



# Kullanıcı ve Öğe Bazlı, Geniş ve Derin Öğrenme Tabanlı Seyahat Öneri Sistemi

Alihan Öz<sup>1\*</sup>, Meryem Uzun-Per<sup>2,3</sup>, Mert Bal<sup>4</sup>

<sup>1</sup>Yıldız Teknik Üniversitesi, Kimya ve Metalurji Fakültesi, Matematik Mühendisliği Bölümü, İstanbul, Türkiye, (ORCID: 0009-0000-7934-6335), alihanozmm@gmail.com

<sup>2</sup>İstanbul Sağlık ve Teknoloji Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID:0000-0002-4958-4575), meryem.uzunper@istun.edu.tr

<sup>3</sup>Technoly İstanbul A.Ş, AR-GE Departmanı, İstanbul, Türkiye (ORCID:0000-0002-4958-4575), meryem.uzunper@petour.com

<sup>4</sup>Yıldız Teknik Üniversitesi, Kimya ve Metalurji Fakültesi, Matematik Mühendisliği Bölümü, İstanbul, Türkiye, (ORCID: 0000-0001-6250-929X), mertbal@yildiz.edu.tr

(İlk Geliş Tarihi 12 Mayıs 2023 ve Kabul Tarihi 25 Ağustos 2023)  
(DOI: 10.31590/ejosat.1296379)

**ATIF/REFERENCE:** Öz, A., Uzun-Per, M & Bal, M. (2023). Kullanıcı ve Öğe Bazlı, Geniş ve Derin Öğrenme Tabanlı Seyahat Öneri Sistemi. *Avrupa Bilim ve Teknoloji Dergisi*, (51), 334-351.

## Öz

Teknolojinin gelişmesi ile birlikte artan dijital bilgi miktarı ve internetin yaygınlaşması ile internet üzerinden ürün, hizmet, abonelik gibi ticaret işlemlerinin gerçekleştiği web sitelerinin sayısının da artması, beraberinde, müşterilere kişiselleştirilmiş ve doğru; ürün, hizmet ve abonelikleri sunmanın (önermenin) de önemini artmıştır. Müşterilere önerilerde yaygın olarak kullanılan ürün bazlı, kullanıcı tabanlı ve bu ikisinin birlikte kullanıldığı hibrit geleneksel yaklaşımlar çoğu çalışmada kullanılmaktadır. Geleneksel yaklaşımların, büyük ve seyrek veriler ile çalışma, kullanıcı ve ürün arasındaki karışık ilişkileri bulamama ve soğuk başlangıç (cold start) gibi problemlerinin üstesinden gelmek, derin ve geniş öğrenme sistemlerinin kullanımı ile mümkün olmuştur.

Bu çalışma kapsamında, öncelikle derin ve geniş sinir ağlarına ve bunların seyahat öneri sistemlerindeki uygulamalarına kapsamlı bir bakış açısı sunulmuş ve en popüler öneri algoritmaları olan Google'ın Geniş ve Derin Algoritması ve Facebook'un Deep Learning Recommendation Model (DLRM) algoritmasına yer verilmiştir. Ardından, geniş ve derin öğrenme yaklaşımı ile kullanıcı ve ürün özelliklerinin kategorik olanlarının gömme işlemi uygulanarak, nümerik veriler ile modele beslendiği yeni bir seyahat öneri sistemi oluşturulmuştur. Önerilen yöntem gerçek bir seyahat acentesi şirketinin veri seti üzerinde uygulanmıştır. Sonuçta, kullanıcılara verilen en iyi beş öneride, %82.37 doğruluk oranı yakalanmıştır.

**Anahtar Kelimeler:** öneri sistemleri, derin öğrenme, derin ve geniş öneri sistemleri, içerik tabanlı, işbirlikçi, hibrit, seyahat öneri sistemleri.

## A User and Item-Based, Wide and Deep Learning Based Travel Recommendation System

### Abstract

With the development of technology, the increasing amount of digital information and the widespread use of the internet, and the increase in the number of websites where trade transactions such as products, services and subscriptions are carried out on the internet, along with it, personalized and accurate; The importance of recommending products, services and subscriptions has also increased. Product-based, user-based, and hybrid traditional approaches, which are widely used in recommendations to customers, are used in most studies. Overcoming the problems of traditional approaches such as working with large and sparse data, inability to find complex relationships between user and product, and cold start has been possible with the use of deep and wide learning systems.

\* Sorumlu Yazar: [alihanozmm@gmail.com](mailto:alihanozmm@gmail.com)

Within the scope of this study, first, a comprehensive view of deep and wide neural networks and their applications in travel recommendation systems is presented. After giving the basic information about deep learning, Google's Wide and Deep Algorithm and Facebook's Deep Learning Recommendation Model (DLRM) algorithm, which are the most popular recommendation algorithms, are included. Then, a new travel recommendation system was created in which numerical data is fed to the model by applying the embedding process of categorical user and product features with a broad and deep learning approach. The proposed method was applied on the data set of a real travel agency company. As a result of this study, 82.37% accuracy rate was achieved in the top five recommendations given to the users.

**Keywords:** recommendation systems, deep learning, deep and wide recommendation systems, content-based, collaborative, hybrid, travel recommendation systems.

## 1. Giriş

Günümüzde, seyahat öneri sistemleri, kullanıcılara, tercihlerine ve geçmiş davranışlarına göre kişiselleştirilmiş seyahat önerileri sağlayan turizm endüstrisinde ayrılmaz bir bileşen haline gelmiştir. Birçok popüler internet sitesi, trafiklerini ve dolayısıyla kazançlarını arttırabilmek için, istatistiksel ve makine öğrenmesi yöntemlerini içeren öneri (tavsiye) sistemlerini kullanmaktadır. Bu sistemler sadece şirketlere dolaylı maddi katkı sağlamakla kalmayıp müşteri memnuniyeti ve bağlılığını, kullanıcılara zaman tasarrufu ve karmaşıklığı azaltarak sunması itibarıyla kazanarak ortak yararlı bir ekosistem oluşmasını sağlamışlardır. Bu sistemlerin etkinliği, verilerin kalitesi ve öneri üretimi için kullanılan algoritmalar dahil olmak üzere çeşitli faktörlere bağlıdır.

Son yıllarda yapay zeka ve makine öğrenmesindeki gelişmeler, daha sofistike seyahat öneri sistemlerinin geliştirilmesinin yolunu açmıştır. Derin ve geniş sinir ağları, çeşitli uygulamalarda umut verici sonuçlar veren makine öğrenmesi alanındaki en son yeniliklerden biridir [1]. Bu ağlar, seyahat öneri sistemleri gibi karmaşık sorunları çözmek için güçlü bir araç sağlayan derin sinir ağları ile geniş sinir ağlarının bir kombinasyonunu kullanmaktadır ve bu yolla kullanıcılar ve öğeler arasındaki doğrusal olmayan ilişkileri etkili bir şekilde modelleyebilmektedirler. Ayrıca gömme (embedding) ve karma (hashing) gibi teknikler kullanarak büyük ölçekli ve seyrek verileri verimli bir şekilde işleyebilirler. Ek olarak Deep Learning Recommendation Model (DLRM) [2], kullanıcı ve ürünlere ait yardımcı özellikleri kullanarak, öneri doğruluğunu iyileştirip soğuk başlatma sorununun üstesinden gelmektedir.

Bu çalışma kapsamında, kullanıcı ve ürün özelliklerinin kategorik olanlarına ayrı ayrı gömme işlemi uygulanarak, nümerik ve kategorik verilerin ürün ve kullanıcı bazlı verilerle ayrı ayrı beslenen bir geniş ve derin öğrenme algoritması önerilmiştir. Önerilen bu yöntem gerçek bir seyahat acentesi şirketinin veri setine uygulanmıştır ve geniş ve derin öğrenme algoritmasına, faydalarına ve seyahat öneri sistemlerindeki performanslarına ilişkin değerlendirmelere bir bakış sunmaktadır.

Bu çalışma, seyahat öneri sistemlerinde derin ve geniş sinir ağlarının kullanımına ilişkin artan araştırma kapsamına katkıda bulunmakta ve bu alanda daha fazla ilerleme kaydedilmesi potansiyelini vurgulamaktadır. Seyahat öneri sistemlerinde derin ve geniş sinir ağlarının kullanımı büyük bir potansiyele sahiptir ve bu makale, bu alanla ilgilenen araştırmacılar, uygulayıcılar ve öğrenciler için değerli bir kaynak sunmaktadır.

Literatür özeti bölümünde, öneri sistemlerinin ne olduğu, kullanım alanları ve öneri sistemleri üzerine yapılan çalışmalar hakkında bilgi verilmektedir. Ardından, yapay sinir ağları ve derin öğrenmenin temelleri hakkında ayrıntılı bilgiler verilmiştir. Öneri sistemleri süreçleri ve seyahat öneri sistemleri hakkında bilgi verilmiştir. Ayrıca, yapay sinir ağlarının öneri sistemlerine katkıları ele alınmıştır.

Makalenin ana odak noktası, derin öğrenme tabanlı bir seyahat öneri sistemi olan Bölüm 3 ve Bölüm 4'tür. Bölüm 3'te önerilen yöntem açıklanmış, Bölüm 4'te ise veri kümesi incelenmiş, veri ön işleme ve model öncesi hazırlık aşamaları ile model uygulama aşamalarına yer verilmiştir. Ardından deneysel sonuçlar ve değerlendirmeler sunulmuştur. Son olarak, 5. bölümde, çalışma özetlenmiştir.

## 2. Literatür Özeti

Öneri sistemleri, tercihlerimize ve ilgi alanlarımıza göre uyarlanmış öğeler, hizmetler veya içerik önererek bilinçli kararlar vermemize yardımcı olmak için tasarlanmıştır. Son yıllarda çeşitli endüstrilerde giderek daha önemli hale gelen öneri sistemlerinin geliştirilmesinde de buna paralel önemli bir artış olmuştur.

Öneri sistemlerinin tarihini, gelişimini ve ilerlemelerini derinlemesine inceleyen Fan, W. (2022) gibi akademisyenler ve araştırmacılar güvenilir öneri sistemlerinin önemine dikkat çekmişlerdir [3]. Öneri sistemlerinin e-ticaret üzerindeki etkisi, Karimova'nın çalışmasında açıkça görülmektedir [4]. Ayrıca, öneri sistemlerinin kullanıcı deneyimi ve iş büyümesine etkisi 2012'de Belluf v.d. [5] ve 2015'te Amatriain v.d. [6] tarafından kanıtlanmıştır.

