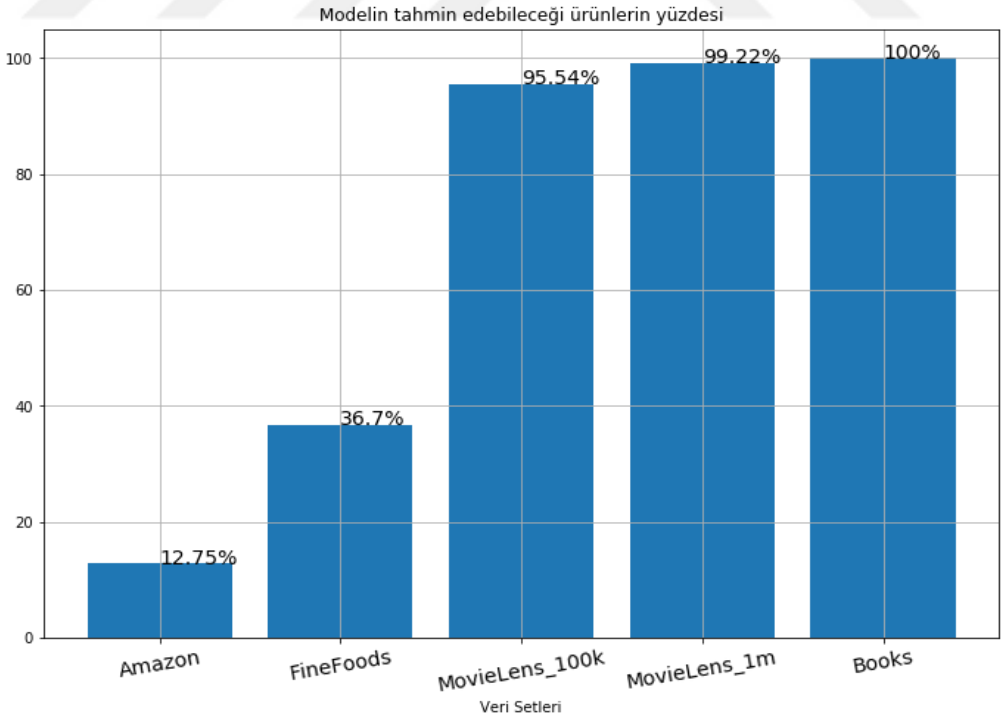
Veri Seti	Minimum	Maksimum	Ortalama	Standart Sapma
Movielens 100k	0,5	5,0	3,500	1,036
Movielens 1m	1,0	5,0	3,583	1,116
Amazon Musical Instruments	1,0	10,0	7,948	1,696
Fine Foods	1,0	5,0	4,209	1,236
Books	1,0	5,0	4,357	1,045

Tablo 6. Modelin eğitim ve test sonuçları

Veri Seti / Ölçüt	Eğitim Sonuçları		Deneysel Sonuçlar	
	MAE	**RMSE**	**MAE**	**RMSE**
MovieLens 100k	0,5602	0,7258	0,6377	0,8331
MovieLens 1m	0,7145	0,8930	0,7211	0,9098
Amazon Musical Instrument	0,5705	0,7695	0,7824	1,0519
Fine Food	0,6702	0,8931	0,7794	1,0527
Books	0,9637	1,2635	1,1839	1,5424

Tablo 7. Hassasiyet, hatırlama ve isabet oranı metriklerinin üst 5 ve 10 öğe için değerlendirme sonuçları

