


Öneri Sistemlerinde Veri Seyrekliği Problemine Otomatik Kodlayıcı Yaklaşımlarının Karşılaştırmalı Bir Çalışması

A Comparative Study of Autoencoder Approaches to the Data Sparsity Problem in Recommender Systems

Ecem Bölük¹ , Mustafa Özgür Cingiz² 

¹Bilgisayar Mühendisliği Ana Bilim Dalı, Lisansüstü Eğitim Enstitüsü, Bursa Teknik Üniversitesi, 16310, Bursa, Türkiye

²Bilgisayar Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, Bursa Teknik Üniversitesi, 16310, Bursa, Türkiye

(ecemboluk@gmail.com, mustafa.cingiz@btu.edu.tr)

Received:Sep.08,2022

Accepted:Sep.16,2022

Published:Oct.10,2022

Özetçe— Öneri sistemleri kullanıcıların geçmişteki tercihlerinden hareketle gelecekteki tercihlerini tahmin eden sistemlerdir. Fakat kullanıcılar her zaman tercihlerini sistemlere belirtmeyebilir. Bu durum, öneri sistemleri tasarlanırken karşılaşılan en büyük sorunlardan biri olan veri seyrekliğine neden olur. Derin öğrenme algoritmalarından otomatik kodlayıcılar, seyrek kullanıcı matrisini verilerden öğrendiği iç görülerden hareketle yeniden doldurarak veri seyrekliği probleminin çözülmesini sağlar. Bu çalışmada derin öğrenme algoritmalarından Temel Otomatik Kodlayıcı, Gürültü Giderici Otomatik Kodlayıcı, Seyrek Otomatik Kodlayıcı ve Varyasyonel Otomatik Kodlayıcı olmak üzere dört farklı otomatik kodlayıcı modeli kullanılarak veri seyrekliğine karşı performansları karşılaştırılmıştır. Veri seti olarak %93,6 oranında seyrek veri içeren MovieLens-100K veri seti kullanılmıştır. Otomatik kodlayıcı modelleri öge tabanlı öneri sistemlerinde kullanıcı tabanlı öneri sistemlerine göre daha başarılı sonuçlar sağladığı gözlemlenmiştir. Öge tabanlı öneri sistemlerde Temel Otomatik Kodlayıcı daha iyi performans sağlarken, kullanıcı tabanlı öneri sistemlerinde ise Temel Otomatik Kodlayıcı ve Seyrek Otomatik Kodlayıcı birbirlerine çok yakın bir performans sağladıkları gözlemlenmiştir.

Anahtar Kelimeler : Öneri Sistemleri, Derin Öğrenme, Otomatik Kodlayıcılar, Veri Seyrekliği

Abstract— Recommendation systems are systems that predict future preferences of users based on their past preferences. However, users may not always indicate their preferences to the systems. This causes data sparseness, which is one of the biggest problems when designing recommender systems. Autoencoders from deep learning algorithms solve the data sparsity problem by re-populating the sparse user matrix based on insights learned from the data. In this study, performances of deep learning algorithms were compared against data sparsity by using four different autoencoder models, namely Vanilla Autoencoder, Denoising Autoencoder, Sparse Autoencoder and Variational Autoencoder. The MovieLens-100K dataset, which contains 93.6% sparse data, was used as the data set. It has been observed that automatic encoder models provide more successful results in item-based recommendation systems than user based recommendation systems. It has been observed that Vanilla Autoencoder provides better performance in item-based recommendation systems, while Vanilla Autoencoder and Sparse Autoencoder provide very close performance in user-based recommendation systems.

Keywords : Recommendation Systems, Deep Learning, Autoencoders, Data Sparsity

1.Giriş

Son yıllarda, bilgi teknolojisi yeteneklerinin gelişmesiyle birlikte, kullanıcılar şirketlere büyük miktarda veri sağlamaktadır. Şirketler, bu veriler üzerinde istatistik ve makine öğrenimi yöntemlerini kullanarak satışları teşvik etmeyi amaçlar. Şirketlerin bu alanda en çok kullandığı yöntem öneri sistemleridir. Öneri sistemleri kullanıcıların

geçmiş tercih ve davranışlarına göre gelecekteki tercihlerini tahmin eden sistemlerdir. Sosyal medya uygulamaları ve e-ticaret şirketleri, kullanıcı deneyimini geliştirmek için öneri sistemlerini kullanır. Örneğin Netflix'te izlenen filmlerin yüzde 80'i ile Youtube'da izlenen videoların %60'ı öneri sistemleri sayesinde kullanıcılar tarafından tercih edilmiştir [1]. Amazon, kârının %35'ini öneri algoritmaları ile temin etmektedir [2].