İşbirlikçi filtreleme (collaborative filtering) (CF) ve içeriğe dayalı öneri (content-based recommendation) (CBR), seyahat öneri sistemlerinde en yaygın kullanılan yaklaşımlardan ikisidir. CF, geçmişte tercihlerinde aynı fikirde olan kişilerin gelecekte muhtemelen tekrar aynı fikirde olacakları fikrine dayanırken, CBR, özelliklerin kullanıcının tercih ettiği öğelerle benzerliğine dayalı olarak öğeler önerir.

Koren v.d., öneri sisteminin doğruluğunu artırmak için gizli faktör modellerini komşuluk modelleri ile birleştiren çok yönlü bir işbirlikçi filtreleme modeli önermiştir [7]. Rendle, öneri sistemlerinin ölçeklenebilirliğini ve doğruluğunu iyileştirmek için çarpanlara ayırma ve regresyon tekniklerini birleştiren bir model olan çarpanlara ayırma makinelerini (factorization machines) önermiştir [8].

Paparizos v.d. (2011), kullanıcılara iş tanımlarına göre iş önermek için içeriğe dayalı filtrelemeyi kullanan, makine tarafından öğrenilen bir iş öneri sistemi geliştirmiştir [9]. Al-Otaibi v.d. (2012), iş öneri sistemleri üzerine bir araştırma yapmış ve içerik tabanlı filtrelemenin iş öneri sistemlerinde kullanılan en popüler teknik olduğunu bulmuştur [10].

Her ne kadar CF ve CBF tekniklerinin güçlü yönleri olsa da her iki tekniğin de soğuk başlatma ve veri seyrekliği sorunları gibi sınırlamaları da vardır. Bu sınırlamaları minimize etmek için CF ve CBR tekniklerini birleştiren hibrit seyahat öneri sistemleri, Albadvi v.d. [11] ve Balabanovi [12] gibi birkaç makale tarafından önerilmiştir. Aynı şekilde, Pizzato v.d., kullanıcıları birbirine önermek için her iki tekniği birleştiren çevrimiçi flört için karşılıklı bir öneri sistemi geliştirmiştir [13].

Son yıllarda, derin öğrenme teknolojisi, veri madenciliği ve makine öğrenmesi topluluğunda sıcak bir konu olarak ortaya çıkan kayda değer bir ilerleme kaydetti [14]. Başlangıçta görüntü ve konuşma tanıma ve doğal dil işleme için geliştirilen derin öğrenme, bu alanlarda dikkate değer sonuçlar elde etti [15].

Derin öğrenmeye dayalı öneri sistemlerindeki (DLRS) son gelişmeler, geleneksel modellerin sınırlamalarını aşarak ve yüksek öneri kalitesine ulaşarak büyük ilgi görmesini sağladı. Derin öğrenme, doğrusal olmayan ve önemsiz olmayan kullanıcı-öge ilişkilerini etkin bir şekilde yakalama ve daha yüksek katmanlarda veri temsilleri yaparak daha karmaşık soyutlamaların kodlanmasını sağlama yeteneğine sahiptir [16-27]. Ayrıca, bağlamsal, metinsel ve görsel bilgiler gibi bol miktarda erişilebilir veri kaynaklarından verilerin kendi içindeki karmaşık ilişkilerini yakalayabilir.

Wang v.d. (2017) öğelerle kullanıcı etkileşimlerini birden çok ayrıntı düzeyinde yakalayan ve kullanıcı tercihlerine ve öge özelliklerine dayalı olarak kişiselleştirilmiş önerilerde bulunan hiyerarşik bir model olan Dionysius'u önermiştir [28]. Zhang v.d. (2019), derin öğrenmeye dayalı öneri sistemleri üzerine bir çalışma yürütmüş ve hem CF hem de CBR tekniklerini birleştiren hibrit öneri sistemlerinin öneri doğruluğunu iyileştirmede daha etkili olduğunu göstermiştir [29]. Wang içerik tabanlı ve işbirlikçi filtreleme tekniklerini birleştiren ve seyahat önerilerinin kalitesini artıran hibrit bir seyahat önerisi sistemi sunmaktadır [30]. Muhammet Ç. v.d. (2020), kullanıcı ve öge ID yerleştirmeleri ile yardımcı bilgilerden oluşan bir derin sinir mimari ağı geliştirmiştir [31]. Aynı zamanda, Covington v.d. (2016) yardımcı bilgileri modellemek için popüler bir CBR yaklaşımı olan derin sinir ağlarını benimsemiştir [32].

Son olarak, Heng-Tze Cheng v.d. (2016), görüntüleme geçmişlerine ve diğer ilgili özelliklere dayalı olarak YouTube kullanıcılarına kişiselleştirilmiş video önerileri yapabilen geniş ve derin bir sinir ağı modeli önermiştir [1] ve Maxim Naumov v.d. (2019) büyük ölçekli uygulamalarda yüksek performans gösteren derin öğrenme öneri makinesi geliştirmiştir [2].

İlk alt bölümde, çalışmayı anlatmaya yardımcı olacak temel kavramlar ile yapay sinir ağları ve derin öğrenme yaklaşımı açıklanmıştır. Sonraki alt bölümde ise bu algoritmaların seyahat endüstrisindeki uygulamalarına yer verilmiştir.

## 2.1. Yapay Sinir Ağları ve Derin Öğrenmenin Çalışma Prensi ve Mimarisi

Yapay sinir ağları (artificial neural networks), hayvan beynini oluşturan biyolojik sinir ağlarından esinlenilerek verilerin sinir ağları içinde hareket etmesi prensibi ile oluşturulan istatistiksel denetimli bir makine öğrenmesi yöntemidir.

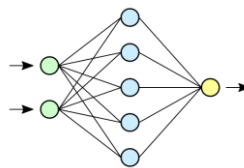
Derin öğrenme (Deep Learning) ise, girdi ve çıktılar dahil olmak üzere üç veya daha fazla katmandan oluşan yapay sinir ağlarından oluşan makine öğrenmelerine verilen isimdir. İnsan beyni gibi çalışmasından ötürü büyük verilerdeki bilgiyi öğrenmeye olanak sağlamaktadır.

Derin öğrenme mimarisi birkaç katmandan oluşur; **girdi katmanı**, **gizli katmanlar**, **çıkı katmanı** [33].

**Girdi katmanı** (input layer), eğitim kümesindeki özellikler kadar nöronlara sahiptir ve sinir ağının öğrendiği verileri alan katmandır [34].

**Çıkı katmanı** (output layer), sinir ağının sonunda yer alan ve model çıktısını üreten katmandır. Bu katmandaki girdiler, çıkı katmanındaki nöronların değerlerini temsil eder [35].

Bir tahmin vektörü y elde etmek için, ağı giriş ve çıkış katmanları arasındaki katmanlarda gerçekleştirdiği belirli matematiksel işlemleri gerçekleştirildiği katmanlara **gizli katmanlar** denir [36]. Modelin derinliğine ve istenen sonuçlara bağlı olarak girdi katmanı sayısı yani modelin derinliği ve her bir girdi katmanındaki girdi sayısı değişiklik gösterebilir.



Şekil 1. İki Değişkenli, Bir Gizli Katmanlı, İkili Çıkı Üreten Basit Bir Sinir Ağı  
Bir gizli katmanlı basit bir sinir ağı için matematiksel formül şöyledir:

$$y = g_2(w_2 * g_1(w_1 * x + b_1) + b_2) \quad (1)$$

İfade (1)'de,  $x$  girdi vektörünü,  $y$  bağımlı değişkeni,  $w_1$  ve  $w_2$  girdi ve gizli katmanların ağırlık matrislerini,  $b_1$  ve  $b_2$  yanlılık veya sapma vektörlerini,  $g_1$  ve  $g_2$  gizli ve çıktı katmanlarının aktivasyon fonksiyonlarını ifade etmektedir.

Her iki katman arasında bağlantıyı sağlayan sayısal gösterimlere **ağırlık** ismi verilir [37]. Bir ağırlık matrisinin boyutları, bu ağırlık matrisi ile birbirine bağlanan iki katmanın boyutlarından kaynaklanır. Ağırlıklar ağırlık matrisleri  $W$  ile gösterilir.  $W$  ağırlık matrisindeki her bir  $W_{ij}$  elemanı bir ağırlığa sahiptir. Burada  $i$  verinin geldiği katmanın sayısını,  $j$  ise verinin aktarıldığı katmanın sayısını ifade etmektedir.

Aynı zamanda modelde yanlılığı azaltmak adına nöronların ağırlıklarla çarpımlarının sonucunda çıktı değişkenini oluşturmak üzere doğrusal bir bileşen eklenir. Bu bileşene **yanlılık** (bias) **bileşeni** denir ve  $b_j$  ile ifade edilir [33].

İki girdiden oluşan  $X$  girdi vektörünün verildiği ve üç çıktının tahminlendiği basit bir modelin denklemi, ifade (2) ve (3)'teki gibidir.

$$X = (X_1, X_2) \quad (2)$$

$$W = (w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}) \quad (3)$$

$X$  girdi matrisi (2) ile,  $W$  ağırlık matrisi ise (3) ile ifade edilecektir. Şu ana kadarki bilgiler ışığında  $y$  bağımlı değişkeninin şu şekilde oluştuğu söylenebilir:

$$y = x_1 w_{11} + b_{11} + x_2 w_{21} + b_{21}, x_1 w_{12} + b_{12} + x_2 w_{22} + b_{22}, x_1 w_{13} + b_{13} + x_2 w_{23} + b_{23} \quad (4)$$

$$y = (z_1, z_2, z_3) \quad (5)$$

Oluşan  $z$  vektörü her ne kadar  $y$ 'yi ifade etmede kullanılabilir olsa da, doğrusallığın modelde aşırı öğrenmeye neden olabileceği göz önünde bulundurulduğundan, her bir  $z_j$  çıktısı bir fonksiyondan geçirilerek lineer olmayan bir hale getirilir. Bu fonksiyona **aktivasyon fonksiyonu** denir [38].

En sık kullanılan aktivasyon fonksiyonlarından olan relu, girdinin pozitif olduğu her değeri kendisine, negatif olduğu her değeri ise sıfıra eşitlemektedir. Bu durumda denklemin son hali aşağıdaki gibi olacaktır:

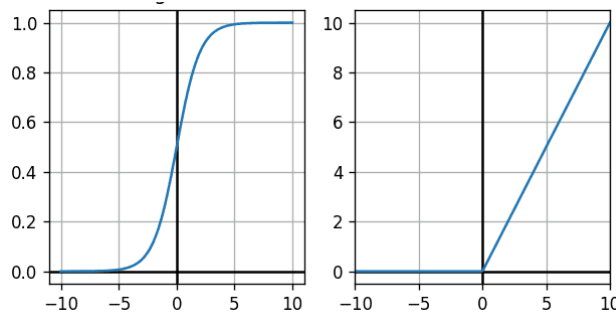
$$h = (f(z_1), f(z_2), f(z_3)) \quad (6)$$

İfade (6)'da,  $h$ , çıktı katmanını,  $f$  fonksiyonu aktivasyon fonksiyonunu,  $z_i$  ise girdi değerlerini temsil etmektedir.