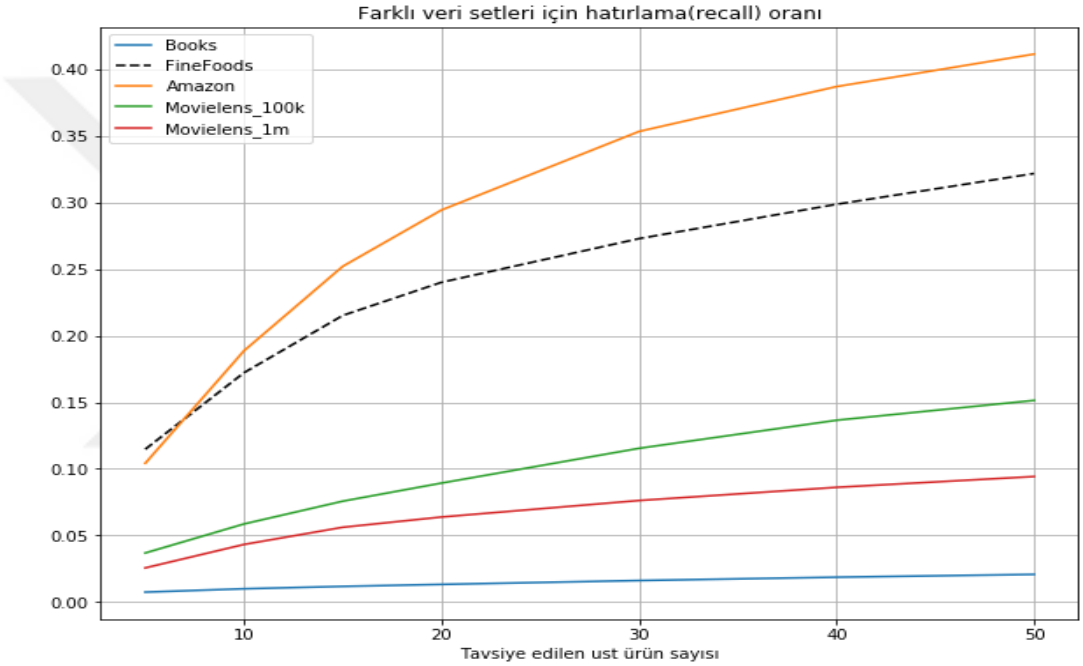
Veri Seti	P@5	P@10	R@5	R@10	HR @5	HR @10
MovieLens 100k	0,297	0,2672	0,0368	0,0586	0,0049	0,0082
MovieLens 1m	0,3279	0,2939	0,0256	0,0432	0,002	0,003
Fine Food	0,1663	0,1568	0,1147	0,1722	0,0	0,0
Books	0,051	0,039	0,0074	0,0099	0,001	0,002
Amazon Musical Instrument	0,2608	0,2444	0,1042	0,1887	0,0	0,0



Şekil 20. Tüm veri kümelerindeki öğelerin tahmini kapsamı

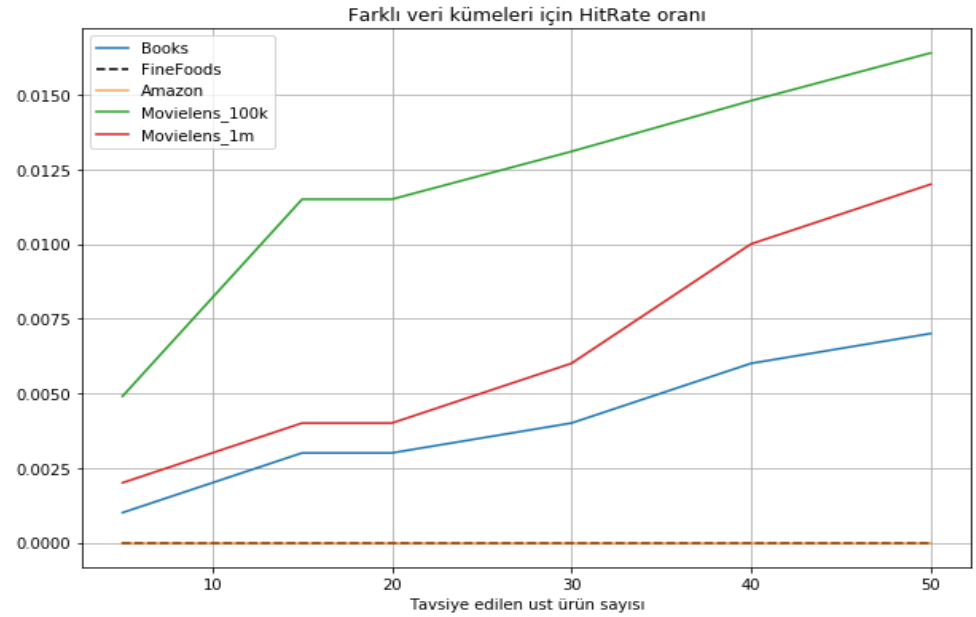
Tüm veri setlerinde, 5, 10, 15, 20, 30, 40, 50 için en iyi N öğenin kesinlik değeri, geri çağırma ve isabet oranları hesaplanmaktadır. Hem hatırlama (recall) hem de isabet oranı (hit

