T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TABANLI SEYAHAT ÖNERİ SİSTEMİ GELİŞTİRME

Alihan ÖZ

YÜKSEK LİSANS TEZİ

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Danışman

Dr. Öğr. Üyesi Mert BAL

Eş Danışman

Dr. Öğr. Üyesi Meryem UZUN-PER

Haziran, 2023

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TABANLI SEYAHAT ÖNERİ SİSTEMİ GELİŞTİRME

Alihan ÖZ tarafından hazırlanan tez çalışması 01.06.2023 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı, Matematik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi. Mert BAL Yıldız Teknik Üniversitesi Danışman	Dr. Öğr. Üyesi Meryem UZUN-PER İstanbul Sağlık ve Teknoloji Üniversitesi Eş-Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Mert BAL, Danışman
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi. Meryem UZUN-PER, Danışman
İstanbul Sağlık ve Teknoloji Üniversitesi _____

Prof. Dr. Fatih TAŞÇI, Bölüm Başkanı
Yıldız Teknik Üniversitesi _____

Doç.Dr. Yusuf Yaslan
İstanbul Teknik Üniversitesi _____

Danışmanlarım Dr. Öğr. Üyesi Mert BAL ve Dr. Öğr. Üyesi Meryem UZUN-PER sorumluluğunda tarafımca hazırlanan Derin Öğrenme Tabanlı Seyahat Öneri Sistemi Geliştirme başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Alihan ÖZ

İmza

Aileme,

ruh eşime ve kendime

TEŞEKKÜR

Tez çalışmam boyunca desteklerini esirgemeyen ve değerli rehberlikleri için danışmanlarım Dr. Öğr. Üyesi Mert BAL ve Dr. Öğr. Üyesi Meryem UZUN-PER'e en derin şükranlarımı sunarım. Teşvikleri, sabırları ve uzmanlıkları bu projenin başarıyla tamamlanmasında etkili oldu.

Matematik Mühendisliği Bölümüne de bana bu tezi sürdürme ve bu alanda bilgimi genişletme fırsatı verdiği için içten teşekkürlerimi sunmak isterim.

Koşulsuz sevgi ve desteklerini esirgemeyen aileme, özellikle de bana olan sürekli cesaretlendirmesi ve inancı için ruh eşime teşekkür ederim.

Ayrıca destekleri, motivasyonları ve değerli görüşleri için arkadaşlarıma ve meslektaşlarıma minnettarım.

Son olarak, bu projeyi yürütmek için bana gerekli esnekliği sağladığı için şirketim Google'a ve tez çalışması için veri seti desteği sağlayan Biletbank şirketine şükranlarımı sunmak isterim.

Yolculuğum boyunca katkılarınız ve desteğiniz için hepinize bir kez daha teşekkür ederim.

Alihan ÖZ

İÇİNDEKİLER

TEŞEKKÜR iv
KISALTMA LİSTESİ vii
ŞEKİL LİSTESİ viii
TABLO LİSTESİ x
ÖZET xi
ABSTRACT xiii
1 GİRİŞ 15
1.1 Giriş 15
1.2 Literatür Özeti 16
1.3 Tezin Amacı 18
1.4 Hipotez 18
2 MAKİNE ÖĞRENMESİ 19
2.1 Giriş 19
2.2 Yaygın Makine Öğrenmesi Metotları 20
2.3 Sonuç 31
3 DERİN ÖĞRENME 32
3.1 Giriş 32
3.2 Derin Öğrenme Nasıl Çalışır? 32
3.3 Makine Öğrenmesi ve Derin Öğrenme Arasındaki Farklar 39
3.4 Yaygın Derin Öğrenme Metotları 41
3.5 Sonuç 43
4 ÖĞRENME SİSTEMLERİ VE SEYAHAT ENDÜSTRİSİ 44
4.1 Giriş 44
4.2 Öneri Sistemleri Süreçleri 45
4.3 Seyahat Öneri Sistemleri 47
4.4 Temel Yaklaşımlar 47
4.5 Geniş ve Derin Öğrenme Sistemleri 54
4.6 Sonuç 55
5 DERİN ÖĞRENME TABANLI SEYAHAT ÖNERİ SİSTEMİ GELİŞTİRME 56
5.1 Giriş: Önerilen Yöntem 56

5.2 Veri Kümesi 58
5.3 Veri Hazırlama 62
5.4 Model Öncesi Hazırlık 72
5.5 Model 74
5.6 Sonuçlar ve Öneriler 76
6 SONUÇ **77**
KAYNAKÇA **79**
TEZDEN ÜRETİLMİŞ YAYINLAR **85**

KISALTMA LİSTESİ

ANN	Artificial Neural Network (Yapay Sinir Ağları)
CART	Classification and Regression Trees (Sınıflandırma ve Regresyon Ağaçları)
CBR	Content-based Recommendation (İçerik Tabanlı Öğrenme)
CF	Collaborative Filtering (İşbirlikçi Filtreleme)
CNN	Convolutional Neural Networks (Evrişimsel Sinir Ağları)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
DLRM	Deep Learning Recommendation Model (Derin Öğrenme Öneri Modeli)
KNN	K-Nearest Neighbor (K-En Yakın Komşu)
MLP	Multilayer Perceptron (Çok katmanlı algılayıcı)
PCA	Principal Component Analysis (Temel Bileşenler Analizi)
MNIST	Modified National Institute of Standards and Technology database (Değiştirilmiş Ulusal Standartlar ve Teknoloji Enstitüsü Veri Tabanı)
RNNs	Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
SVM	Support Vector Machines (Destek Vektör Makineleri)
TFIDF	Term frequency-inverse document frequency (Terim Frekansı Ters Belge Frekansı)
v.d.	ve diğerleri