Sigmoid aktivasyon fonksiyonu ise S şeklinde sigmoid bir eğri oluşturan bir aktivasyon fonksiyonudur ve formülü aşağıdaki gibidir:

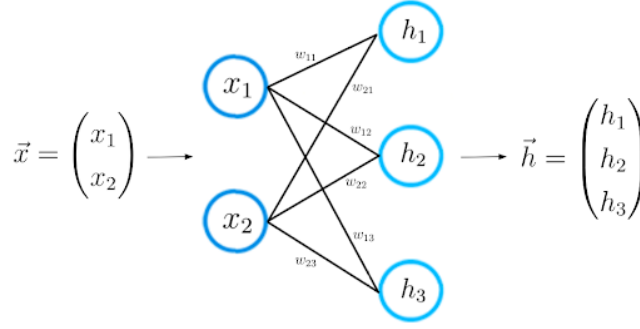
$$h = \frac{1}{1 + e^{-x}} \quad (7)$$

İfade (7)'de,  $h$ , çıktı katmanını, euler sayısını ve  $x$  girdi değerlerini temsil etmektedir.



Şekil 2. Sırasıyla, Sigmoid ve Relu Aktivasyon Fonksiyonları

Bir modelde en önemli etkiyi sağlayan ağırlıklardır. Eğitim süresince bazı nöronlar gerçek değere yaklaşmasını bazılarının ise uzaklaşmasını sağlamaktadır. Bunun sonucunda da ağırlıkları tayin edilmektedir.

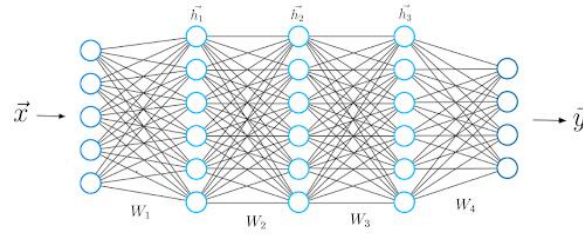


Şekil 3. Basit Bir Sinir Ağı Mimarisi

Daha derin bir öğrenme için oluşturulabilecek aşağıdaki gibi bir model için çıktı katmanı genel olarak aşağıdaki gibi hesaplanır:

$$y = f(f(f(f(X^T W_1 + b_1)W_2 + b_2)W_3 + b_3)W_4 + b_4) \quad (8)$$

İfade (8)'de,  $y$  bağımlı değişkeni,  $f$  fonksiyonu aktivasyon fonksiyonunu,  $X$  girdi matrisini,  $T$  matris transpozisini,  $W_1, W_2, W_3$  ve  $W_4$  ağırlık matrislerini  $b_1, b_2, b_3$  ve  $b_4$  yanlılık bileşenlerini temsil etmektedir.



Şekil 4. Bir Derin Öğrenme Modeli Mimarisi

Model ileri doğru beslendikten (**ileri besleme** ya da forward propagation [34]) sonra her adım sonrası gerçekte olması gereken değerler ile eğitim sürecinin sonuçların uzaklığının hesaplandığı **kayıp fonksiyonları** (loss functions [35]) kullanılır. Ortalama kare hatası, ortalama mutlak hata, ikili ve kategorik çapraz entropi en sık kullanılan kayıp fonksiyonlarıdır. Kategorik çapraz entropi, çok sınıflı sınıflandırma görevlerini içeren makine öğrenmesi modellerinde genellikle bir kayıp işlevi olarak kullanılan matematiksel bir işlevdir. Tahmin edilen olasılık dağılımı ile hedef değişkenin gerçek olasılık dağılımı arasındaki farkı ölçer. Kategorik çapraz entropinin hesaplanması şu şekildedir:

$$y = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C I_{y_i \in C_c} \log(p_{model}[y_i \in C_c]) \quad (9)$$

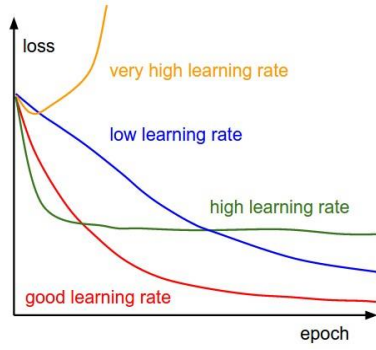
İfade (9)'da  $N$  gözlem sayısını,  $C$  kategorilerini ifade etmektedir:  $I_{y_i \in C_c}$ ,  $c$  kategorisine ait  $i$ . gözlemin gösterge işlevi ve  $p_{model}[y_i \in C_c]$ ,  $i$ . gözlemin  $c$ . kategoriye ait olması için model tarafından tahmin edilen olasılığıdır.

Kayıp fonksiyonun sonucuna göre modelde düğümler arasında atanan her bir ağırlık değeri ve bias değeri güncellenir. Bu güncelleme işlemine **geri besleme** (backpropagation [39]) denir. Bu adım ile modelde doğruluğun artmasına önemli ölçüde katkı sağlanır. Ancak bazı durumlarda hatanın düzeltilmesi için ağırlıklarda büyük değişimler yapılır ve bu durum gerçek yerel minimum'dan yani hedeften uzaklaşabilir. Bu nedenle kaybı/genelleme hatasını azaltmak için **gradyan azaltma** (gradient descent [40]) fonksiyonu kullanılır. En basit haliyle gradyan azaltma formülü aşağıdaki gibidir:

$$\theta_{t+1} = \theta_t - \alpha * \nabla J(\theta_t) \quad (10)$$

İfade (10)'da  $\theta_t$ ,  $t$  zamanındaki ağırlık vektörünü,  $\alpha$  öğrenme oranını,  $J(\theta_t)$  mevcut ağırlıklarda değerlendirilen kayıp fonksiyonunu ve  $\nabla J(\theta_t)$   $t$  zamanındaki ağırlıklara göre kayıp fonksiyonunun gradyanını ifade etmektedir.

Model için bir diğer önemli husus ise **öğrenme oranı** (learning rate [41]). Öğrenme oranı, her yinelemede maliyet fonksiyonundaki minimizasyon miktarı olarak tanımlanır ve en iyi sonuca en etkili şekilde ulaşmanın anahtarlarından birisi bu oranı optimize etmektir.



Şekil 5. Öğrenme Oranının Uygun Seçilmesinin Önemi [42]

Yüksek öğrenme oranlarını kullanarak öğrenme hızını arttırmak için ise normalizasyon yöntemlerinden **yığın normalizasyonu** (batch normalization [43]) kullanılabilir, formülü şu şekildedir:

$$\hat{y} = \frac{(x - \mu)}{\sqrt{(\sigma^2 + \epsilon)}} * \gamma + \beta \quad (11)$$

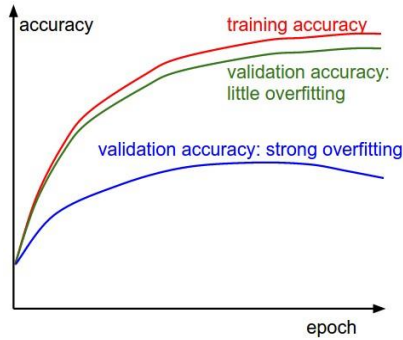
İfade (11)'de  $x$  katmanın girişi,  $\mu$  ve  $\sigma$ , giriş yığınının ortalama ve standart sapması,  $\epsilon$ , sıfıra bölünmeyi önlemek için kullanılan küçük bir değer,  $\gamma$  ve  $\beta$ , modelin her yığın için en uygun ölçeği ve ortalamayı öğrenmesini sağlayan öğrenilebilir ölçekleme (scaling) ve kaydırma (shifting) parametreleridir.

Modeldeki her ağırlığın gradyanı hesaplamak için model eğitilirken geri besleme kullanılır; hesaplanan gradyanlara göre modelin ağırlıklarını güncellemek için ise bir **optimizasyon algoritması** (optimization function [44]) kullanılır.

Derin Öğrenme modellerinde en yaygın kullanılan optimizasyon fonksiyonlarından birisi ADAM'dır. SGD gibi diğer optimizasyon algoritmalarına kıyasla derin sinir ağlarının eğitiminde onu daha etkili kılan bazı benzersiz özelliklere sahiptir. Bu özellikler arasında uyarlanabilir öğrenme hızı, momentum, verimli bellek kullanımı ve gürültülü gradyanlara karşı sağlamlık yer alır. Adam optimizasyonu daha hızlı yakınsama sağlayabilir ve çeşitli uygulamalar için derin öğrenme modellerinde yaygın olarak kullanılır. Adam'ın formülü şu şekildedir:

$$\theta_{t+1} = \theta_t - \frac{\alpha x mt}{\sqrt{vt + \epsilon}} \quad (12)$$

İfade (12)'de  $\theta_t$ ,  $t$  zamanındaki ağırlık vektörü,  $\alpha$  öğrenme oranı,  $mt$  ve  $vt$ , gradyanların birinci ve ikinci moment tahminleri,  $\epsilon$  ise, sıfıra bölünmeyi önlemek için kullanılan küçük bir değerdir.



Şekil 6. Eğitim ve Doğrulamanın Öğrenme Üzerine Etkisi [45]

Modelin sonuçlarını değerlendirmek için kullanılan performans ölçümlerine model metriği denir. Doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skor sınıflandırma modellerinde en yaygın kullanılan model metrikleridir. Doğruluk, doğru tahmin sayısının toplam tahmin sayısına oranı iken, F1 skorun formülü aşağıdaki gibidir:

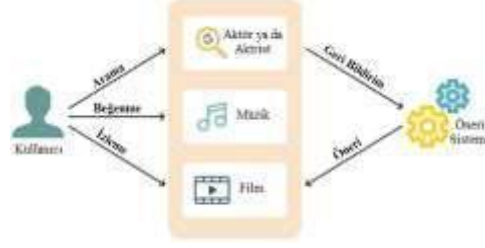
$$F = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (13)$$

İfade (13)'te  $TP$  gerçek pozitifler değer sayısı iken,  $FP$  ve  $FN$ , yanlış pozitif ve negatif sayılarıdır.