rate) metrikleri için, değerler önerilen en üst kümedeki öğe sayısına göre değişmektedir. Tüm veri setleri için hatırlama oranı, Şekil 4.2.'de kaydedilmiştir. Önerilen kümedeki en üstteki öğe sayısı arttıkça, hatırlama da artmaktadır. Amazon veri seti kaydedilen en yüksek hatırlama oranını göstermiştir. Kullanıcı daha önce gördüğü öğeler yerine yeni öğeler öneren bir tavsiye sistemi istediği için düşük hatırlama değerleri görülmüştür ve bu aslında kötü bir şey değildir. Hit rate oranı ayrıca sistemin veri kümesindeki her kullanıcı için bir öğe önerme yeteneğini de hesaplamaktadır. Şekil 4.3.'de görülebileceği gibi, Hit rate, üst N büyüklüğüyle değişmektedir. Dikkat edilmesi gereken önemli nokta, Fine Foods ve Amazon veri setlerinin %0 isabet oranı kaydetmesidir. Bu, kullanıcıların büyük çoğunluğunun 20'den daha az öğe puanlamasıyla ilişkilendirilebilir.

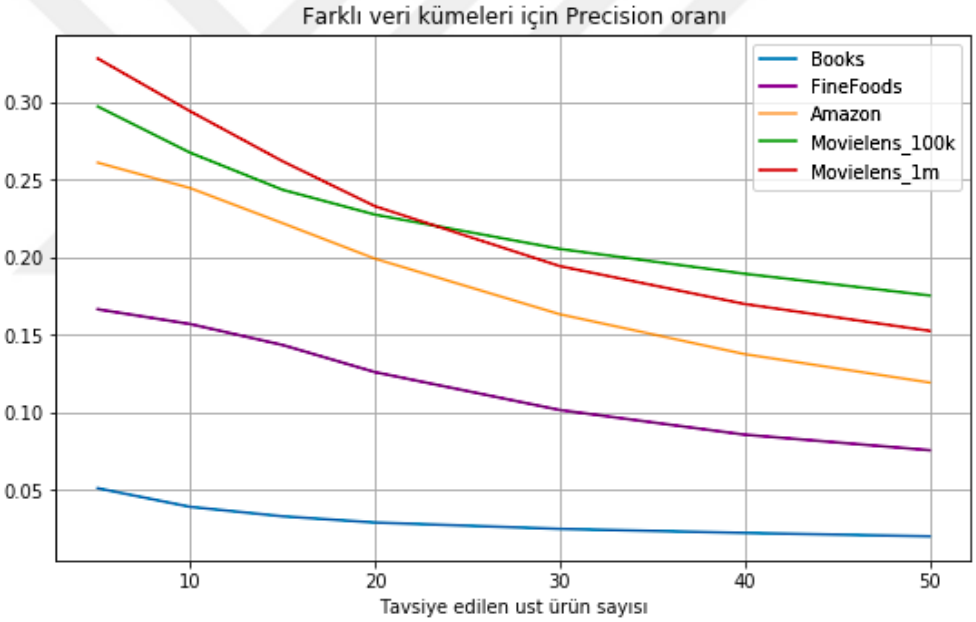


Şekil 21. Tüm veri kümeleri için en iyi önerilen N adet öğe için hatırlama değeri

Hatırlama gibi hassaslık da N sayısıyla değişmektedir, ancak değişiklik tersine dönmektedir. Önerilen en üstteki setin boyutu arttıkça hassasiyet azalmaktadır.



Şekil 22. Tüm veri setleri için en iyi önerilen N adet öğe için isabet oranı.



Şekil 23. Tüm veri setleri için en iyi önerilen N adet öğe için hassasiyet oranı

Modelin performansını anlamak için literatürde bulunan diğer tavsiye sistemi modelleri ile karşılaştırma yapılmıştır. Tavsiye edilen sistemlerin karşılaştırılması zordur çünkü kullanılan modeller çoğunlukla farklıdır ve bu nedenle oldukça karşılaştırılamaz durumdadır.