ŞEKİL LİSTESİ

Şekil 2.1 Bir Doğrusal Regresyon Örneği 20
Şekil 2.2 Bir Lojistik Regresyon Modeli 21
Şekil 2.3 Bir Karar Ağacı Modeli 23
Şekil 2.4 İki Ağaçlı Rastgele Orman Algoritması Çalışması 24
Şekil 2.5 Doğrusal SVM'in Destek Vektörlerini Maksimize Ederek Optimal Hiper Düzlemi Belirlemesi 24
Şekil 2.6 KNN Modelinde Sınıf Belirlenmesi Örneği 25
Şekil 2.7 PCA'in Çalışma Yöntemi 26
Şekil 2.8 İki Değişkenli, Bir Gizli Katmanlı, İkili Çıktı Üreten Basit Sinir Ağı 28
Şekil 2.9 Karar Ağacı ve Gradyan Arttırma Modeli Kıyaslaması 29
Şekil 2.10 Naive Bayes Sınıflandırıcısı Çıktısı Örneği 30
Şekil 3.1 Biyolojik Beyin ve Yapay Sinir Ağları İlişkisi 33
Şekil 3.2 Sırasıyla, Sigmoid ve Relu Aktivasyon Fonksiyonları 35
Şekil 3.3 Basit Bir Sinir Ağı Mimarisi 36
Şekil 3.4 Bir Derin Öğrenme Modeli Mimarisi 36
Şekil 3.5 Öğrenme Oranının Uygun Seçilmesinin Önemi 37
Şekil 3.6 Eğitim ve Validasyon Verisinin Öğrenme Üzerine Etkisi 38
Şekil 3.7 Makine Öğrenmesi ve Derin Öğrenmede Özellik İşleme 40
Şekil 3.8 Derin Öğrenmenin Veri Boyutu-Performans İlişkisi 41
Şekil 3.9 Basit Bir İleri Beslemeli Sinir Ağı Örneği 41
Şekil 3.10 Bir Evrişimsel Sinir Ağı Örneği 42
Şekil 3.11 Basit Bir Tekrarlayan Sinir Ağı Örneği 43
Şekil 4.1 Basit Bir Öneri Sistemi Mimarisi 44
Şekil 4.2 Öneri Sistemlerinde Müşteriden Toplanabilecek Örnek Veri Kümesi 46
Şekil 4.3 Öneri Sistemlerinde Filtreleme Yöntemleri 47
Şekil 4.4 İşbirlikçi ve İçerik Tabanlı Öneri Sistemleri 49
Şekil 4.5 Matris Çarpanlara Ayırma Denklemi 51
Şekil 5.1 Derin ve Geniş Seyahat Öneri Sistemi Model Mimarisi 57
Şekil 5.2 Origin Kolonunun Dağılımı 67
Şekil 5.3 Destination Kolonunun Dağılımı 68

Şekil 5.4 Booking Kolonunun Dağılımı 68
Şekil 5.5 OperatingAirline Kolonunun Dağılımı 68
Şekil 5.6 Location_City Kolonunun Dağılımı 69
Şekil 5.7 Flight_Baggage_Dep Kolonunun Dağılımı 69
Şekil 5.8 Flight_Baggage_Arr Kolonunun Dağılımı 69
Şekil 5.9 SeatCount Kolonunun Dağılımı 70
Şekil 5.10 SellingMonth Kolonunun Dağılımı 70
Şekil 5.11 PassengerType Kolonunun Dağılımı 70
Şekil 5.12 Gender Kolonunun Dağılımı 71
Şekil 5.13 VatFlag Kolonunun Dağılımı 71
Şekil 5.14 SellingDayParts Kolonunun Dağılımı 71
Şekil 5.15 DepartureHourDayParts Kolonunun Dağılımı 72
Şekil 5.16 SellingYear Kolonunun Dağılımı 72
Şekil 5.17 CustomerAge Kolonunun Dağılımı 72

TABLO LİSTESİ

Tablo 5.1 Çalışmada Kullanılan Veri Kümesindeki Kolonların Açıklamaları 58

Tablo 5.2 Çalışmada Kullanılan Veri Kümesindeki Kolonların Veri Tipleri 60

Tablo 5.3 Çalışmada Kullanılan Veri Kümesindeki Kolonların Boş Değer ve Benzersiz Değer Sayıları 61

Tablo 5.4 Çalışmada Kullanılan Veri Kümesindeki Nümerik Kolonların Yüzdelik Değerleri 62

Tablo 5.5 Çalışmada Kullanılan Veri Kümesinin Özellik-Veri Tipi Dağılımı 64

Tablo 5.6 Veri Kümesindeki Kolonların Ön İşleme Sonrası Veri Tipleri 65

Tablo 5.7 Veri Kümesindeki Kolonların Ön İşleme Sonrası Benzersiz Değer Sayıları 66

Tablo 5.8 Kullanıcı-Öğe, Nümerik-Kategorik Özellik Dağılımı 73

Tablo 5.9 Model Sonuçları 76

ÖZET

Derin Öğrenme Tabanlı Seyahat Öneri Sistemi Geliştirme

Alihan ÖZ

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Mert BAL

Eş-Danışman: Dr. Öğr. Üyesi Meryem UZUN-PER

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 tez kapsamında, derin ve geniş sinir ağlarına ve bunların seyahat öneri sistemlerindeki uygulamalarına kapsamlı bir bakış açısı sunulmuştur. Derin öğrenme ile ilgili temel bilgilere yer verildikten sonra, en popüler öneri algoritmaları olan Google'ın Geniş ve Derin Algoritması ve Facebook'un Deep Learning Recommendation Model (DLRM) algoritmasına yer verilmiştir.