Derin öğrenmenin en avantajlı olduğu noktalardan birisi büyük veri setlerinde çalışabilmenin kolaylığıdır. Bunu sağlayan en önemli etkenlerden birisi, derin öğrenme algoritmalarının öğrenmeyi toplu olarak değil de verileri küçük **yığınlar** (batch [46]) bölerek eğitmeye müsait olmasıdır. **Devir** (epoch [47]), her bir öğrenme döngüsüne verilen ad iken, **bırakma** (dropout [48]), gizli katmanlardaki rastgele nöronların ilgili epochta eğitime dahil edilmeyip bu şekilde modelin aşırı öğrenme konusunda düzenleme yapıldığı tekniktir.

## 2.2. Öneri Sistemleri ve Seyahat Endüstrisi

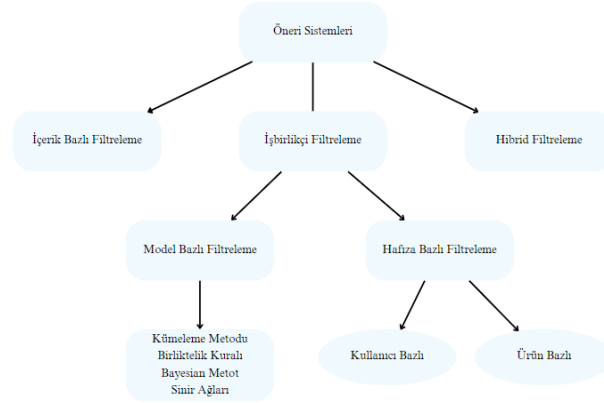
Her geçen gün veri yoğunluğunun arttığı ve günlük yaklaşık 118 zettabayt (yaklaşık 130 trilyon gigabyte) verinin üretildiği günümüzde [49], bu verilerin anlamlandırılması ve kullanıcılara tercihlerine ve davranışlarına göre kişiselleştirilmiş öneriler sunarak yeni ürünleri, hizmetleri ve içerikleri keşfetmelerini kolaylaştırmak günümüzün en önemli konularından birisi haline gelmiştir.



Şekil 7. Basit Bir Öneri Sistemi Mimarisi

Seyahat öneri sistemleri, seyahat planlamayı daha erişilebilir ve daha az zaman alıcı hale getirerek günlük yaşamlarımız üzerinde önemli bir etkiye sahip olmuştur. Bu sistemlerin kullanımıyla gezginler, tercihlerine ve bütçelerine göre varış yerleri, oteller ve aktiviteler için kişiselleştirilmiş öneriler alabilirler. Bu sadece zamandan tasarruf sağlamakla kalmaz, aynı zamanda gezginlerin daha keyifli bir seyahat deneyimi yaşamasını da sağlar. Ayrıca müşteri deneyimi artırması nedeniyle, seyahat öneri sistemlerinin de seyahat endüstrisi üzerinde önemli bir etkisi olmuştur.

Öneri sistemlerinin birçoğunda, sistemin bulunduğu ortama kullanıcı veya ortam ile ilgili eklenebilecek verilerin sağlanmasının ardından filtreleme algoritmaları kullanılmaktadır. Günümüzde oldukça fazla ve farklı amaçlara uygun filtreleme algoritması bulunmakla beraber en popüler 3 tanesi: kullanıcıların önceki işlemlerinde kullandığı içeriklere uygun içeriklerin önerildiği İçeriğe Dayalı (Content-based) Öneri Sistemleri, kullanıcı davranışlarına göre benzer kullanıcı gruplarına göre içerik önerisi sunan İşbirliğine Dayalı (Collaborative) Öneri Sistemleri ve bunların birlikte kullanılmasıyla oluşan Hibrit Öneri Sistemleridir.



Şekil 8. Öneri Sistemlerinde Filtreleme Yöntemleri

Google tarafından geliştirilen geniş ve derin sinir ağı (wide and deep neural networks) ve Facebook tarafından geliştirilen derin öğrenme öneri modeli (deep learning recommendation model), öneri görevleri için kullanılan en popüler makine öğrenme algoritmalarındandır. Her ikisi de, kullanıcı ve öğe özellikleri arasındaki etkileşimleri yakalamak ve kullanıcılara öğeleri önermede yüksek doğruluk elde etmek için doğrusal ve doğrusal olmayan modellerin bir kombinasyonunu kullanır.

**Geniş ve Derin Sinir Ağı** [1], Google tarafından öneri sistemleri için geliştirilen bir makine öğrenme algoritmasıdır. Temelinde, kullanıcı ve öğe özellikleri arasındaki hem doğrusal hem de doğrusal olmayan ilişkileri yakalamak için geniş bir doğrusal modeli ve derin bir sinir ağını birleştirerek tasarlanır.

Geniş doğrusal model, kullanıcı ve öğe özelliklerinin ana etkilerini yakalayan basit bir doğrusal regresyon modelidir. Derin sinir ağı ise bu özellikler arasındaki karmaşık etkileşimleri yakalar.

Geniş ve derin sinir ağı iki ana bileşenden oluşur:

- **Geniş Model:** Bu model, kullanıcı ve öğe özellikleri arasındaki etkileşimleri yakalamak için çapraz çarpım dönüşümleri kullanan doğrusal bir modeldir.
- **Derin Model:** Bu model, kullanıcı ve öğe özellikleri arasındaki doğrusal olmayan etkileşimleri yakalamak için birden çok gizli katman kullanan ileri beslemeli bir sinir ağıdır.

Geniş ve derin sinir ağı, hem doğrusal hem de doğrusal olmayan modellerin güçlerini birleştirerek yüksek doğruluk elde eder. Geniş model, kullanıcı ve öge özelliklerinin ana etkilerini yakalarken derin model, bunlar arasındaki karmaşık etkileşimleri yakalar.

**Derin Öğrenme Öneri Modeli (DLRM)** [2], Facebook tarafından öneri görevleri için oluşturulan bir makine öğrenmesi algoritmasıdır. Öneri puanları üretmek için kullanıcı ve öge yerleştirmelerini hesaba katan bir tür sinir ağıdır.

DLRM algoritması üç ana bileşenden oluşur:

- **Gömme Katmanı (Embedding Layer):** Bu katman, seyrek girdi özelliklerini, kullanıcıların ve öğelerin gizli özelliklerini yakalayan yoğun katmanlara dönüştürür. Kategorik özellikleri benzersiz sayısal değerlere dönüştürmek için bir karma tablo (hash table) kullanır.
- **Alt MLP:** Bu katman, aralarındaki etkileşimleri öğrenmek için girdi özelliklerini işleyen çok katmanlı bir algılayıcıdır (multilayer perceptron).
- **Üst MLP:** Bu, alt MLP'nin çıktısını işleyen ve öneri puanlarını oluşturan başka bir çok katmanlı algılayıcıdır.

DLRM yöntemi, öneri sistemleri, yüksek derecede seyrekliğe sahip büyük veri kümelerini işlemek içindir. Hem alt hem de üst MLP'leri kullanarak kullanıcı ve öge gömmeleri arasındaki etkileşimleri öğrenerek yüksek doğruluk elde eder.

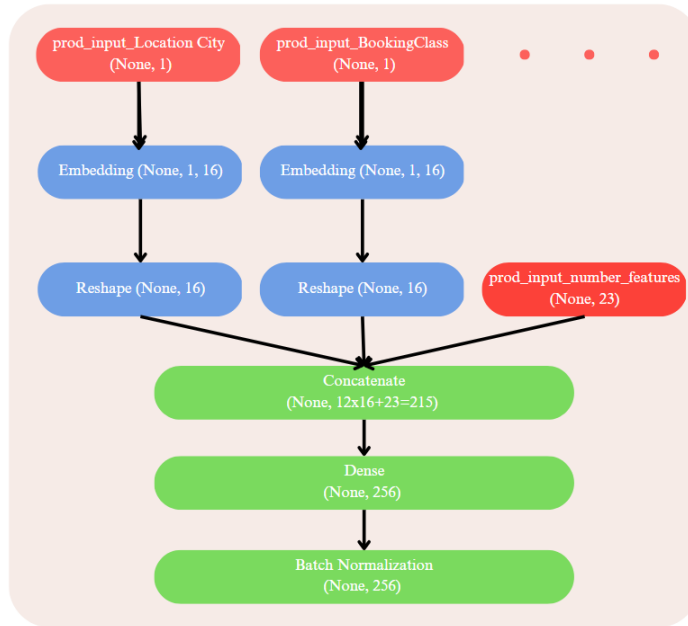
### 3. Önerilen Yöntem: Derin ve Geniş Öğrenme Tabanlı Seyahat Öneri Sistemi

Bu çalışma kapsamında, gerçek bir seyahat acentesi şirketinin veri seti kullanılarak, geniş ve derin öğrenme yaklaşımı ile kullanıcı ve ürün özelliklerinin kategorik olanlarının gömme işlemi uygulanarak, nümerik veriler ile modele beslendiği yeni bir seyahat öneri sistemi oluşturulur.

Oluşturulan model, kullanıcı modeli, ürün modeli ve iki modeli birleştiren birleştirme katmanı olmak üzere üç ana bölümden oluşur. Kullanıcı ve ürün modellerinin her biri, yoğun (dense) vektörleri sabit boyuta dönüştüren yerleştirmelerden (embedding) ve tamamen bağlantılı (fully connected layer) bir katmandan oluşur.

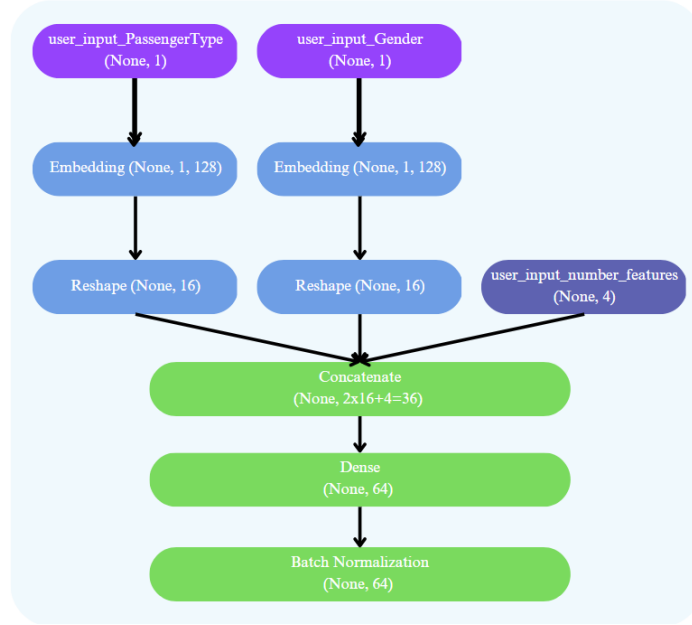
Her kategorik özellik gömme işlemine tabi tutulur. Gömme katmanından sonra, birleştirmelerin yapılabilmesi için boyut dönüştürme katmanı (Reshape) uygulanır. Kategorik katmanlar ayrı ayrı gömme işlemine tabi tutulup dönüşüm yapıldıktan sonra, tüm nümerik özelliklerin bulunduğu bir katman ile birleştirilir. Ardından yoğun katmandan (dense) geçirilirler.