Modelimiz, Palumbo *et al.* (2018) tarafından "Öğe Önerisi için node2vec ile Bilgi Grafiği Gömmeleri" başlıklı makalede açıklanan bir modelle karşılaştırıldı. Bu makalede araştırmacılar, node2vec gömme algoritmasını kullanıcıyı, öğeleri ve diğer varlıkları bir bilgi grafiğine dönüştürmek için kullanmaktadır. Movielens 1m veri seti kullanılarak, bilgi

grafiğini ve graf düğümlerini yeniden oluşturmak için kullanıcılar, öğeler ve bir film veri seti diğer öğeleriyle birlikte oluşturulur. Bir graf, sadece bir çift düğüm olan bir kenar listesi tarafından temsil edilir. Kullanıcı ile 4 veya daha fazla derecelendirmeyi paylaşan bir öğe arasında kenar oluşturmak için seçilir. Bu, bir kullanıcı 4 veya daha yüksek bir öğeyi derecelendirdiğinde, kullanıcı öğesi çiftinin graf kenar boşluğuna eklendiği anlamına gelmektedir. Kenarlar, diğer varlıkların kullanıcıları ile öğeler ve aralarında bir ilişki varsa oluşturulur. Düğümler node2vec algoritması kullanılarak oluşturulduktan sonra, Palumbo *et al*. (2018) kosinüs benzerlik ölçüsünü belirli bir öğeye göre öğe tavsiye vermek için bir sıralama işlevi olarak kullanır. Tablo 4.4, modelimiz ile Palumbo *et al*. (2018) tarafından yayınlanan MovieLens 1m veri seti için hassasiyet ve hatırlama metriklerinin sonuçları arasındaki bir karşılaştırmayı göstermektedir. Palumbo *et al*. (2018), model değerlendirmelerini bilgi grafiği yerleştirme için 2 farklı gömme modeli, yani node2vec (opt) ve node2vec (default) kullanarak gerçekleştirmiştir. İki model arasındaki fark, gömme işlemi sırasında kullanılan gömme hiper parametresidir.

Node2vec (default): Bu model node2vec algoritmasının varsayılan hiper parametrelerini kullanır, yani {p = 1, q = 1, num walks = 10, yürüyüş uzunluğu = 80, pencere boyutu = 10, iter = 1, boyutlar = 128}.

Node2vec (opt): node2vec (opt), {p = 4, q = 1, num walks = 50, yürüyüş uzunluğu = 100, pencere boyutu = 30, iter = 5, boyutlar = 200} gibi hiper parametrelerin optimize edilmiş bir yapılandırmasını kullanır.

Sürpriz olarak bilinen standart bir öneri kitaplığı kullanarak, modelimizin RMSE, MAE, hassasiyet ve hatırlama sonuçları kütüphane tarafından üretilen standart öneri sistemleri modelleriyle karşılaştırıldı. Hassasiyet ve hatırlama değerleri önerilen en üst 5 ve en üst 10 öge için hesaplanmıştır. Node2vec-NCF'ye kıyasla farklı standart kütüphane modelinin RMSE ve MAE sonuçları Tablo 4.5'de özetlenmiştir. Node2vec-NCF modelimizle karşılaştırıldığında farklı standart modellerin kesinlikleri ve hatırlama sonuçları Tablo 8'de verilmiştir.

Tablo 8. MovieLens 1m için hassasiyet ve hatırlama oranları sonuçları (Palumbo *et al*. 2018)

Model	**P@5**	**P@10**	**R@5**	**R@10**
Node2vec-NCF	**0,328**	**0,294**	0,026	0,043
node2vec (opt)	0,224	0,196	0,095	0,155
node2vec (default)	0,120	0,129	0,067	0,122
Entity2vec	0,2814	0,2127	-	-
MostPop	0,145	0,129	0,049	0,085
SVD	0,068	0,062	0,020	0,037
ItemKNN	0,057	0,054	0,019	0,032
Random	0,007	0,007	0,002	0,003

Tablo 8'te görüldüğü üzere, modelimizin hassasiyet metriği değerlerinin node2vec uygulamalarından (Palumbo et al.,2018), yani node2vec (opt) ve node2vec'e (default) göre sonuç olarak daha iyi performans gösterdiği gözlemlenmiştir. Aynısı, Palumbo et al. (2018) tarafından yapılan taban çizgisi karşılaştırmasındaki diğer tüm modeller için de geçerlidir.