Öneri sistemleri genellikle üç farklı yaklaşım kullanılarak geliştirilmiştir. Bu yaklaşımlar, işbirlikçi filtreleme, içerik tabanlı yaklaşım ve her iki yaklaşımı birleştiren hibrit öneri sistemleridir. İşbirlikçi filtreleme, kullanıcıların veya öğelerin tercihlerini benzer kullanıcılar veya öğelerle karşılaştırarak tahmin etme yöntemidir. İçerik temelli yaklaşım, benzer içeriğe sahip bir öğenin, öğe içeriğine göre tavsiye edilmesidir. Hibrit öneri sistemi ise, işbirlikçi filtreleme ile içerik temelli bir yaklaşımı birleştiren bir yöntemdir.

Bu sistemler, aşırı bilgi yükleme sorunuyla başa çıkmak için etkili yollar sağlasa da farklı bir zorlukla da karşı karşıyadır. Bu zorluk, veri seyrekliğidir. Veri seyrekliği sorunu, genellikle sistemdeki yetersiz bilgi nedeniyle ortaya çıkan ve işbirlikçi filtreleme yöntemlerini etkileyen bir sorundur. Bu sorun, veri tabanındaki kullanıcı tarafından derecelendirilen öğelerin toplam sayısının az olması nedeniyle kullanıcı öğe matrisinin seyrek bir matris olmasıyla sonuçlanır. Aynı zamanda başarılı komşular bulamamaya ve nihayetinde kötü bir tavsiye sürecine yol açar [3].

Öneri sistemi algoritmaları, geleneksel modelleri kullanarak seyrek matrisi yeniden oluşturmaya çalışır. [4,5,6] Ancak bu modeller doğrusal yaklaşımı benimser. Bu yaklaşım aşırı basitleştirilmiştir ve modelleme ifadelerini büyük ölçüde sınırlayacaktır. Ayrıca lineer yaklaşımlar karmaşık yapıları yakalamazlar. Derin öğrenme algoritmaları, seyrek matrisi doğrusal yaklaşımlara göre yeniden yapılandırma daha başarılıdır. [7,8] Derin öğrenme algoritmaları, aktivasyon fonksiyonlarını (sigmoid, relu vb.) kullanılarak lineer olmayan yapıları kolaylıkla modeller. Bu özellik, karmaşık etkileşim kalıplarıyla başa çıkmayı mümkün kılar ve kullanıcının tercihini tam olarak yansıtır [1].

Derin öğrenme algoritmaları öneri sistemlerindeki veri seyrekliğini problemini çözmekte diğer modellere göre oldukça başarılıdır. Öneri sistemlerinde derin öğrenme algoritmaları, heterojen veri kaynaklarından çıkarılan özniteliklerin öneri sürecine entegre edilmesiyle ele alınmaktadır. Özellikle Otomatik Kodlayıcılar veri seyrekliği sorunuyla başa çıkmak için kullanıcıların yararlı özelliklerini öğrenmek ve yeniden yapılandırmak amacıyla öneri sistemlerinde yaygın olarak kullanılmaktadır [9]. Sedhain [10] kullanıcı-öge kısmi vektörlerini girdi olarak alır ve bunları çıktı katmanında yeniden yapılandırmayı amaçlayan bir model önerir. Cao ve diğ.[11] orijinal seyrek kullanıcı ögesi matrislerinden yararlı düşük boyutlu özellikleri çıkarmak için denetimsiz bir derin öğrenme yöntemi olan gürültü giderme özelliğine sahip otomatik kodlayıcı kullanan yeni bir işbirlikçi filtreleme yöntemi önermektedir.