Kullanıcı ve öge modelleri yığın normalleştirme (batch normalization) yapılarak tamamen bağlantılı katmana (fully connected layer) beslenir. Başka bir deyişle, kullanıcı ve ürün yerleştirmeleri, birbirinden bağımsız olarak her bir kullanıcı ve ürün için tek bir vektör oluşturmak üzere birleştirilir. İki vektör daha sonra model için son giriş vektörünü oluşturmak üzere birleştirilir. Modelin son katmanı, ürünler üzerinde bir olasılık dağılımı veren softmax aktivasyon fonksiyonuna sahip tamamen bağlantılı bir katmandır.

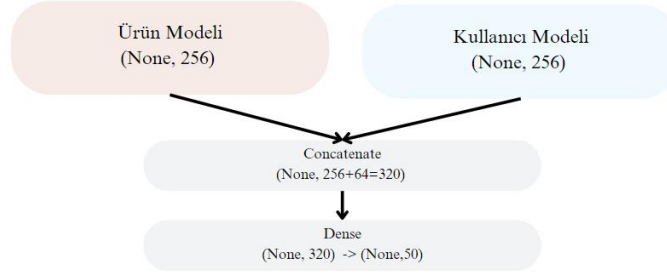


Şekil 9. Derin ve Geniş Seyahat Öneri Sistemi Ürün Modeli Mimarisi





Şekil 10. Derin ve Geniş Seyahat Öneri Sistemi Kullanıcı Modeli Mimarisi



Şekil 11. Derin ve Geniş Seyahat Öneri Sistemi Model Mimarisi

Her bir yığın normalleştirmeden önce 0.1 oranında bırakma (dropout) ve son katman hariç her yoğun katmanda relu aktivasyon fonksiyonu uygulanır.

Modeldeki kategorik özellikleri temsil etmek için kullanılan gömme vektörlerinin boyutunu ifade eden embedding\_size parametresi, her kategori özelliği 16 boyuttan oluşan bir vektörle temsil edilecek şekilde ayarlanır.

Eğitim sürecinin her yinelemesinde kullanılan eğitim örneklerinin sayısını ifade eden batch\_size, 512 olarak ayarlanır; bu, modelin bir seferde 512 eğitim örneğini analiz ettikten sonra ağırlıklarını güncelleyeceği anlamına gelir.

Eğitim sırasında eğitim veri kümesinin model üzerinden çalıştırılma sayısını ifade eden epochs, modelin 10 dönem eğitilmesi için ayarlanır.

Modelin yoğun katmanındaki nöron sayısı olan dense\_layer\_size, ürün özellik gömmeleri için 256 ve kullanıcı özellik gömmeleri için 64 olarak ayarlanır.

Model, kategorik çapraz entropi (categorical cross-entropy) kayıp fonksiyonu, doğruluk (accuracy) ve F1 skoru ölçümleri ve ADAM iyileştirici ile derlenir.

Önerilen model, hem kullanıcıların hem de varış noktalarının özelliklerini birleştirerek, farklı değişkenler arasındaki karmaşık ilişkileri yakalayabilen güçlü, doğrusal olmayan bir model oluşturur.

Facebook'un DLRM'i ve Google'ın geniş ve derin sinir ağı gibi mevcut yaklaşımlarla karşılaştırıldığında, önerilen yöntem, kullanıcı ve ürün verilerini model içinde ayrı ayrı işleme konusundaki benzersiz yaklaşımıyla öne çıkıyor. Bu yöntem ile kullanıcı tercihleri ve davranışları daha incelikli bir şekilde anlaşılması sağlanarak daha doğru ve kişiselleştirilmiş önerilere yol açılmaktadır.

Genel olarak önerilen yöntem, öneri sistemleri alanında ileriye doğru atılmış önemli bir adımı temsil ediyor ve gerçek dünya verilerine ve kullanıcı davranışına dayanan seyahat varış noktası önerisine güçlü ve yenilikçi bir yaklaşım sunuyor. Daha fazla araştırma ve geliştirme ile bu yaklaşım, seyahat endüstrisini dönüştürme ve dünya çapında milyonlarca insan için seyahat deneyimini iyileştirme potansiyeline sahiptir.

## 4. Deneysel Çalışmalar ve Sonuçlar

Bu bölümde, yapılan çalışmanın içeriğine ve sonuçlarına dair açıklamalara yer verilmiştir. Birinci alt bölümde, veri kümesi incelenmiş, ikinci ve üçüncü alt bölümlerde veri ön işleme adımları ile model öncesi adımlar gerçekleştirilmiş, dördüncü alt bölümde model geliştirilmiş ve son alt bölümde sonuçlar incelenmiştir.

### 4.1. Veri Kümesi

Derin öğrenme modelleri, sistemlerin bireysel müşteriler için doğru tahminler yapmasını sağlayarak, büyük veri kümelerinden karmaşık örüntüleri öğrenme yeteneğine sahiptir. Bu nedenle, uçak bileti satışı konsolidatör acentesi olan BiletBank'ın 2012-2017 yılları arasındaki verisi kullanılarak önerilen uçuş öneri sistemi test edilmiştir. Aynı veri seti kısmi olarak [50-53] çalışmalarında da kullanılmıştır. [50]'de Spark büyük veri platformu üzerinde koşturulan işbirlikçi algoritmalar kullanılarak bir uçuş öneri sistemi geliştirilmiştir. [51]'te uçuş için satılan ek hizmetler değerlendirilerek ek hizmetler öneri sistemi geliştirilmiştir. [52, 53]'de acentelerin sattıkları uçak biletleri ve davranışları incelenerek müşteri kayıp analizi üzerine çalışmalar yapılmıştır.

Veri kümesi olarak, üye acente şirketinin 2175911 tekil müşterinin, 4218964 satırlık yaklaşık 1 gigabyte boyutundaki 2017 ve öncesindeki müşteri ve uçuş bilgileri ile ilgili acentelerin bilgilerini içeren seyahat verisi kullanılmıştır. Toplanan veriler ön işleme aşamalarından geçirilmiş; veri temizliği, normalizasyon, merkezileştirme, dönüştürme, bütünleştirme ve aykırı durum tespiti gibi ön işlemler uygulanmıştır.

Tablo 1. Veri Kümesindeki Kolonların Açıklamaları

Özellik	Açıklama
<i>AgencyId</i>	Satışı yapan Acente ID
<i>AgencyType</i>	Acente türü
<i>ArrivalDay</i>	Variş günü
<i>ArrivalTime</i>	Variş saati
<i>BirthDate</i>	Doğum Tarihi
<i>BookingClass</i>	Uçuş sınıfı
<i>BookingCode</i>	PNR
<i>DefaultCurrency</i>	Varsayılan para birimi
<i>DepartureDay</i>	Kalkış günü
<i>DepartureTime</i>	Kalkış saati
<i>Destination</i>	Variş havalimanı
<i>FareBasis</i>	Sınıf ve ücret tipinin kodu
<i>FareType</i>	Ücret tipi kodu
<i>Flight_Baggage</i>	Bagaj boyutu
<i>Gender</i>	Cinsiyet
<i>IsExpirationChange</i>	Opsiyon değişikliği etiketi
<i>IsRefund</i>	Geri iade edilmiş mi etiketi
<i>IsScheduleChange</i>	Uçuş saati değişti mi etiketi
<i>Location_City</i>	Acentenin bulunduğu şehir
<i>Location_Country</i>	Acentenin bulunduğu ülke
<i>LoyaltyProgram</i>	Sadakat programı dahil mi?
<i>MarketingAirline</i>	Satan havayolu
<i>OD_Destination</i>	Variş havalimanı
<i>OD_Origin</i>	Başlangıç havalimanı
<i>OperatingAirline</i>	Uçan havayolu
<i>Origin</i>	Başlangıç havalimanı
<i>PassengerId</i>	Yolcu ID
<i>PassengerType</i>	Yolcu Tipi
<i>PaymentType</i>	Ödeme türü
<i>SeatCount</i>	Biletlenen koltuk sayısı
<i>SellingDate</i>	Satış tarihi
<i>Status_IfActive</i>	Aktif mi

<i>T_CreationDate</i>	Acente kayıt oluşturulma tarihi
<i>TicketTotal</i>	Toplam satış tutarı
<i>TripType</i>	Yolculuk tipi
<i>Type</i>	PassengerType değeri
<i>VatFlag</i>	Yurt içi, Yurt dışı
<i>WheelChairServiceType</i>	Tekerlekli sandalye isteniyor mu?

Veri Kümesi, PassengerId, PassengerType, Gender, BirthDate ve LoyaltyProgram gibi yolcu bilgileri de dahil olmak üzere uçak bileti satışlarıyla ilgili çeşitli veri özelliklerini içermektedir. Rezervasyon bilgileri, SellingDate, BookingCode, BookingClass ve FareBasis gibi ayrıntıları içerir. Ayrıca MarketingAirline, OperatingAirline, OD\_Origin, OD\_Destination, DepartureDay, ve ArrivalDay gibi uçuş bilgilerini gösteren özellikler bulunmaktadır. Diğer önemli özellikler, ödeme ayrıntıları, geri ödemeler ve para birimi bilgileri hakkında bilgi sağlayan PaymentType, IsRefund, VatFlag ve DefaultCurrency'dir. Tablo 1'de tüm öznitelikler ve açıklamaları sunulmaktadır.

Ham veri kümesi kolonlarının veri türleri, boş değer sayıları (#null), benzersiz veri sayıları ve nümerik kolonlar için yüzdelik değerleri Tablo 2'de verilmiştir.