Tablo 9. Modelin, sürpriz tavsiye sistemi kütüphanesi kullanılarak uygulanan temel modellerle karşılaştırılma sonuçları

		SVD	**SVD++**	**NMF**	**Baseline**	**Random**	**Node2vec-NCF**
MovieLens 100k	***rmse***	0,934	0,92	0,963	0,944	1,514	**0,8331**
	mae	0,737	0,722	0,758	0,748	1,215	**0,6377**
MovieLens 1m	***rmse***	0,873	**0,862**	0,916	0,909	1,504	0,9098
	mae	0,686	**0,673**	0,724	0,719	1,206	0,7211
Fine Foods	***rmse***	1,088	**0,759**	1,154	1,173	1,666	1,0527
	mae	0,797	1,066	0,825	0,907	1,245	**0,7794**
Books	***rmse***	3,060	3,417	3,527	3,417	3,450	**1,5424**
	mae	3,417	3,060	3,190	3,059	3,090	**1,1839**
Amazon	***rmse***	1,141	1,146	1,302	1,141	1,534	**1,0519**
	mae	0,863	0,863	1,013	0,869	1,138	**0,7854**

Tablo 9'da, standart kütüphanenin diğer modellerine karşı modelimiz için RMSE ve MAE değerleri gösterilmektedir. Movielens 1m veri seti dışındaki tüm veri setlerinde, modelimiz en iyi (en düşük) MAE değerlerini kaydeder. Modelimize ait RMSE sonuçları Movielens 1m ve Fine Foods veri kümeleri hariç tüm veri kümelerinde en iyi sonucu vermektedir.

Tablo 10. Sürpriz tavsiye sistemi kütüphane modellerinin ve modelimizin (node2vec-ncf) hassasiyet ve hatırlama değerlerinin tüm veri setleri için sonuçlarının karşılaştırılması

Movielens 100k				
	P@5	**P@10**	**R@5**	**R@10**
Node2vec-NCF	**0,297**	**0,2672**	**0,0368**	**0,0586**
SVD	0,1144	0,1131	0,0021	0,0043
NMF	0,0672	0,0699	0,0012	0,0027
Random	0,0788	0,0725	0,0014	0,0023
Baseline	0,10762	0,1064	0,003854	0,00191
SVD++	0,1275	0,10100	0,006709	0,001673
Movielens 1m				
	P@5	P@10	R@5	R@10
Node2vec-NCF	**0,3279**	**0,2939**	**0,0256**	**0,0432**
SVD	0,13525	0,13637	0,00364	0,007285
NMF	0,084	0,09163	0,00189	0,004422
Random	0,11725	0,1110	0,002987	0,00577
Baseline	0,1029	0,10100	0,00356	0,00167
SVD++	0,13912	0,1375	0,007644	0,00373
Amazon musical instruments				
	P@5	P@10	R@5	R@10
Node2vec-NCF	**0,2608**	**0,2444**	**0,1042**	**0,1887**
SVD	0,00025	0,000125	0,000179	0,000179
NMF	0,00025	0,000125	8,92857e-05	0,000109
Random	0,0	0,000125	0,0	2,0492e-05
Baseline	0,000125	0,0	0,0003125	0,0
SVD++	0,000125	0,00025	1,126126e-05	1,12613e-05
Fine Foods				
	P@5	**P@10**	**R@5**	**R@10**
Node2vec-NCF	**0,1663**	**0,1568**	**0,1147**	**0,1722**
SVD	0,00025	0,00025	7,3529e-0.5	0,000124
NMF	0,0	0,00025	0,0	1,7089e-05
Random	0,0	0,0	0,0	0,0
Baseline	0,000125	0,0	5e-05	0,0
SVD++	0,0	0,0	0,0	0,0
Books				
	P@5	P@10	R@5	R@10
Node2vec-NCF	0,051	0,039	0,0074	0,0099
SVD	0,0	0,0	0,0	0,0
NMF	0,0	0,0	0,0	0,0
Random	0,0	0,0	0,0	0,0
Baseline	0,0	0,0	0,0	0,0
SVD++	0,0	0,0	0,0	0,0

Tablo 10'da, modelimiz ve standart sürpriz kütüphanelerinin modellerinde test edilen tüm veri setleri için ilk 5 ve ilk 10 hassasiyet ve hatırlama değerlerinin sonuçları özetlenmiştir. Tüm veri setleri ve standart kütüphane modellerine karşı, modelimizin en iyi hassasiyet ve hatırlama sonuçlarını verdiği görülmektedir. Books veri kümesinde, standart

kütüphane modellerinin node2vec-NCF modelimiz dışındaki tüm modellerde tahmin ve hatırlama metrikleri üzerine 0,0 değeri kaydettiğini belirtmek önemlidir.