Bu tez kapsamında, 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. Bu çalışma sonucunda kullanıcılara verilen en iyi beş öneride, %88.71 doğruluk oranı yakalanmıştır.

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

ABSTRACT

Developing a Deep Learning Based Travel Recommendation System

Alihan ÖZ

Department of Mathematical Engineering

Master of Science Thesis

Supervisor: Asst. Prof. Mert BAL

Co-supervisor: Asst. Prof. Meryem UZUN-PER

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.

Within the scope of this thesis, 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.

Within the scope of this thesis, 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, 88.71% accuracy rate was achieved in the top five recommendations given to the users.

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

1
GİRİŞ

1.1 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 itibariyle 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 [31]. 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) [32], kullanıcı ve ürünlere ait yardımcı özellikleri kullanarak, öneri doğruluğunu iyileştirip soğuk başlatma sorununun üstesinden gelmektedir.

Bu tez 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. Bu tez, bu algoritmaya, faydalarına ve seyahat öneri sistemlerindeki performanslarına ilişkin değerlendirmelere bir bakış sunmaktadır.

Bu tez, 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. İkinci bolumde, makine öğrenmesi; üçüncü bölümde yapay sinir ağları ve derin öğrenmenin temelleri hakkında ayrıntılı bilgiler verilmiştir. Dördüncü bölümde, 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.

1.2 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 [1]. Öneri sistemlerinin e-ticaret üzerindeki etkisi, Karimova'nın çalışmasında açıkça görülmektedir [2]. Ayrıca, öneri sistemlerinin kullanıcı deneyimi ve iş büyümesine etkisi 2012'de Belluf v.d. [3] ve 2015'te Amatriain v.d. [4] 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 [5]. Rendle , öneri sistemlerinin ölçeklenebilirliğini ve doğruluğunu iyileştirmek için çarpanlara ayırma ve regresyon tekniklerini birleştiren bir model olan çarpanlara ayırma makinelerini (factorization machines) önermiştir [6].

Paparizos v.d. (2011), kullanıcılara iş tanımlarına göre iş önermek için içeriğe dayalı filtrelemeyi kullanan, makine tarafından öğrenilen bir iş öneri sistemi geliştirmiştir [7]. 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 [8].

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. [9] ve Balabanovi [10] gibi birkaç makale tarafından önerilmiştir. Aynı şekilde, Pizzato v.d., her iki tekniği birleştiren çevrimiçi flört için karşılıklı bir öneri sistemi geliştirmiştir [11].

Son yıllarda, derin öğrenme teknolojisi, veri madenciliği ve makine öğrenmesi topluluğunda sıcak bir konu olarak ortaya çıkan kayda değer bir ilerleme kaydetti [12]. Başlangıçta görüntü ve konuşma tanıma ve doğal dil işleme için geliştirilen derin öğrenme, bu alanlarda dikkate değer sonuçlar elde etti [13].

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-öğe ilişkilerini etkin bir şekilde yakalama ve daha yüksek katmanlarda veri temsilleri yaparak daha karmaşık soyutlamaların kodlanmasını sağlama yeteneğine sahiptir [14-25]. 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 öğe özelliklerine dayalı olarak kişiselleştirilmiş önerilerde bulunan hiyerarşik bir model olan Dionysius'u önermiştir [26]. 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 [27]. Wang içerik tabanlı ve işbirlikçi filtreleme tekniklerini birleştiren ve seyahat önerilerinin kalitesini artıran hibrit bir seyahat önerisi sistemi sunmaktadır [28]. Muhammet Ç. v.d. (2020), kullanıcı ve öğe ID yerleştirmeleri ile yardımcı bilgilerden oluşan bir derin sinir mimari ağı geliştirmiştir [29]. Aynı zamanda, Covington v.d. (2016) yardımcı bilgileri modellemek için popüler bir CBR yaklaşımı olan derin sinir ağlarını benimsemiştir [30]. 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 [31] ve Maxim Naumov v.d. (2019) büyük ölçekli uygulamalarda yüksek performans gösteren derin öğrenme öneri makinesi geliştirmiştir [32].

1.3 Tezin Amacı

Bu tezin amacı, derin ve geniş sinir ağlarına ve bunların seyahat öneri sistemlerindeki uygulamalarına kapsamlı bir genel bakış sağlamak ve derin ve geniş sinir ağlarına dayalı yeni bir seyahat öneri sistemi önermektir.

1.4 Hipotez

Derin ve geniş öğrenme yaklaşımı kullanan büyük verili bir seyahat öneri sisteminin varış noktası önerisinde yüksek performans gösterdiği ve derin ve geniş sinir ağlarının seyahat öneri sistemlerinde büyük potansiyele sahip olduğudur. Tez, bu alanda büyüyen araştırmalara katkıda bulunmayı ve seyahat öneri sistemleriyle ilgilenen araştırmacılar, uygulayıcılar ve öğrenciler için değerli bir kaynak sağlamayı amaçlamaktadır.