Tablo 2. Veri Kümesindeki Kolonların Veri Tipleri, Boş Değer ve Benzersiz Değer Sayıları

Özellik	Veri Tipi	#NULL	#Unique
<i>AgencyId</i>	int32	0	3342
<i>AgencyType</i>	category	13406	5
<i>ArrivalDay</i>	object	0	2406
<i>ArrivalTime</i>	object	0	1082
<i>BirthDate</i>	int16	0	251
<i>BookingClass</i>	category	94	108
<i>BookingCode</i>	object	11	2359497
<i>DefaultCurrency</i>	category	2	4
<i>DepartureDay</i>	object	0	2405
<i>DepartureTime</i>	object	0	780
<i>Destination</i>	object	0	1060
<i>FareBasis</i>	object	151760	21172
<i>FareType</i>	category	3855213	30
<i>Flight_Baggage</i>	category	41193	161
<i>Gender</i>	category	0	2
<i>IsExpirationChange</i>	int8	0	2
<i>IsRefund</i>	int8	0	2
<i>IsScheduleChange</i>	int8	0	2
<i>Location_Country</i>	object	2	294
<i>LoyaltyProgram</i>	category	2	31
<i>MarketingAirline</i>	int8	0	2
<i>OD_Destination</i>	object	0	321
<i>OD_Origin</i>	object	0	1059
<i>OperatingAirline</i>	object	0	1012
<i>Origin</i>	object	0	393
<i>PassengerId</i>	object	0	1014
<i>PassengerType</i>	int32	0	2175911
<i>PaymentType</i>	category	0	5
<i>SeatCount</i>	category	5515	8
<i>SellingDate</i>	int8	0	12
<i>Status_IfActive</i>	object	0	2311087
<i>T_CreationDate</i>	float64	2	2

<i>TicketTotal</i>	object	2	2699
<i>TripType</i>	float16	0	7861
<i>Type</i>	category	0	3
<i>VatFlag</i>	int8	0	5
<i>WheelChairServiceType</i>	category	0	2
<i>Location_Country</i>	int8	0	4

AgencyType, BookingClass, BookingCode, DefaultCurrency, FareBasis, FareType, Flight\_Baggage, Location\_Country, PaymentType, Status\_IfActive, T\_CreationDate kolonlarının kayıp veriye (boş değere) sahip olduğu Tablo 5.3'te de görülmektedir. Veri dönüşüm aşamalarında bu değerler için işlemler yapılmıştır.

Aynı şekilde, bazı kolonların nümerik olup kategorik davrandığı ve tersi Tablo 2'de görülmektedir ve bu kolonlara tip dönüşüm işlemleri, veri dönüşüm kısmında yapılmıştır.

Tablo 3. Veri Kümesindeki Nümerik Kolonların Yüzdellik Değerleri

Özellik	0	0.05	0.5	0.95	1
<i>AgencyId</i>	10088	1545960	1547653	1553033	1554048
<i>BirthDate</i>	1	1956	1986	2007	2017
<i>IsExpirationChange</i>	0	0	0	0	1
<i>IsRefund</i>	0	0	0	0	1
<i>IsScheduleChange</i>	0	0	0	0	1
<i>LoyaltyProgram</i>	0	0	0	0	1
<i>PassengerId</i>	1	271454	2591078	4956012	5218779
<i>SeatCount</i>	0	1	1	4	11
<i>Status_IfActive</i>	0	0	1	1	1
<i>TicketTotal</i>	0	65	168	864	29408
<i>Type</i>	1	1	1	4	5
<i>WheelChairServiceType</i>	0	0	0	0	3

Tablo 3'te görüldüğü üzere bazı kolonlar düşük kardinaliteye sahip olmasına rağmen nümerik veri tipinde bulunmaktadır ve bu veri tiplerinin dönüşümleri veri dönüşüm aşamasında yapılmıştır.

## 4.2. Veri Ön İşleme

Veriyi modelde kullanabilmek için gerekli temizleme, filtreleme ve dönüştürme gibi veri madenciliği işlemleri uygulanmıştır ve yeni özellikler oluşturulmuştur.

Çalışmaya, kütüphaneler eklenip veriler okunduktan sonra, çalışmada kullanılacak veri kümeleri (uçuş ve acente verileri) birleştirilmiştir ve tek bir veri kümesi elde ederek başlanmıştır. Ardından, 1920'den sonra doğanlar ya da kalkış tarihi mantıklı olanlar gibi mantıksal veri filtrelemeleri yapılmıştır.

Aykırı veriler tespit edilerek bu aşamada filtrelenmiştir. Eksik verileri doldurma işleminde, veri setinin büyük bir kısmı boş olan önemsiz özelliklerin kaldırılması, diğerlerinin de doldurulması işlemi uygulanmıştır. Kategorik ve nümerik veriler ile tarih bazlı verilerin dönüşümleri yapılmıştır. Kardinalitesi yüksek özellikler belli bir eşik değerinden sonra (Others gibi) gruplanmıştır.

Mantıksal olarak modele daha uygun olacağı için, her bir müşterinin her bir aktarmalı uçuşunun iniş ve kalkış zamanları o rezervasyonun ilk kalkış ve son iniş zamanına eşitlenmiştir. Bilet ücreti kolonu TicketTotal 0-250, 250-500, 500-750, 750-1000, 1000-1250, 1250-1500, 1500-2000 ve 2000+ için kategorik değerlere gruplanmıştır.

Çift kopyalı (duplike) özellikler yok edilmiş ve doğum tarihinden müşteri yaşı; bilet satış, uçak kalkış ve iniş tarihlerinden saat, gün, hafta, ay ve yıl verileri elde edilmiştir.

Son olarak bu işlemlerin sonucunda oluşan gereksiz kolonlar elenmiş, toplamda 50 özellikli ve Tablo 4'te verilen dağılıma sahip bir veri kümesi oluşturulmuştur

Tablo 4. Veri Kümesinin Özellik-Veri Tipi Dağılımı

	Adet	Özellikler
ID	3	PassengerId,AgencyId,BookingCode
Nümerik	8	CustomerAge, SellingDateDayofyear, SellingDateWeekofyear, ArrivalDateTimeDayofyear ...
Düşük Kardinalite, Kategorik	12	TicketTotal, Type, PassengerType, Gender, VatFlag, TripType, PaymentType, AgencyType, DefaultCurrency ...
Yüksek Kardinalite, Kategorik	11	Origin, Destination, BookingClass, MarketingAirline, OperatingAirline, FareBasis, Location_City ...
Düşük Kardinalite, Kategorik Nümerik	10	WheelChairServiceType, IsRefund, LoyaltyProgram, IsScheduleChange, IsExpirationChange, Status_IfActive ...
Yüksek Kardinalite, Kategorik Nümerik	6	SeatCount, Flight_Baggage_Dep, Flight_Baggage_Arr, SellingDateMonth, ArrivalDateTimeMonth ...

Tablo 5. Veri Kümesindeki Kolonların Ön İşleme Sonrası Veri Tipleri ve Benzersiz Değer Sayıları

Özellik	Veri Tipi	#Benzersiz Değer
AgencyId	int32	3339
AgencyType	category	5
ArrivalDayofyear	int32	366
ArrivalHourDayParts	object	4
ArrivalMonth	int32	12
ArrivalWeekofyear	int32	53
ArrivalYear	int32	8
BookingClass	category	107
BookingCode	object	2355592
CustomerAge	int64	97
DefaultCurrency	category	4
DepartureDayofyear	int32	366
DepartureHourDayParts	object	4
DepartureMonth	int32	12
DepartureWeekofyear	int32	53
DepartureYear	int32	8
DeparturetoArrivalDuration	float64	3
Destination	object	884
FareBasis	object	20689
Flight_Baggage_Arr	int32	19
Flight_Baggage_Dep	int32	21
Gender	category	2
IsExpirationChange	int8	2
IsRefund	int8	2
IsScheduleChange	int8	2
Location_City	object	294
Location_Country	category	31

Özellik	Veri Tipi	#Benzersiz Değer
LoyaltyProgram	int8	2

MarketingAirline	object	314
OD_Destination	object	1055
OD_Origin	object	1006
OperatingAirline	object	388
Origin	object	982
PassengerId	int32	2174194
PassengerType	category	5
PaymentType	category	3
SeatCount	int8	12
SellingDayofyear	int32	366
SellingHourDayParts	object	4
SellingMonth	int32	12
SellingWeekofyear	int32	53
SellingYear	int32	6
SellingToDepartureDuration	float64	172243
Status_IfActive	float64	2
T_CreationDate	object	2696
TicketTotal	category	7843
TripType	category	2
Type	category	5
VatFlag	category	2
WheelChairServiceType	int32	2

### 4.3. Model Öncesi Hazırlık

Modelden önce veri üzerinde yapılan dönüşümler-değişiklikler aşağıdaki gibidir:

- Veriler, yalnızca en popüler 50 uçuş noktası tutularak filtrelenir ve geri kalanlar (tüm verinin %4,52'si) verilerden kaldırıldı.
- Kategorik özellikler kategorik veri tipine, sayısal özellikler ise float veri tipine dönüştürüldü.
- Veriler, eğitim, test ve doğrulama kümelerine bağımlı ve bağımsız değişkenler olarak ayrıldı.
- Eğitim verilerinin PassengerId, OD\_Origin ve OD\_Destination bazında gruplanarak aynı yolcu tarafından belirli bir seyahatin kaç kez rezerve edildiğini gösteren yeni bir özellik olan Affinity oluşturuldu.
- Test ve doğrulama veri kümelerinde, Affinity kolonuna eğitim veri kümesindeki Affinity kolonu değeri atandı ve eksik olanlar için Affinity'ye 0 atandı (Bu işlem modelde yanlışlık oluşmasını engellemek için yapılmıştır).
- Origin ve Destination kolonlarını kapsayan OD\_Origin ve OD\_Destination özellikleri kaldırıldı.
- Model tüm özellikleri nümerik cinsten almak istediği için, hedef değişken (y\_train, y\_test, y\_val) one-hot kodlamaya tabi tutuldu ve kategorik özellikler, sklearn'den OrdinalEncoder kullanılarak kodlandı. Sayısal özellikler ise aykırı değerleri önemsizleştiren, sklearn'den RobustScaler kullanılarak ölçeklendi.
- Kodlanmış kategorik ve ölçeklendirilmiş sayısal özellikler, eğitim, test ve doğrulama için nihai veri kümelerini oluşturmak üzere birleştirildi ve gereksiz değişkenler silindi.
- Öğe ve kullanıcıların nümerik ve kategorik özellikleri Tablo 6'daki gibi ayrıldı.