Genel olarak, modelimiz doğruluk metriklerinin yanı sıra hassasiyet ve hatırlama metriklerinde geleneksel yöntemlerden daha iyi performans göstermiştir. Modelimizin performansı, node2vec gömme yönteminin veri setinde bulunan gizli özellikleri gerçekten yakalama yeteneğiyle ilişkilendirilebilir. Ayrıca, kullanıcıların tercih ettiği öğelerin yakınlığını ve öğelerle olan ilişkisini de yakalayabilir. Bu node2vec algoritmasına giriş için kullanılan grafiğin oluşturulma biçimine bağlanabilir. Palumbo et al. (2018) önerdiği gibi, bir kullanıcının öğeye verdiği derecelendirme değerine göre graf kenarları oluşturulmamıştır. Grafiğimizde, daha önce kesişen yollara sahip tüm öğeler ve kullanıcılar arasında bir ilişki kurulmuştur. Modelimizin sinir ağı bileşeni, basit bir kümeleme algoritması veya benzerlik tabanlı bir ölçü ile yapılamayan karmaşık özellikleri de yakalayabilme kabiliyetine sahiptir.

Modelimizin nitelikleri aşağıdaki gibi özetlenebilir:

- Node2vec algoritması, graf düğümlerinin fiziksel organizasyonunu yakalama yeteneğine sahiptir. Kullanıcı ve öğe varlıklarının node2vec kodlamasını kullanarak, doğrudan ilişkili olmalarına rağmen varlıklar arasında var olan bazı fiziksel ilişkileri yakalanabilir.
- Kullanıcı ve öğe kimliklerinin kullanımı için, kullanıcıları ve öğeleri birbirine bağlayan bir ilişkinin varlığı sayesinde, bu model kolayca uygulanabilir.
- Önerilen modelimiz, öğe yorumlar, öğeler ve kullanıcı özellikleri gibi farklı veri türleri içerecek kadar esnektir.

SONUÇ

Bu çalışmada, herhangi bir tavsiye sisteminin nasıl çalıştığını anlamak için bir arka plan hazırlanmıştır. Böylece bu alandaki çalışmalara öncü olabilecek ve katkı sunabilecek bir bilgi tabanı oluşturulmuştur. Yeni bir gömme tekniği ve sinirsel işbirliğine dayalı filtreleme yaklaşımını birleştiren yeni bir tavsiye sistemi modeli sunulmaktadır. Model, bir graftaki düğümler için vektör gösterimi üreten node2vec gömme tekniğini kullanmaktadır. Veri setine, düğümlerin veri niteliklerini temsil ettiği ve kenarların aralarındaki ilişki olduğu iki taraflı bir graf perspektifinden bakılması gerekir. Node2vec yaklaşımı, doğal dil işlemede çok önemli olan word2vec tekniğinden esinlenmiştir. Node2vec gömme yöntemi, düğümler arasındaki fiziksel yakınlığın yanı sıra, benzerliği veya benzerlikleri gösteren yapısal bütünlükleri de yakalayabilir. Gömmeler sinirsel işbirlikçi filtreleme çerçevesi için giriş olarak kullanılır. Sinirsel işbirlikçi filtreleme ağı, her bir kullanıcı çifti için derecelendirmeyi öngören paralel sinir ağı mimarisine sahip çok katmanlı bir algılayıcıdır. Yapay sinir ağı, gerçek ve öngörülen derecelendirme arasındaki hatayı minimize ederek optimize eder. Tavsiye sistemlerinin değerlendirilmesi ve kıyaslanması zor olmasından dolayı, model türleri değişkendir ve adil bir şekilde karşılaştırılamaz. Farklı veri setleri kullanılarak, sonuçları literatürdeki kıyaslamalar ile karşılaştırmanın yolları bulunmaya çalışılmıştır.