Otomatik kodlayıcılar kullanıcılar hakkındaki meta veriler üzerinden yararlı bilgileri öğrenerek, seyrek kullanıcı-öge matrisini yeniden oluşturur. Örneğin, He ve diğ.[12] ögenin hem içerik hem de etiket bilgilerini dikkate alan varyasyonel otomatik kodlayıcı adlı yeni bir derin öğrenme modeli önermiştir. Model, denetimsiz bir şekilde içerik bilgileri ve etiket bilgileri dahil olmak üzere ek yan bilgilerden etkili gizli temsilleri öğrenir. Grafik modellerin yardımıyla, kullanıcılar ve öğeler arasındaki örtük ilişkileri etkin bir şekilde çıkarır. Zhu ve diğ.[13] verilerin farklı özelliklerini yakalamak için kullanıcı tabanlı ve öge tabanlı özellik temsilleri öğrenebilen bir otomatik kodlayıcı modeli önermiştir. Model, öğelerin niteliklerini seyreklik sorununu hafifletmek için yarı otomatik kodlayıcı ile özellik temsillerinde birleştirilir.

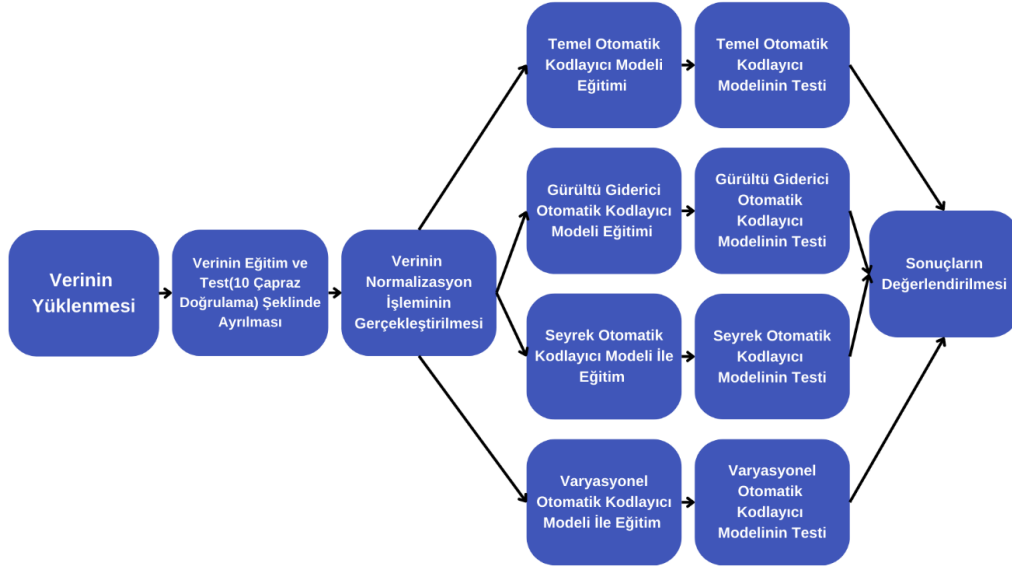
Meta veriler, öneri sistemlerinin doğruluğunu arttırmak için yaygın olarak kullanılsa da modellerin daha karmaşık olmasına neden olur. Ayrıca veri gizliliği nedeniyle her zaman meta verileri kullanmak mümkün olmaz. Otomatik kodlayıcılar sadece kullanıcı-öge matrisini kullanarak da etkili temsilleri öğrenir. Örneğin, Ferreira ve diğ.[7] seyrek kullanıcı-öge matrisindeki temsilleri öğrenen derin otomatik kodlayıcı modeli önermiştir. Zhang ve diğ.[14] derin seyrek otomatik kodlayıcı ve tekil değer ayrışımı (Singular Value Decomposition - SVD++) algoritmalarını kullanarak hibrit bir öneri sistemi modeli önermiştir. Model, derin seyrek otomatik kodlayıcı kullanarak kullanıcı-öge matrisindeki etkili temsilleri öğrenir. Öğrenilen özellik temsilleri, SVD++ algoritmasında öge gizli vektörünü değiştirmek için kullanır. Chen ve diğ.[15] eksik veri sorununu herhangi bir kullanıcı meta verisi kullanmadan çözmek için dikkat birimlerini kullanan bir gürültü giderici otomatik kodlayıcı önerir.