Tablo 6. Kullanıcı-Öğe, Nümerik-Kategorik Özellik Dağılımı

	Özellikler
Kategorik Kullanıcı Özellikleri	PassengerType, Gender
Nümerik Kullanıcı Özellikleri	CustomerAge, LoyaltyProgram, Affinity
Kategorik Öğe Özellikleri	TicketTotal, Origin, BookingClass, MarketingAirline, OperatingAirline, Location_City, Location_Country, VatFlag, TripType, PaymentType, AgencyType, DefaultCurrency
Nümerik Öğe Özellikleri	SeatCount, Flight_Baggage_Dep, Flight_Baggage_Arr

### 4.4. Modeli Uygulama

Derin öneri sistemi model kodu, parametrelerin tanımlanması, modelin oluşturulması ve modelin eğitimi olmak üzere üç bölümden oluşmaktadır.

Modelde kullanılan hiper parametreler şu şekildedir:

- **embeddings\_size:** Modeldeki kategorik özellikleri temsil etmek için kullanılan gömme vektörlerinin boyutunu ifade etmektedir. Bu çalışmada her kategori özelliği 16 katman boyutundan (layer size) oluşan bir vektörle temsil edilmiştir.
- **batch\_size:** Eğitim sürecinin her yinelemesinde kullanılan eğitim örneklerinin sayısını ifade eder. Bu çalışmada 512 olarak ayarlanmıştır; bu, modelin bir seferde 512 eğitim örneğini analiz ettikten sonra ağırlıklarının güncelleyeceği anlamına gelmektedir.
- **epochs:** Eğitim sırasında eğitim veri kümesinin model üzerinden çalıştırılma sayısını ifade eder. Bu çalışmada model 10 dönem için eğitilmiştir.
- **dense\_layer\_size:** Modelin yoğun katmanındaki nöron sayısıdır. Yoğun katman, modelin çıktı katmanından önceki son katmandır ve amacı, önceki katmanlardan gelen bilgileri nihai bir tahmine dahil etmektir. Bu çalışmada ürün özellik gömmeleri için 256 ve kullanıcı özellik gömmeleri için 64 olarak ayarlanmıştır.

Model, kullanıcı modeli, ürün modeli ve iki modeli birleştiren birleştirme katmanı olmak üzere üç ana bölümden oluşmaktadır. Kullanıcı ve ürün modellerinin her biri, yoğun vektörleri sabit boyuta dönüştüren yerleştirmelerden ve tamamen bağlantılı bir katmandan oluşmaktadır.

Her kategorik özellik gömme işlemine tabi tutulur. Gömme katmanından sonra, birleştirmelerin yapılabilmesi için boyut dönüştürme katmanı uygulanır. Kategorik katmanlar ayrı ayrı gömme işlemine tabi tutulup dönüşüm yapıldıktan sonra, tüm nümerik özelliklerin bulunduğu bir katman ile birleştirilir. Ardından yoğun katmandan geçirilirler.

Kullanıcı ve öğe modelleri yığın normalleştirme yapılarak tamamen bağlantılı katmana beslenir. Başka bir deyişle, kullanıcı ve ürün yerleştirmeleri, birbirinden bağımsız olarak her bir kullanıcı ve ürün için tek bir vektör oluşturmak üzere birleştirilir. İki vektör daha sonra model için son giriş vektörünü oluşturmak üzere birleştirilir. Modelin son katmanı, ürünler üzerinde bir olasılık dağılımı veren softmax aktivasyon fonksiyonuna sahip tamamen bağlantılı bir katmandır.

Her bir yığın normalleştirmeden önce 0.1 oranında bırakma ve son katman hariç her yoğun katmanda relu aktivasyon fonksiyonu uygulanmıştır.

Model, kategorik çapraz entropi kayıp fonksiyonu, doğruluk ve F1 skoru ölçümleri ve ADAM iyileştirici ile derlenmiştir.

Doğruluk, sınıflandırma görevleri için tasarlanmış makine öğrenmesi modellerinde yaygın olarak kullanılan bir performans ölçüsüdür. Doğru sınıflandırılmış örneklerin toplam örnek sayısına oranını ölçer. Doğruluğun bir metrik olarak kullanılmasının nedeni, modelin yeni veri noktalarını doğru bir şekilde sınıflandırma yeteneği açısından ne kadar iyi performans gösterdiğini değerlendirmenin basit bir yolunu sağlamasıdır. Ancak, bir modelin performansını tam olarak değerlendirmek için diğer ölçümleri doğrulukla birlikte kullanmak önemlidir. Bu nedenle modelde doğruluğun yanında F1 metriği de kullanılmıştır. F1 metriği, hem kesinliği hem de geri çağırmaı hesaba kattığı için dengesiz veri kümeleri için yararlı bir metriktir. Bunlara ek olarak model çıktıları, k=5 ve k=10 için her bir kayıt için en iyi k çıktının doğruluğunu veren en iyi k-doğruluk (top k-accuracy) ile değerlendirilmiştir. Başka bir deyişle, her bir kullanıcıya k adet öneri sunulduğunda, kullanıcının tercih ettiği gerçek doğru (ground-truth) olan seçeneğin, algoritmanın önerdiği seçenekler arasında olup olmadığı kontrol edilir. Eğer kullanıcının gerçekte tercih ettiği (ground-truth) seçenek önerilen seçenekler arasında ise kullanıcı için yapılmış olan öneri doğru kabul edilir. En iyi k-doğruluk oranı, önerisi doğru olarak kabul edilen kullanıcı sayılarının toplamının tüm kullanıcı sayısına oranıdır.

## 4.5. Sonuçlar

Hiç şüphesiz ki performans, herhangi bir sinir ağı modelinin en kritik yönlerinden biridir. Bu çalışmada, derin ve geniş bir sinir ağı kullanan önerilen bir seyahat öneri sistemi ile Tablo 7'deki sonuçlar elde edilmiştir.

Tablo 7. Model Sonuçları

	Eğitim (onuncu devir)	Test
Kategorik çapraz entropi	1.4003	1.3991
En iyi 5 Doğruluk	0.8237	0.8236
En iyi 10 Doğruluk	0.9089	0.9090
Doğruluk	0.5412	0.5418
F1	0.8236	0.5732

Modelin büyük veri kümeleri üzerinde etkili bir şekilde eğitim verme yeteneği, genellikle bu zorlukla mücadele eden geleneksel makine öğrenmesi ve öneri sistemi modellerine göre önemli bir avantajdır. Bu yaklaşımın başarısı, karmaşık veri analizi ve işleme ihtiyaçlarının ele alınmasında derin öneri sistemlerinin etkinliğinin ikna edici bir doğrulaması olarak hizmet eder.

Çalışma esnasında karşılaşılan en büyük zorluklardan birisi örtülü/gizli geri besleme(implicit feedback) öneri sistemlerinin en zorlayıcı konularından birisidir. Kullanıcının puanlamasını sayfa görüntülemeleri, tıklamalar ve satın alma geçmişi gibi çeşitli davranışlardan tespit etmek meydan okuyucu bir zorluktur. Bu zorlukla başa çıkmak için mümkün varış duraklarının puanlaması değil de müşterinin hangi varış durağını seçeceğine yönelik bir model geliştirilerek örtülü besleme sorunu çözülmüştür.

Model eğitimi esnasında mümkün olduğunda hesaplama karmaşıklığını düşürecek optimal parametreler kullanılarak eğitim işlemi gerçekleştirilmiştir. Daha optimize ve/veya karmaşık modeller kullanılarak sonuçların iyileştirilmesi mümkündür.

## 5. Sonuç

Literatür araştırmasında da görüldüğü üzere kişiselleştirilmiş öneri sistemleri birçok alanda başarılı bir şekilde uygulama alanı bulmuştur.

Bu makalede, yapay sinir ağları ve derin öğrenmenin temel kavramları ve özellikleri incelenmiştir. Öneri sistemleri süreçleri ve seyahat öneri sistemleri hakkında bilgi verilmiştir ve yapay sinir ağlarının öneri sistemlerine katkıları ele alınmıştır.

Facebook'un DLRM'i ve Google'ın geniş ve derin sinir ağı ile karşılaştırıldığında, kullanıcı ve ürün verilerini model içinde ayrı ayrı işleme konusundaki benzersiz yaklaşımıyla öne çıkan, hem kullanıcıların hem de varış noktalarının özelliklerini birleştirerek, farklı değişkenler arasındaki karmaşık ilişkileri yakalayabilen güçlü, doğrusal olmayan bir model oluşturan derin ve geniş öğrenme tabanlı bir seyahat öneri sistemi geliştirilmiştir. Bu yöntem ile kullanıcı tercihleri ve davranışları daha incelikli bir şekilde anlaşılması sağlanarak daha doğru ve kişiselleştirilmiş öneriler oluşturulmuştur.

Gerçek bir seyahat acentesi verisi kullanılarak uygulanan çalışma sonucunda 50 hedef noktadan en iyi 5 öneri ile eğitim verisinde %82.37, test verisinde %82.36 doğruluk oranına ulaşılmıştır.

Derin ve geniş öneri sistemlerinin gelişmiş teknikleri ve yetenekleri, onları karmaşık verileri analiz etmek ve doğru, kişiselleştirilmiş öneriler sağlamak için güçlü bir araç haline getirmektedir. Veriler karmaşıklık ve ölçekte büyümeye devam ettikçe, derin ve geniş öneri sistemleri iş başarısını artırmada ve kullanıcı deneyimlerini iyileştirmede giderek daha önemli hale geleceği kuşkusuzdur.

Önerilen derin ve geniş öğrenme tabanlı seyahat öneri sistemi yöntemi, seyahat öneri sistemleri alanında ileriye doğru atılmış önemli bir adımı temsil etmekte ve gerçek dünya verilerine ve kullanıcı davranışına dayanan seyahat varış noktası önerisine güçlü ve yenilikçi bir yaklaşım sunmaktadır. Daha fazla araştırma ve geliştirme ile bu yaklaşım, seyahat endüstrisini dönüştürme ve dünya çapında milyonlarca insan için seyahat deneyimini iyileştirme potansiyeline sahiptir: önerilen derin ve geniş öğrenme tabanlı seyahat tavsiye sistemi yöntemi, bu alanda umut verici bir gelişmeyi temsil etmektedir.

## Teşekkür

Bu çalışmada veri sağlayıcısı olan BiletBank'a desteklerinden dolayı teşekkür ederiz.