RMSE metriği, tahmin doğruluğunu değerlendirmede en yaygın kullanılan ölçümlerden birisidir. Model, RMSE metriği ile değerlendirildiğinde geleneksel tavsiye sistemi modellerinden daha iyi performans gösterdiği saptanmıştır. Bu modelin sonuçları, literatürdeki RMSE puanı açısından son teknolojiye sahip yöntemlerin bazıları kadar rekabetçi olmasa da, göz ardı edilemeyecek bir başarıyı ortaya koymaktadır.

Bu çalışmada ayrıca, node2vec gömme yöntemi, tavsiye sistemleri için kullanan diğer modellerle de karşılaştırılmıştır. Model, hassas ölçümle daha iyi sonuçlar vermektedir. Model göreceli olarak daha düşük hatırlama oranına sahiptir ve bundan dolayı modelin kullanıcıya ne denli iyi yeni ürünler önerdiğinin bir göstergesi olduğu da söylenebilir.

Ayrıca modelin önerebileceği öğelerin yüzdesi de ölçülmüştür. Modelin değerlendirildiği 5 veri kümesinden üçü, %90'ın üzerinde öğe kapsamına sahip olduğunu göstermektedir. Bu, çevrimiçi sistemlerde modelin çok etkili olduğunu kanıtlayabilir.

Gelecekteki çalışmalarda, bu modeli ve yaklaşımı daha da geliştirmek için aşağıdaki yaklaşımlar düşünülebilir:

- Daha iyi bir model performansı sağlanması için, node2vec algoritmasının parametre konfigürasyonunun araştırılması.
- Verilen geri bildirime dayanmayan, tüm varlıkları azaltan, eşit derecede güçlü bir bağlanma ilişkisine dayanmasını sağlayan grafı oluşturmak için farklı bir yaklaşımın araştırılması.
- Daha iyi tahminler yapabilmek için diğer sinir ağı yapılandırmalarının keşfedilmesi.

Sonuç olarak, tavsiye sistemlerini pratikte değerlendirmek çok zordur, çünkü önerilerinizin kullanıcı tarafından çevrimiçi sistemlerde çevrimdışı olarak ne kadar sevildiğini ölçmenin bir yolu yoktur. Bu nedenle, bir tavsiye sisteminin en iyi test edildiği zaman, nihai testin kullanıldığı zamandır. Buna rağmen, çevrimdışı değerlendirmeler bize hangi modellerin üretim modunda denenmesi gerektiğini belirleme konusunda rehberlik eder.

KAYNAKLAR

Anonim, 2020. node2vec-NCF-Recommender-system.https://github.com/yolita/node2vec-NCF-Recommender-system

Ashley-Dejo, E., Ngwira, S., & Zuva, T. (2015, December). A survey of context-aware recommender system and services. In *2015 International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-6).

Berg, R. V. D., Kipf, T. N., & Welling, M. (2017). Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Catherine, R., & Cohen, W. (2017, August). Transnets: Learning to transform for recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 288-296).

Chen, H. H. (2017). Weighted-SVD: Matrix factorization with weights on the latent factors. *arXiv preprint arXiv:1710.00482*.

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, *25*(2), 99- 154.

Chu, W. T., & Tsai, Y. L. (2017). A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*, *20*(6), 1313-1331.

Collaborative Filtering on MovieLens 100K (n.d.). Retrieved from https://paperswithcode.com/sota/collaborative-filtering-on-movielens-100k

Collaborative Filtering on MovieLens 1M (n.d.). Retrieved from. https://paperswithcode.com/sota/collaborative-filtering-on-movielens-1m

Dong, R., Schaal, M., O'Mahony, M. P., & Smyth, B. (2013, June). Topic extraction from online reviews for classification and recommendation. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Einstein, Michael. "Some Amazing Statistics about Online Data Creation and Growth Rates." *Information Overload Research Group Join the Fight for Reducing Information Pollution*, 17 Apr. 2019, https://iorgforum.org/case-study/some-amazing-statistics-about-online-data-creation-and-growth-rates/.

Elior C., (2018, April 16). *node2vec: Embeddings for Graph Data*. Retrieved from https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef.

Elkahky, A. M., Song, Y., & He, X. (2015, May). A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 278-288).

Gao, Y., Yu, W., Chao, P., Zhang, R., Zhou, A., & Yang, X. (2015, April). A restaurant recommendation system by analyzing ratings and aspects in reviews. In *International Conference on Database Systems for Advanced Applications* (pp. 526-530). Springer, Cham.