Çalışmamızda derin öğrenme algoritmalarından otomatik kodlayıcılar, seyrek derecelendirme matrisleri kullanılarak eğitilmiş ve performansları karşılaştırılmıştır. Çalışmada karşılaştırma yapmak için otomatik kodlayıcı modellerinden Temel Otomatik Kodlayıcı (Vanilla Autoencoder-AE), Gürültü Giderici Otomatik Kodlayıcı (Denoising Autoencoder-DAE), Seyrek Otomatik Kodlayıcı (Sparsity Autoencoder-SAE) ve Varyasyonel Otomatik Kodlayıcı (Variational Autoencoder-VAE) kullanılmıştır. Çalışmanın ikinci bölümünde önerilen yöntem, veri seti ve kullanılan otomatik kodlayıcı modellerinden bahsedilmiştir. Üçüncü bölümde otomatik kodlayıcılar kullanılarak elde edilen deneysel sonuçlara yer verilmiştir. Son bölümde tartışma ve gelecekte yapılacak çalışmalardan bahsedilmiştir.

2.Yöntem

2.1. Önerilen Yöntem

Bu çalışmada otomatik kodlayıcıların veri seyrekliğine karşı gösterdikleri performansın karşılaştırılması amaçlanmıştır. Bu amaçla kullanılan 4 farklı otomatik kodlayıcı modelleri Movielens-100K veri seti kullanılarak eğitim gerçekleştirilmiştir. Her bir modelin eğitim sonuçları çeşitli hata metriklerine göre karşılaştırılmıştır. Çalışmada önerilen yöntem Şekil 1’te sunulmuştur.



Şekil 1. Önerilen Yöntem

Önerilen yöntemde öncelikle veri setini 10 çapraz doğrulama kullanarak eğitim ve test veri seti olarak iki bölüme ayrılması gerçekleştirildi. İkinci adımda verileri otomatik kodlayıcı modellerinde eğitebilmek için veri normalizasyonu işlemi gerçekleştirildi. Burada min-max normalizasyon yöntemi kullanıldı. Üçüncü adımda hazırlanan veri seti otomatik kodlayıcı modellerinde eğitim işlemi gerçekleştirildi. Dördüncü bölümde ise test işlemleri gerçekleştirilerek, başarı sonuçları değerlendirilmiştir.

2.2. Veri Seti

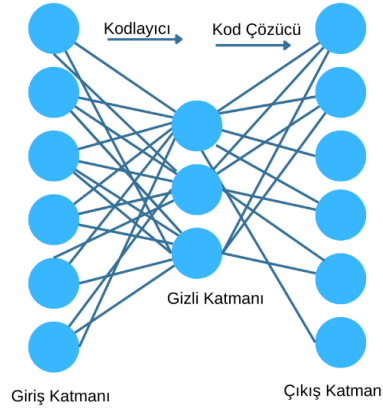
Bu çalışmada modellerin performanslarını karşılaştırmak için MovieLens veri seti kullanılmıştır. MovieLens, Minnesota Üniversitesi Bilgisayar Bilimi ve Mühendisliği Bölümü’nde bulunan GroupLens araştırma laboratuvarı tarafından hazırlanan bir kullanıcı-film derecelendirmeleri bulunan bir veri setidir. Bu veri seti içerisinde kullanıcı-film derecelendirmeleri(puan ataması) ile film isimleri, film türleri ve kullanıcı yan bilgileri de bulunmaktadır. Bu çalışmada veri olarak sadece kullanıcı-film derecelendirmeleri kullanılmıştır. MovieLens projesi içerisinde birçok boyutta derecelendirilmiş veri setleri bulunmaktadır. Bu çalışmada bu veri setleri arasından MovieLens-100K veri seti kullanılmıştır. MovieLens-100K içerisinde 100.000 adet derecelendirilmiş veri bulunan bir veri setidir. Bu veri seti 943 kullanıcının 1682 adet filmi puanlaması ile oluşturulmuştur. Puanlama sistemi 1-5 arasında yapılmıştır. MovieLens-100K veri seti %93,6 oranında derecelendirilmemiş veri içeren çok seyrek bir veri setidir.

2.3. Otomatik Kodlayıcı Modelleri

Otomatik kodlayıcı, gizli katmanda elde edilen temsil gösterimini kullanarak çıktı katmanı üzerinde giriş katmanını yeniden oluşturan ileri beslemeli bir sinir ağıdır. Bir otomatik kodlayıcı giriş katmanı, gizli katman ve çıktı katmanı olmak üzere üç katmandan oluşur. Giriş katmanındaki nöron sayısı, çıkış katmanındaki nöron sayısına eşittir. Öğrenme süreci sırasında otomatik kodlayıcı, kodlayıcı ve kod çözücü olarak adlandırılan iki eşleme kullanır. Kodlayıcı, verileri giriş katmanından gizli katmana eşlerken, kod çözücü, kodlanmış verileri gizli katmandan çıktı katmanına eşler [9]. Bu çalışmada 4 farklı Otomatik Kodlayıcıların performanslarını karşılaştırılmıştır. Kullanılan modellerin detayları aşağıda sunulmuştur.