## Kaynakça

- [1] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & Deep Learning for Recommender Systems. İçinde Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. DLRS 2016: Workshop on Deep Learning for Recommender Systems. ACM. <https://doi.org/10.1145/2988450.2988454>
- [2] Naumov, M., Mudigere, D., Shi, H.-J. M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.-J., Azzolini, A. G., Dzhulgakov, D., Mallevich, A., Cherniavskii, I., Lu, Y., Krishnamoorthi, R., Yu, A., Kondratenko, V., Pereira, S., Chen, X., ... Smelyanskiy, M. (2019). Deep Learning Recommendation Model for Personalization and Recommendation Systems (Versiyon 1). arXiv. <https://doi.org/10.48550/ARXIV.1906.00091>
- [3] Fan, W., Zhao, X., Chen, X., Su, J., Gao, J., Wang, L., Liu, Q., Wang, Y., Xu, H., Chen, L., & Li, Q. (2022). A Comprehensive Survey on Trustworthy Recommender Systems (Versiyon 1). arXiv. <https://doi.org/10.48550/ARXIV.2209.10117>
- [4] Karimova, F. (2016). A Survey of e-Commerce Recommender Systems. İçinde European Scientific Journal, ESJ (C.12, Issue 34, s. 75). European Scientific Institute, ESI. <https://doi.org/10.19044/esj.2016.v12n34p75>
- [5] Belluf, T., Xavier, L., & Giglio, R. (2012). Case study on the business value impact of personalized recommendations on a large online retailer. İçinde Proceedings of the sixth ACM conference on Recommender systems. RecSys '12: Sixth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2365952.2366014>
- [6] Amatriain, X., & Basilico, J. (2016). Past, Present, and Future of Recommender Systems. İçinde Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2959100.2959144>
- [7] Koren, Y. (2008). Factorization meets the neighborhood. İçinde Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD08: The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. <https://doi.org/10.1145/1401890.140194>



- [8] Rendle, S. (2010). Factorization Machines. İçinde 2010 IEEE International Conference on Data Mining. 2010 IEEE 10th International Conference on Data Mining (ICDM). IEEE. <https://doi.org/10.1109/icdm.2010.127>
- [9] Paparrizos, I., Cambazoglu, B. B., & Gionis, A. (2011). Machine learned job recommendation. İçinde Proceedings of the fifth ACM conference on Recommender systems. RecSys '11: Fifth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2043932.2043994>
- [10] T. Al-Otaibi, S. (2012). A survey of job recommender systems. İçinde International Journal of the Physical Sciences (C. 7, Issue 29). Academic Journals. <https://doi.org/10.5897/ijps12.482>
- [11] Albadvi, A., & Shahbazi, M. (2009). A hybrid recommendation technique based on product category attributes. İçinde Expert Systems with Applications (C. 36, Issue 9, ss. 11480-11488). Elsevier BV. <https://doi.org/10.1016/j.eswa.2009.03.046>
- [12] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. İçinde Communications of the ACM (C. 40, Issue 3, ss. 66-72). Association for Computing Machinery (ACM). <https://doi.org/10.1145/245108.245124>
- [13] Pizzato, L., Rej, T., Chung, T., Koprinska, I., & Kay, J. (2010). RECON a reciprocal recommender for online dating. İçinde Proceedings of the fourth ACM conference on Recommender systems. RecSys '10: Fourth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/1864708.186474>
- [14] Bengio, Y., Lecun, Y., & Hinton, G. (2021). Deep learning for AI. İçinde Communications of the ACM (C. 64, Issue 7, ss. 58-65). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3448250>
- [15] Peng, Y., Zhu, W., Zhao, Y., Xu, C., Huang, Q., Lu, H., Zheng, Q., Huang, T., & Gao, W. (2017). Cross-media analysis and reasoning: advances and directions. İçinde Frontiers of Information Technology & Electronic Engineering (C. 18, Issue 1, ss. 44-57). Zhejiang University Press. <https://doi.org/10.1631/fitee.1601787>
- [16] Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J., & Jing, H. (2017). Recurrent Recommender Networks. İçinde Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. WSDM 2017: Tenth ACM International Conference on Web Search and Data Mining. ACM. <https://doi.org/10.1145/3018661.3018689>
- [17] Elkahky, A. M., Song, Y., & He, X. (2015). A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. İçinde Proceedings of the 24th International Conference on World Wide Web. WWW '15: 24th International World Wide Web Conference. International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/2736277.2741667>
- [18] Song, Y., Elkahky, A. M., & He, X. (2016). Multi-Rate Deep Learning for Temporal Recommendation. İçinde Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. SIGIR '16: The 39th International ACM SIGIR conference on research and development in Information Retrieval. ACM. <https://doi.org/10.1145/2911451.2914726>
- [19] Vasile, F., Smirnova, E., & Conneau, A. (2016). Meta-Prod2Vec. İçinde Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2959100.2959160>
- [20] Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., & Sharp, D. (2015). E-commerce in Your Inbox. İçinde Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '15: The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. <https://doi.org/10.1145/2783258.2788627>
- [21] Hsieh, C.-K., Yang, L., Cui, Y., Lin, T.-Y., Belongie, S., & Estrin, D. (2017). Collaborative Metric Learning. İçinde Proceedings of the 26th International Conference on World Wide Web. WWW '17: 26th International World Wide Web Conference. International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/3038912.3052639>
- [22] Roy, S., & Guntuku, S. C. (2016). Latent Factor Representations for Cold-Start Video Recommendation. İçinde Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2959100.2959172>
- [23] Li, S., Kawale, J., & Fu, Y. (2015). Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. İçinde Proceedings of the 24th ACM International Conference on Information and Knowledge Management. CIKM'15: 24th ACM International Conference on Information and Knowledge Management. ACM. <https://doi.org/10.1145/2806416.2806527>
- [24] Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint Deep Modeling of Users and Items Using Reviews for Recommendation. İçinde Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. WSDM 2017: Tenth ACM International Conference on Web Search and Data Mining. ACM. <https://doi.org/10.1145/3018661.3018665>
- [25] Bansal, T., Belanger, D., & McCallum, A. (2016). Ask the GRU: Multi-task Learning for Deep Text Recommendations. İçinde Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2959100.2959180>
- [26] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys, 52(1), 1–38. <https://doi.org/10.1145/3285029>
- [27] Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016). Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. İçinde Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. WSDM 2016: Ninth ACM International Conference on Web Search and Data Mining. ACM. <https://doi.org/10.1145/2835776.2835837>
- [28] Wang, J., Kenthapadi, K., Rangadurai, K., & Hardtke, D. (2017). Dionysius: A Framework for Modeling Hierarchical User Interactions in Recommender Systems (Versiyon 1). arXiv. <https://doi.org/10.48550/ARXIV.1706.03849>
- [29] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System. İçinde ACM Computing Surveys (C. 52, Issue 1, ss. 1-38). Association for Computing Machinery (ACM). <https://doi.org/10.1145/3285029>

- [30] Wang, M. (2020). Applying Internet information technology combined with deep learning to tourism collaborative recommendation system. İçinde Z. Lv (Ed.), PLOS ONE (C. 15, Issue 12, s. e0240656). Public Library of Science (PLoS). <https://doi.org/10.1371/journal.pone.0240656>
- [31] Çakır, M., Ögüdücü, Ş. G., & Tugay, R. (2019). A Deep Hybrid Model for Recommendation Systems. İçinde Lecture Notes in Computer Science (ss. 321-335). Springer International Publishing. [https://doi.org/10.1007/978-3-030-35166-3\\_23](https://doi.org/10.1007/978-3-030-35166-3_23)
- [32] Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. İçinde Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems. ACM. <https://doi.org/10.1145/2959100.2959190>
- [33] Geron, A. (2019). Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media.
- [34] Chollet, F. (2022). Deep learning with python. Manning Publications.
- [35] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [36] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.
- [37] Cram101 Textbook Reviews. (2013). Studyguide for pattern recognition and machine learning by bishop, Christopher M. Cram101.
- [38] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for Activation Functions (Versiyon 2). arXiv. <https://doi.org/10.48550/ARXIV.1710.05941>
- [39] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. <https://doi.org/10.1038/323533a>
- [40] Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. İçinde Proceedings of COMPSTAT'2010 (ss. 177-186). Physica-Verlag HD. [https://doi.org/10.1007/978-3-7908-2604-3\\_1](https://doi.org/10.1007/978-3-7908-2604-3_1)
- [41] Ruder, S. (2016). An overview of gradient descent optimization algorithms (Versiyon 2). arXiv. <https://doi.org/10.48550/ARXIV.1609.04747>
- [42] Sun, R.-Y. (2020). Optimization for Deep Learning: An Overview. İçinde Journal of the Operations Research Society of China (C. 8, Issue 2, ss. 249-294). Springer Science and Business Media LLC. <https://doi.org/10.1007/s40305-020-00309-6>
- [43] Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (Versiyon 3)*. arXiv. <https://doi.org/10.48550/ARXIV.1502.03167>
- [44] Jose, G. V. (2019, February 10). Effect of Learning rate on Loss. Towards Data Science. <https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45>
- [45] Jose, G. V. (2019, February 10). Accuracy Plot. Towards Data Science. <https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45>
- [46] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. IEEE. Institute of Electrical and Electronics Engineers, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- [47] Hinton, G. E. (2012). A Practical Guide to Training Restricted Boltzmann Machines. İçinde Lecture Notes in Computer Science (ss. 599-619). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-35289-8\\_32](https://doi.org/10.1007/978-3-642-35289-8_32)
- [48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
- [49] Taylor, P. (2022, September 8). *Amount of data created, consumed, and stored 2010-2020, with forecasts to 2025*. Statista. <https://www.statista.com/statistics/871513/worldwide-data-created/>
- [50] Uzun-Per, M., Can, A. B., Volkan Gurel, A., & Aktas, M. S. (2021). *Big Data Testing Framework for Recommendation Systems in e-Science and e-Commerce Domains*. İçinde 2021 IEEE International Conference on Big Data (Big Data). 2021 IEEE International Conference on Big Data (Big Data). IEEE. <https://doi.org/10.1109/bigdata52589.2021.9672082>
- [51] Uzun-Per, M., Gurel, A. V., Can, A. B., & Aktas, M. S. (2022). *Scalable recommendation systems based on finding similar items and sequences*. İçinde Concurrency and Computation: Practice and Experience (C. 34, Issue 20). Wiley. <https://doi.org/10.1002/cpe.6841>
- [52] Sancar, S., & Uzun-Per, M. (2023). *Testing the Performance of Feature Selection Methods for Customer Churn Analysis: Case Study in B2B Business*. İçinde Computational Intelligence, Data Analytics and Applications (ss. 509-519). Springer International Publishing. [https://doi.org/10.1007/978-3-031-27099-4\\_39](https://doi.org/10.1007/978-3-031-27099-4_39)
- [53] Sancar, S., & Uzun-Per, M. (2022). *Feature Selection in Customer Churn Analysis: Case Study in B2B Business*. İçinde 2022 IEEE International Conference on e-Business Engineering (ICEBE). 2022 IEEE International Conference on e-Business Engineering (ICEBE). IEEE. <https://doi.org/10.1109/icebe55470.2022.00053>