2
MAKİNE ÖĞRENMESİ

2.1 Giriş

Teknolojinin gelişmesiyle birlikte hayatımızın bir parçası haline gelen yapay zeka teknolojileri, her geçen gün daha popüler hale gelmekte ve yaşamımızı büyük ölçüde değiştirmektedir.

Makine öğrenmesi ve derin öğrenme bu değişimin en önemli aktörleridir. Veri modellerinin anlamlandırılmasına ve modellerin geliştirilmesine odaklanan bu iki kavram, önceki örneklerden öğrenen kademeli matematiksel öğrenme üzerine inşa edilmişlerdir. İki çeşit makine öğrenmesi modeli bulunmaktadır: denetimli (supervised) ve denetimsiz (unsupervised) öğrenme.

Denetimli öğrenme, etiketli bir veri kümesi üzerinde eğitme sürecini (training) içerirken, denetimsiz öğrenmede bu durum söz konusu değildir. Bir başka deyişle denetimli öğrenmede girdi ve çıktıların bilindiği bir veri kümesinden söz edilebilir iken denetimsiz öğrenmede çıktıların bilinmesine gerek yoktur. Denetimli öğrenmede giriş özelliklerini bu özellikler için uygun olan çıkış etiketlerine çevirmesi öğretilir ve kategorik sınıflandırma, nümerik tahminleme, görüntü ve konuşma tanıma, doğal dil işleme gibi uygulamalarda yaygın olarak kullanılır. Denetimsiz öğrenme ise giriş özelliklerinin birbirine benzeyen noktaların gruplanması sonucunda çıkış etiketlerini oluşturması öğretilir ve kümeleme, anormallik tespiti ve boyutluluk azaltma gibi uygulamalarda yaygın olarak kullanılır.

Derin öğrenme, verilerin giderek daha soyut temsillerini öğrenebilen çok katmanlı sinir ağları oluşturmaya odaklanan makine öğrenmesinin bir alt kümesidir. Derin Öğrenme modellerinin görüntü ve konuşma tanıma, doğal dil işleme ve oyun oynama gibi görevlerde oldukça etkili olduğu görülmüştür.

Makine öğrenmesi ve derin öğrenmedeki devrim, şimdiden günlük yaşamlarımız üzerinde önemli bir etkiye sahip. Amazon, Netflix ve Spotify gibi şirketlerin kullanıcıları için oluşturduğu kişiselleştirilmiş öneri sistemlerinden akıllı

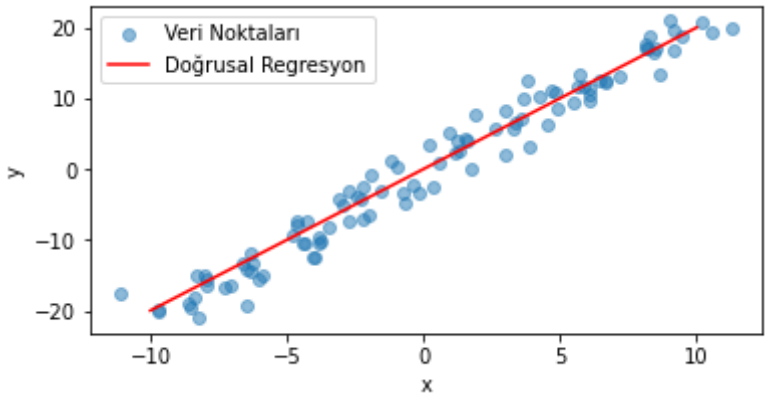
telefonlarımızdaki konuşma ve görüntü tanımaya kadar bu teknolojiler, her geçen gün hayatımızın vazgeçilmez bir parçası olmaktadırlar. Günümüzde hemen hemen her sektörde kullanılan makine öğrenmesi ve derin öğrenmenin, gelecekte de çok çeşitli sektörlerde giderek daha önemli bir rol oynaması kaçınılmazdır.

2.2 Yaygın Makine Öğrenmesi Metotları

Makine öğrenmesi çatısı altına giren birçok algoritma vardır. Bu başlık altındaki alt bölümlerde kullanılan en yaygın on algoritma ve bunların matematiksel arka planlarının kısa bir açıklaması yer almaktadır.

2.2.1 Doğrusal Regresyon

Doğrusal Regresyon, değişken ile bir veya daha fazla bağımsız faktör arasında doğrusal bir bağlantı kurarak nicel bir değişkeni tahmin etmek için kullanılan istatistiksel bir denetimli öğrenme yaklaşımıdır.



Şekil 2.1 Bir Doğrusal Regresyon Örneği

Bağımsız bir değişkenin bağımlı değişkeni doğru bir şekilde tahmin edip etmediğinin ve hangi bağımsız değişken, bağımlı değişkeni tahmin etmede önemli bir rol oynadığının belirlenmesine yardımcı olur.