2.3.1. Temel Otomatik Kodlayıcılar

Temel Otomatik Kodlayıcılar, otomatik kodlayıcı modellerin en temel halini oluşturur. Gizli katmandaki nöron sayısı giriş ve çıkış katmanındaki nöron sayısından daima küçük olmalıdır. Buradaki amaç veriyi daha az boyuta düşürerek önemli iç görüleri öğrenmek ve seyrek kullanıcı-öge matrisini yeniden yapılandırmaktır. Şekil 2’de Temel Otomatik Kodlayıcı modelinin yapısı gösterilmiştir.



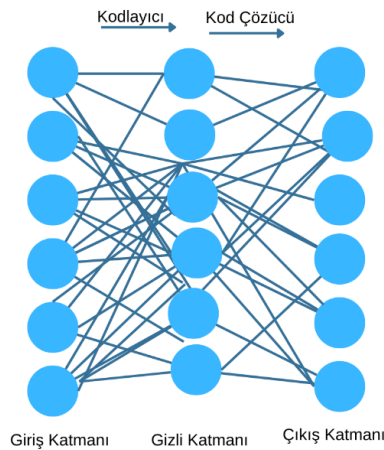
Şekil 2. Temel Otomatik Kodlayıcı Yapısı

2.3.2. Gürültü Giderici Otomatik Kodlayıcılar

Gürültü Giderici Otomatik Kodlayıcılar, gürültü içeren verilerden önemli bilgileri öğrenen bir otomatik kodlayıcı türüdür. Yapısı Temel Otomatik Kodlayıcı ile aynıdır. Veri eğitilmeden önce veri üzerine sentetik gürültü eklenir. Eğitim hataları gürültüsüz veri ve otomatik kodlayıcının oluşturduğu veri kullanılarak hesaplanır. Böylelikle otomatik kodlayıcı hem ilgili iç görüleri öğrenirken, aynı zamanda gürültü gidermeyi de öğrenir.

2.3.3 Seyrek Otomatik Kodlayıcılar

Seyrek Otomatik Kodlayıcılar, gizli katmana düğümlerin sadece bir kısmını aktif etmek için ceza terimi (seyreklik kısıtlaması) olarak adlandırılan bir kısıt eklenerek tasarlanan bir otomatik kodlayıcı modelidir. Eklenen ceza terimi, gizli katmanın giriş katmanındaki nöron sayısına eşit ya da daha fazla nöron sayısına sahip olacak şekilde tasarlanmasını ve otomatik kodlayıcının daha yararlı bilgileri öğrenmesini sağlar. Şekil 3’de Seyrek Otomatik Kodlayıcı modelinin yapısı gösterilmiştir.

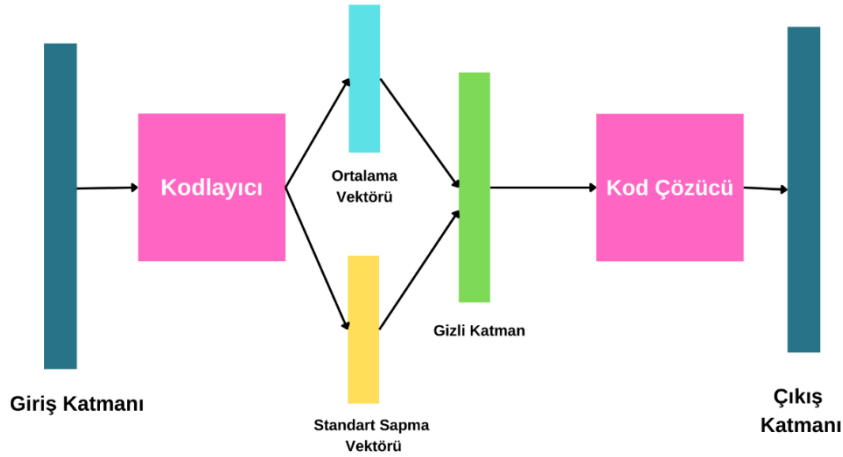


Şekil 3. Seyrek Otomatik Kodlayıcı Yapısı

Ceza terimini oluşturmak için Seyrek Otomatik Kodlayıcılarda farklı yöntemler kullanılmaktadır. Bu çalışmada tasarlanan Seyrek Otomatik Kodlayıcı için L1 Düzenlemesi kullanılmıştır. L1 Düzenlemesi, katsayıların mutlak büyüklük değerini ceza terimi olarak ekleyerek, ceza katsayısını sıfıra düşürme eğilimi gösterir.