Garcia Esparza, S., O'Mahony, M. P., & Smyth, B. (2011, October). A multi-criteria evaluation of a user generated content based recommender system. In *3rd Workshop on Recommender Systems and the Social Web (RSWEB-11), 5th ACM Conference on Recommender Systems, Chicago, IL, USA, 23-27 October 2011*.

Gatzioura, A., & Sànchez-Marrè, M. (2014). A case-based recommendation approach for market basket data. *IEEE Intelligent Systems*, *30*(1), 20-27.

Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010, September). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM Conference on Recommender Systems* (pp. 257-260).

Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855-864).

He, L., & Wu, F. (2009, August). A time-context-based collaborative filtering algorithm. In *2009 IEEE International Conference on Granular Computing* (pp. 209-213).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 173-182).

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261- 273.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30-37.

Kumar, B., & Sharma, N. (2016). Approaches, issues and challenges in recommender systems: a systematic review. *Indian J. Sci. Technol*, *9*(47), 1-12.

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196).

Li, L., Zhou, Y., Xiong, H., Hu, C., & Wei, X. (2017, March). Collaborative filtering based on user attributes and user ratings for restaurant recommendation. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 2592-2597).

McAuley, J., & Leskovec, J. (2013, October). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems* (pp. 165-172).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

MovieLens(n.d.). Retrieved from https://Grouplens.org/Datasets/MovieLens/

Musat, C. C., Liang, Y., & Faltings, B. (2013, June). Recommendation using textual opinions. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Palumbo, E., Rizzo, G., & Troncy, R. (2017, August). Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 32-36).

Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., & Ferro, E. (2018, June). Knowledge graph embeddings with node2vec for item recommendation. In *European Semantic Web Conference* (pp. 117-120). Springer, Cham.

Patel, B., Desai, P., & Panchal, U. (2017, March). Methods of recommender system: a review. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-4).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 701-710).

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (pp. 175-186).

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender Systems Handbook* (pp. 1-35). Springer, Boston, MA.

Said, A., & Bellogín, A. (2014, October). Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems* (pp. 129-136).

Schafer, J. B., Konstan, J., & Riedl, J. (1999, November). Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158-166).

Shah, K., Salunke, A., Dongare, S., & Antala, K. (2017, March). Recommender systems: An overview of different approaches to recommendations. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-4).

Sharma, R. D., Tripathi, S., Sahu, S. K., Mittal, S., & Anand, A. (2016). Predicting online doctor ratings from user reviews using convolutional neural networks. *International Journal of Machine Learning and Computing*, *6*(2), 149.

Tso-Sutter, K. H., Marinho, L. B., & Schmidt-Thieme, L. (2008, March). Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM Symposium on Applied Computing* (pp. 1995-1999).

Vargas-Govea, B., González-Serna, G., & Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys*, *11*(592), 56.

Wen, Y., Guo, L., Chen, Z., & Ma, J. (2018, April). Network embedding based recommendation method in social networks. In *Companion Proceedings of the The Web Conference 2018* (pp. 11-12). International World Wide Web Conferences Steering Committee.

Xu, Z., Lukasiewicz, T., Chen, C., Miao, Y., & Meng, X. (2017, August). Tag-aware personalized recommendation using a hybrid deep model. AAAI Press/International Joint Conferences on Artificial Intelligence.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, *52*(1), 5.

Zhang, W., Ding, G., Chen, L., Li, C., & Zhang, C. (2013). Generating virtual ratings from chinese reviews to augment online recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *4*(1), 9.

Zhang, Y., Ai, Q., Chen, X., & Croft, W. B. (2017, November). Joint representation learning for top-n recommendation with heterogeneous information sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1449-1458).

Zheng, L., Noroozi, V., & Yu, P. S. (2017, February). Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (pp. 425-434).

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Yolande Margaret Goswell
Doğum tarihi:	21.12.1991
Doğum Yeri:	Ziguinchor, Senegal
Uyruğu:	Gambiya
Adres:	Lamin, West Coast Region, The Gambia
E-mail:	gyolandem@gmail.com
Eğitim	
Lise:	St. Joseph's Senior Secondary School
Lisans:	Sharda University
Yüksek lisans:	Atatürk Üniversitesi
Yabancı Dil Bilgisi	
İngilizce:	İyi
Diğer:	Fransızca