Doğrusal regresyonun formülü aşağıdaki gibidir:

$$y = B_0 + B_1 X_1 + \ldots + B_p X_p + \varepsilon \quad (2.1)$$

Burada,

- y, bağımlı değişkeni,

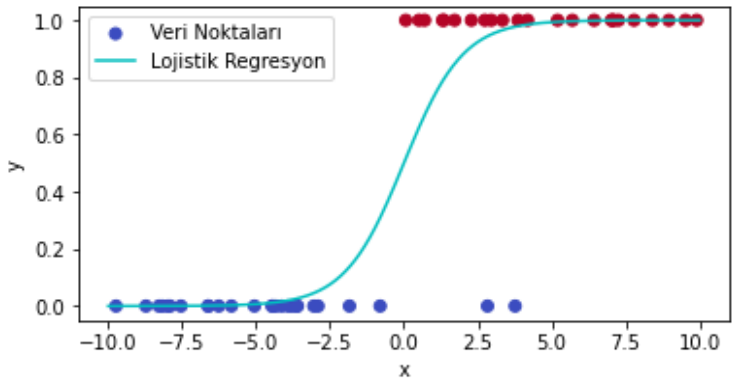
- $X_1, X_2, \ldots, X_p$, açıklayıcı değişkenleri,
- B_0, yanlılık (bias) terimini ya da sabitini,
- ε, hata sabitini ve
- $B_1, B_2, \ldots, B_p$, açıklayıcı değişkenler için eğim katsayılarını ifade etmektedir.

Doğrusal regresyonun amacı, tahmin edilen çıktı ile gerçek çıktı arasındaki kareleri alınmış hataların toplamını en aza indiren β değerlerini bulmaktır.

Basit uygulanabilir ve yorumlanabilir bir model olması, lineer ayrılan veri kümelerindeki performansının yüksek olması, aşırı öğrenmenin (overfitting) regüle edilerek azaltılabilir olması doğrusal regresyonun avantajları iken, az öğrenmeye sebep olabilecek olması (underfitting), aykırı değerlere duyarlı olmfası doğrusal regresyonun dezavantajlarıdır.

2.2.2 **Lojistik Regresyon**

Lojistik Regresyon, bağımlı değişken ve bağımsız faktörler arasında bir bağlantı kurmak için olasılıkları veya olasılıkları tahmin eden logit fonksiyonları olarak bilinen fonksiyonları kullanan, bağımlı bir değişkenin olasılığını belirlemek için kullanılan denetimli bir istatistiksel yöntemdir. Lojistik fonksiyonlar (bazen sigmoid fonksiyonlar olarak anılır), olasılıkları tahminlerde bulunmak için kullanılabilecek ikili değerlere dönüştürür. Doğrusal regresyona benzemesiyle beraber, girdi değişkenlerini çıktı değişkeninin olasılığına eşlemek için doğrusal olmayan bir lojistik fonksiyon kullanımı yönüyle ayrılmaktadır.



Şekil 2.2 Bir Lojistik Regresyon Modeli

Lojistik regresyonun formülü aşağıdaki gibidir:

$$y = \frac{1}{(1+ e^{(B_0+B_1X_1\ldots+B_pX_p)})} \tag{2.2}$$

Benzer şekilde,

- y, bağımlı değişkeni,
- e, Euler sayısını,
- $X_1, X_2, \ldots, X_p$, açıklayıcı değişkenleri,
- B_0 yanlılık ya da sapma (bias) terimini ya da sabitini ve
- $B_1, B_2, \ldots, B_p$ açıklayıcı değişkenler için eğim katsayılarını ifade etmektedir.

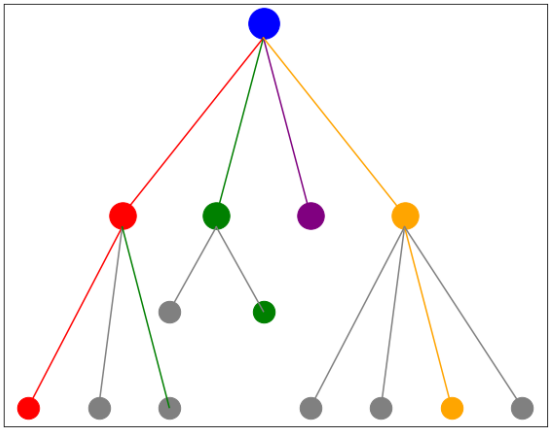
Basit uygulanabilir, yorumlanabilir ve çok etkili öğrenebilir bir model olması, uzayda sınıf dağılımları hakkında bir varsayımda bulunmaması, çoklu sınıf tahminleyebilmesi, lineer ayrılan veri kümelerindeki performansının yüksek olması, aşırı öğrenmenin regüle edilerek azaltılabilir olması lojistik regresyonun avantajları iken, bağımlı değişken ile bağımsız değişkenler arasındaki doğrusallık varsayımı, karmaşık ilişkileri ayırt edemeyerek az öğrenmeye sebep olabilecek olması lojistik regresyonun dezavantajlarıdır.

2.2.3 **Karar Ağaçları**

Karar Ağaçları, bir dizi ikili kural kullanarak hedef değeri hesaplayan denetimli bir istatistiksel öğrenme yöntemidir. Hem sınıflandırma hem de regresyon

problemlerinde kullanılabilirler. Bu nedenle sınıflandırma ve regresyon ağaçları (classification and regression trees, CART) olarak bilinirler.

Karar ağacı modelleri, verileri sürekli olarak birkaç alt uzaya bölen ve böylece her bir nihai alt uzaydaki sonuçlar mümkün olduğunca benzer olmasını sağlayan bir algoritmaya sahiptir. Bu yöntem özyinelemeli bölümleme (recursive partitioning) olarak bilinir. Tipik olarak, CART (Sınıflandırma ve Regresyon Ağaçları) tahmin modelinin karar kuralları bir ikili ağaç olarak temsil edilir. Tahmin modelinin çıktısı, regresyon veya sınıflandırma ağaçlarının kullanılmasına bağlı olarak, sürekli bir değişken veya kategorik bir değişken olabilen sonuç değişkenini tahmin etmek için bir dizi kuraldan ibarettir.