2.3.4. Varyasyonel Otomatik Kodlayıcılar

Varyasyonel Otomatik Kodlayıcılar, gizli katmandaki nöronların dağılımını normal dağılıma göre dağılmaya zorlayan bir otomatik kodlayıcı modelidir. Varyasyonel Otomatik Kodlayıcılar, giriş katmanındaki veriyi, kodlayıcı yardımıyla ortalama ve standart sapma şeklinde iki ayrı vektöre ayırır ve bu vektörleri kullanarak gizli katman içerisinde giriş katmanındaki verilere benzer rastgele şekilde yeni veriler üretir. Varyasyonel Otomatik Kodlayıcılar her bir gizli katmanda yer alan özellik için bir dağılım tanımlar. Bu özelliği ile Varyasyonel Otomatik Kodlayıcılar diğer otomatik kodlayıcı modellerinden ayrılır. Şekil 4'te Varyasyonel Otomatik Kodlayıcı modelinin yapısı gösterilmiştir.



Şekil 4. Varyasyonel Otomatik Kodlayıcı Yapısı

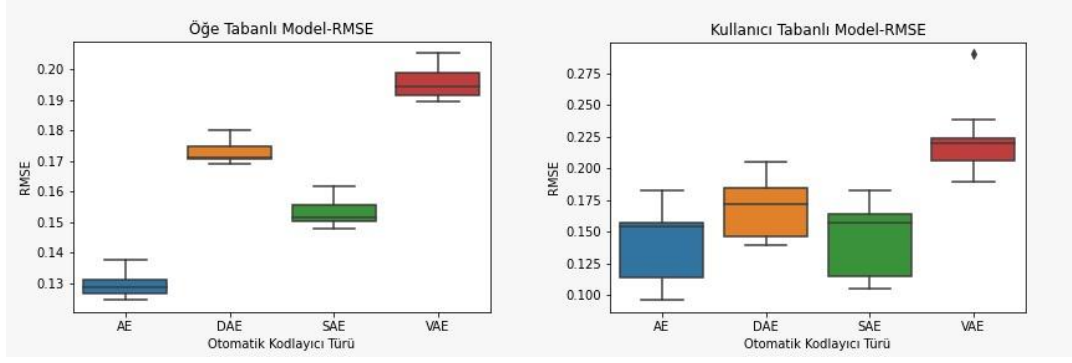
3. Sonuçlar

Bu çalışmada Temel Otomatik Kodlayıcı (Vanilla Autoencoder-AE), Gürültü Giderici Otomatik Kodlayıcı (Denosing Autoencoder-DAE), Seyrek Otomatik Kodlayıcı (Sparsity Autoencoder-SAE) ve Varyasyonel Otomatik Kodlayıcı (Variational Autoencoder-VAE) olmak üzere 4 farklı model kullanılmıştır. Her bir model 10 kat çapraz doğrulama tekniği ile eğitilerek performans sonuçları elde edilmiştir. Performansları değerlendirmek için öneri sistemlerini karşılaştırmada en çok faydalanılan metriklerden Kök Ortalama Kare Hatası (Root Mean Square Error-RMSE) ve Ortalama Mutlak Hata (Mean Absolute Error-MAE) metrikleri kullanılmıştır. RMSE ve MAE sırasıyla Denklem 1 ve Denklem 2'de gösterildiği gibi hesaplanır:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