Şekil 2.3 Bir Karar Ağacı Modeli

Ön işleme sırasında veri hazırlama için daha az çaba gerektirmesi, normalleştirme ve ölçeklendirme gerektirmemesi, eksik değerlerin sonuca etkisinin düşük olması ve sezgisel olması, kategorik ve nümerik özellikleri işleyebilir olması karar ağaçlarının en önemli avantajlarıyken, girdideki küçük değişikliklerin büyük yapısal etkilere sebep olması, algoritmasının daha karmaşık olması ve bu nedenle eğitim süresinin daha uzun olması ve yine bu sebeple maliyetinin fazla olması karar ağaçlarının dezavantajları arasında yer alır.

2.2.4 **Rastgele Orman (Random Forests)**

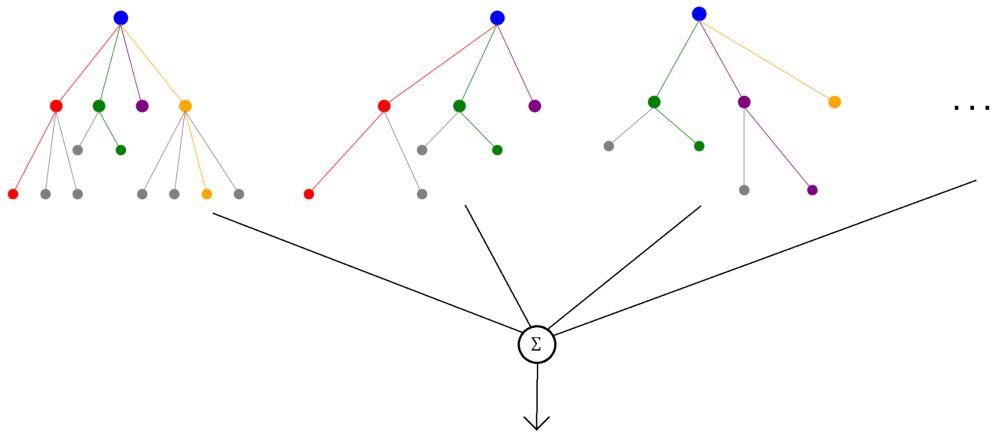
“Rastgele ormanlar, her ağacın bağımsız olarak örneklenen ve ormandaki tüm ağaçlar için aynı dağılıma sahip rastgele bir vektörün değerlerine bağlı olduğu bir ağaç tahmincilerinin bir kombinasyonudur.”[33].

Rastgele orman algoritmasının formülü aşağıdaki gibidir:

$$y = \frac{\Sigma y_i}{n} \tag{2.3}$$

burada

- y bağımlı değişkeni,
- y_i her bir karar ağacının çıktısını ve
- n ormandaki ağaç sayısını temsil eder.

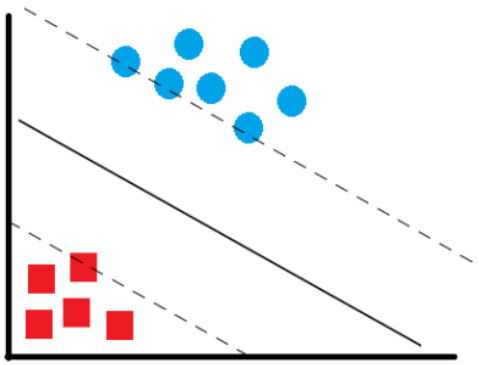


Şekil 2.4 İki Ağaçlı Rastgele Orman Algoritması Çalışması

Karar ağaçlarına kıyasla, rastgele ormanlar, çok sayıda ağacın tahminlerinin ortalamasını almaktadır ve bu sayede modelin girdilerinin model tarafından ezberlenmesini önlemektedir. Çok sayıda ağaç ile eğitim uzun zamanlar alabilir, bireysel karar ağaçlarından daha düşük sapma sergilemesi gerekmez.

2.2.5 **Destek Vektör Makineleri (Support Vector Machines)**

Destek vektör makineleri ya da kısaca SVM, hiper düzlemde verileri sınıflara ayıran çizgiler yaratan, lineer veya lineer olmayan, sınıflandırma ve regresyon problemlerini çözmek için kullanılan denetimli istatistiksel bir yöntemdir.



Şekil 2.5 Doğrusal SVM'in Destek Vektörlerini Maksimize Ederek Optimal Hiper Düzlemi Belirlemesi

Destek vektör makineleri algoritmasının formülü aşağıdaki gibidir:

$$w^T(x) + \beta_0 = 0 \tag{2.4}$$

Burada,

- w, hiper düzlemin normal vektörü,
- T, matrisin transpozu,
- x, giriş vektörü ve
- β_0 yanlılık ya da sapma terimidir.

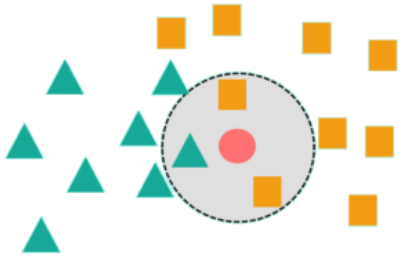
Algoritmanın amacı, hiper düzlem ile her sınıfın en yakın veri noktaları arasındaki mesafeyi maksimize etmektir.

Destek Vektör Makinelerinin yüksek boyutlu uzaylarda ve örnekten çok boyutun olduğu durumlarda efektif olması ve eğitim sırasında hafızayı etkin kullanımı avantajları iken, büyük veri setlerinde ve gürültülü hedef sınıflarını tahmin etmede dezavantajlıdır. Aynı zamanda bu algoritmanın en büyük eksiklerinden birisi de doğrudan olasılık tahminlemesinin yetersiz olmasıdır.

2.2.6 **K-En Yakın Komşu (K-Nearest Neighbour)**

K-En Yakın Komşu ya da kısaca KNN algoritması, regresyon için kullanılsa da genellikle sınıflandırma yöntemleri için tercih edilen, girdi uzayındaki K eğitim

örneğine bakarak bireysel veri noktalarının gruplandırılmasını tahminleyen denetimli istatistiki bir yöntemdir.



Şekil 2.6 KNN Modelinde Sınıf Belirlenmesi Örneği

K-En Yakın Komşu algoritmasının formülü aşağıdaki gibidir:

$$y = mod(y_1, y_2, \ldots, y_k) \tag{2.5}$$

Burada,

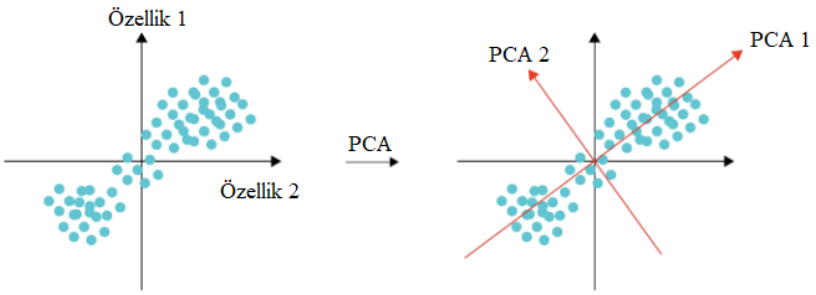
- y bağımlı değişken iken,
- $y_1, y_2, \ldots, y_k$ ise K en yakın komşunun çıktı değerleri ve
- mod en yaygın çıktı değeridir.

Hızlı implemente olması, kolay adapte edilebilmesi ve az sayıda hiper parametresi olması KNN algoritmasının avantajları iken, hafızada çok veri tüketmesi sebebiyle ölçeklenememesi, yüksek boyutlu veri girdilerinde efektif olamaması, K değişkenine bağlı olması ve aşırı öğrenmeye açık olması en büyük dezavantajlarıdır.

2.2.7 Temel Bileşenler Analizi (Principal Component Analysis)

Temel Bileşenler Analizi ya da kısaca PCA, belirli bir veri kümesine dik ve doğrusal olarak bağımsız temel bileşenler bulan, genellikle büyük veri kümelerinin boyutlarını indirmek için kullanılan, bunu yaparken büyük kümedeki bilgilerin çoğunu hala içeren daha küçük bir değişkene dönüştüren istatistiki bir boyut indirgeme yöntemidir.

Bir veri kümesindeki değişken sayısını azaltmak, doğal olarak doğruluğu azaltmaya neden olur fakat boyutsallığı azaltmanın püf noktası, bu ikisi arasındaki dengeyi koruyarak istenen sonuca ulaşmaktır. Daha küçük kümeler üzerinde gerek istatistiki işlemlerin gerekse görselleştirmelerin yapılması karmaşık ve çok değişkenli veri kümelerininkine göre çok daha kolaydır ve hızlıdır.



Şekil 2.7 PCA'in Çalışma Yöntemi

Bir veri kümesi üzerinde PCA, şu adımlar eşliğinde gerçekleştirilir:

1. Standartlaştırma,
2. Kovaryans matrisinin hesaplanması,
3. Özdeğer ve özvektörlerinin hesaplanması ve sıralandırılması,
4. k tane özdeğerin seçilerek özvektör matrisi oluşturulması,
5. Orijinal matrisin özvektör matrisi ile çarpılarak transforme edilmesi

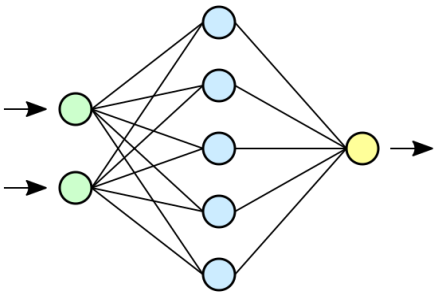
Yüksek boyutlu veri kümeleri üzerinde çalışması, birbiriyle ilişkili değerleri yok etmesi ve aşırı öğrenmeyi azaltması, daha düşük hafıza tüketerek hızda efektif olması ve modellerin daha hızlı sonuca yakınsamasını sağlaması ve görselleştirmeye olanak sağlaması PCA'in avantajları arasında yer alırken, bazı değerli verilerin kaybına neden olabilecek olması ve büyük bileşenlerin yeterince anlaşılamayabilerek doğruluğun düşmesine neden olması PCA'in dezavantajlarıdır.

2.2.8 **Yapay Sinir Ağları (Artificial Neural Networks)**

Yapay Sinir Ağları, hayvan beynini oluşturan biyolojik sinir ağlarından esinlenilerek verilerin sinir ağları içinde hareket etmesi prensibi ile oluşturulan istatistiki denetimli bir makine öğrenmesi yöntemidir.