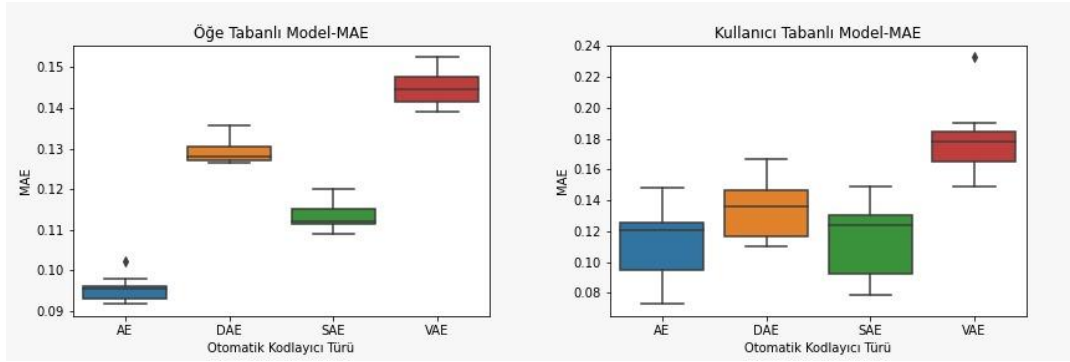
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Dört farklı otomatik kodlayıcı kullanılarak tahmin edilen kullanıcı- film puan tahminlerine ait hata değerleri Şekil 5 ve Şekil 6'da kutu grafiği şeklinde gösterilmiş olup, ortalamaları Tablo 1'de gösterilmiştir. Bu çalışma sürecinde tüm modeller Google Colaboratory ortamında, 2.4 GHz Intel i5-6200U işlemciye ve 12 GB RAM belleğine sahip olan sistem üzerinden elde edilmiştir. Çalışmada Python programlama dili kullanılmıştır. Modellerin geliştirilmesinde Python içerisinde bulunan Keras kütüphanesi kullanılmıştır.



Şekil 5. Otomatik Kodlayıcı Modellerinin Karşılaştırılması (RMSE)

Şekil 5'te RMSE metriğinin karşılaştırmalarında görüldüğü gibi veriler öge tabanlı modelde daha homojen dağılmıştır. Öge tabanlı öneri sisteminde Gürültü Giderici Otomatik Kodlayıcı (DAE) ile Seyrek Otomatik Kodlayıcı (SAE) medyan değerleri birinci çeyrek değerine daha yakın değerdedir. Kullanıcı tabanlı öneri sisteminde ise genel olarak medyan değerlerinin üçüncü çeyrek değerlerine yakın olduğu gözlemlenmiştir. Ayrıca Varyasyonel Otomatik Kodlayıcı kullanılarak geliştirilen kullanıcı tabanlı öneri sisteminde ayırık değerler gözlemlenmiştir.



Şekil 6. Otomatik Kodlayıcı Modellerinin Karşılaştırılması (MAE)

Şekil 6'da MAE metriğinin karşılaştırmalarında görüldüğü gibi veriler öge tabanlı modelde daha homojen dağılmıştır. Öge tabanlı öneri sisteminde Gürültü Giderici Otomatik Kodlayıcı (DAE) ile Seyrek Otomatik Kodlayıcı (SAE) medyan değerleri birinci çeyrek değerine daha yakın iken Temel Otomatik Kodlayıcının medyan değerinin üçüncü çeyrek değerine daha yakın olduğu ve ayırık değerler içerdigi gözlemlenmiştir. Kullanıcı tabanlı öneri sisteminde ise genel olarak medyan değerlerinin üçüncü çeyrek değerlerine yakın olduğu gözlemlenmiştir. Ayrıca Varyasyonel Otomatik Kodlayıcı kullanılarak geliştirilen kullanıcı tabanlı öneri sisteminde ayırık değerler gözlemlenmiştir.

Tablo 1. Öge Tabanlı Modellerin Metriklerinin Karşılaştırılması

Model Adı	Öge Tabanlı		Kullanıcı Tabanlı	
	RMSE	MAE	RMSE	MAE
AE	0,1295	0,0954	0,13981	0,11126
DAE	0,1728	0,1292	0,16794	0,13360
SAE	0,1531	0,1133	0,14557	0,11617
VAE	0,1954	0,1450	0,22186	0,17875

Tablo 1’de görülen modellerin RMSE ve MAE metriklerini incelendiğinde öge tabanlı olarak geliştirilen Temel Otomatik Kodlayıcı’nın diğer modeller ve öneri sistemi yöntemleri ile karşılaştırıldığında en başarılı model olduğu gözlemlenmiştir. Metrik sonuçları incelendiğinde öge tabanlı modellerin kullanıcı tabanlı modellerden daha başarılı sonuçlar verdiği gözlemlenmiştir.

4. Tartışma

Bu çalışmada sadece kullanıcı-öge seyrek matrisi kullanılarak eğitilen otomatik kodlayıcı modellerinin veri seyrekliğine karşı gösterdiği performanslar karşılaştırılmıştır. Öneri sistemlerinde seyrek matrisi yeniden yapılandırmak için genellikle kullanıcı yan bilgilerinden faydalanılır. Bu da modelleri daha karmaşık hale getirir. Ayrıca veri gizliliği nedeniyle her zaman kullanıcı yan bilgilerini kullanmak mümkün olmaz. Otomatik kodlayıcılar kullanıcı yan bilgilerini kullanmaya ihtiyaç duymadan sadece kullanıcı-öge matrisinden etkili temsilleri öğrenerek seyrek derecelendirme matrisini yeniden yapılandırır. Yapılan deneylerde öge tabanlı modellerde otomatik kodlayıcıların seyrek matristeki özellik temsillerini öğrenmede daha başarılı olduğu gözlemlenmiştir. Otomatik kodlayıcılar öğelerin birbiri arasındaki ilişkileri özellik temsillerine daha başarılı şekilde dönüştürmüştür.

Çalışmamızda hedefimiz farklı nöron sayıları ve mimarilerle oluşturulmuş otomatik kodlayıcıların seyrek verilerden oluşan öneri sistemleri verisi üzerinde başarılarını değerlendirmektir. Bu alanda regresyon tabanlı, kümeleme tabanlı işbirlikçi yaklaşımlarından öte verileri anlayan ve modelleyen mimarilerin seyrek veriler üzerindeki başarıları göz önüne alınarak çalışmamız gerçekleştirilmiştir.

Gelecek çalışma için Kısıtlı Boltzmann Makineleri (Restricted Boltzmann Machine-RBM) ve Üretici Çekişmeli Ağlar (Generative Adversarial Networks- GAN) gibi daha kompleks yapıları da kullanarak veri seyrekliği problemlerine karşı yaklaşımlarının incelenmesi planlanmaktadır.

5. Kaynaklar

- [1] 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), 1-38.
- [2] Dong, M., Yuan, F., Yao, L., Wang, X., Xu, X., & Zhu, L. (2022). A survey for trust-aware recommender systems: A deep learning perspective. *Knowledge-Based Systems*, 249, 108954.
- [3] Da’u, A., & Salim, N. (2020). Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, 53(4), 2709-2748.
- [4] Anwar, T., Uma, V., & Srivastava, G. (2021). Rec-cfsvd++: Implementing recommendation system using collaborative filtering and singular value decomposition (svd)++. *International Journal of Information Technology & Decision Making*, 20(04), 1075-1093.
- [5] Joorabloo, N., Jalili, M., & Ren, Y. (2020). Improved collaborative filtering recommendation through similarity prediction. *IEEE Access*, 8, 202122-202132.
- [6] Karpus, A., Raczynska, M., & Przybyłek, A. (2019). Things You Might Not Know about the k-Nearest Neighbors Algorithm. In *KDIR* (pp. 539-547).
- [7] Ferreira, D., Silva, S., Abelha, A., & Machado, J. (2020). Recommendation system using autoencoders. *Applied Sciences*, 10(16), 5510.
- [8] Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S., & Xiong, N. (2019). Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics*, 15(8), 4591-4601.
- [9] Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1), 1-37.
- [10] Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015, May). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web* (pp. 111-112).
- [11] Cao, S., Yang, N., & Liu, Z. (2017, May). Online news recommender based on stacked auto-encoder. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 721-726). IEEE.
- [12] He, M., Meng, Q., & Zhang, S. (2019). Collaborative additional variational autoencoder for top-N recommender systems. *IEEE Access*, 7, 5707-5713.

- [13] Zhu, Y., Wu, X., Qiang, J., Yuan, Y., & Li, Y. (2021). Representation learning with collaborative autoencoder for personalized recommendation. *Expert Systems with Applications*, 186, 115825.
- [14] Zhang, Y., Zhao, C., Chen, M., & Yuan, M. (2021). Integrating stacked sparse auto-encoder into matrix factorization for rating prediction. *IEEE Access*, 9, 17641-17648.
- [15] Chen, S., & Wu, M. (2020). Attention collaborative autoencoder for explicit recommender systems. *Electronics*, 9(10), 1716.