Sinir ağlarını anlamak için matematik ve bilgisayar bilimi için sinirin ne olduğunu anlamak gereklidir. Matematik ve bilgisayar bilimi bağlamında sinirler, girdi alan, bu girdilere işlemler uygulayan ve sonucu veren şeylerin süslü bir adıdır; bir nevi fonksiyonlardır. Sinir ağları da bu sinirlerden yani fonksiyonlardan ya da öğrenme birimlerinden meydana gelen, birbirlerine besleme yaparak *öğrenme*yi meydana getiren ağların genel adıdır.

Genellikle *derin öğrenme* ve *yapay sinir ağları* birbirinin yerine kullanılan iki kavramdır. Derin öğrenmedeki derin kavramı sinir ağlarındaki katman derinliğine atıfta bulunmakla birlikte girdileri ve çıktıları içerecek üçten fazla katmandan oluşan bir sinir ağı, bir derin öğrenme algoritması olarak kabul edilebilir.



Şekil 2.8 İki Değişkenli, Bir Gizli Katmanlı, İkili Çıktı Üreten Basit Sinir Ağı

Bir gizli katmanlı basit bir sinir ağı için matematiksel formül şöyledir:

$$y = g_2(w_2 * g_1(w_1 * x + b_1) + b_2) \tag{2.6}$$

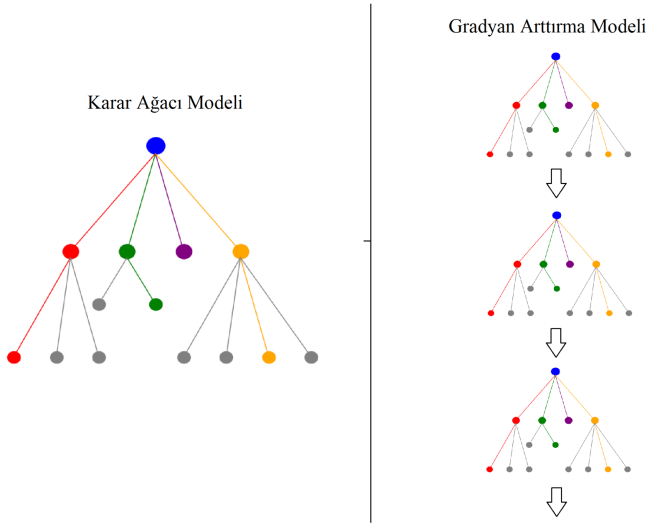
Burada

- x girdi vektörü,
- y bağımlı değişken,
- w_1 ve w_2 girdi ve gizli katmanların ağırlık matrisleri,
- b_1 ve b_2 yanlılık veya sapma vektörleri,
- g_1 ve g_2 gizli ve çıktı katmanlarının aktivasyon fonksiyonlarını ifade etmektedir.

Büyük veriler ile çalışmadaki performansı, hesaplama gücü, birçok uygulamada sorunlara çözüm üretebilir olması ve sürekli öğreniyor olması yapay sinir ağlarının en önemli avantajları iken, tam bir kapalı kutu olması yani gizli katmanların nasıl öğrendiğinin bilinememesi, geliştirme süresinin uzun olması, büyük olmayan veriler ile çalışmasının yetersiz olması ve hesaplama maliyetinin yüksek olması en temel dezavantajlarıdır.

2.2.9 **Gradyan Arttırma (Gradient Boosting)**

Gradyan arttırma, diğer arttırma (boosting) metotları gibi birçok zayıf modelin birleştirilerek daha iyi performans elde etmeleri için kullanılan istatistiki denetimli bir topluluk (ensemble) makine öğrenmesi yöntemidir. Genellikle ağaç modelleri ile regresyon ve sınıflandırma problemleri için kullanılırlar. Önceki artırma yaklaşımlarında olduğu gibi, gradyan arttırma bir ağaç modeli aşamalar halinde oluşturulur, ancak keyfi bir türevlenebilir kayıp fonksiyonunun optimizasyonuna izin vererek diğer yöntemlerden ayrışır.



Şekil 2.9 Karar Ağacı ve Gradyan Arttırma Modeli Kıyaslaması

Gradyan arttırma algoritmasının formülü aşağıdaki gibidir:

$$y = \Sigma\alpha y + \Sigma f_j(x) \tag{2.7}$$

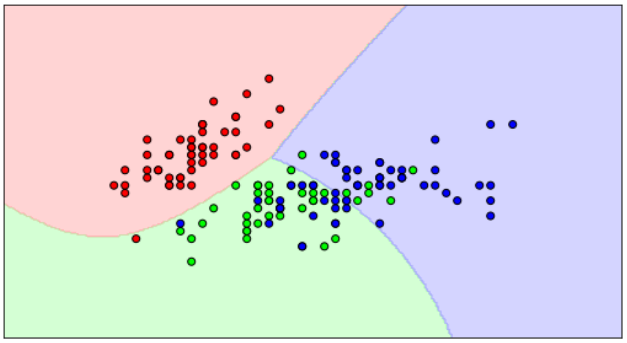
Burada

- y bağımlı değişken iken,
- i. zayıf öğrenici α ile
- α'nin çıktısı y_i ile,
- x'in j. zayıf öğrenicisi $f_j(x)$ ile ifade edilmiştir.

Genellikle diğer arttırma yöntemlerine göre daha yüksek doğruluklu sonuçlar üretmesi, büyük veri kümelerinde eğitim süresinin düşük olması, kategorik özelliklerin işlenmesine izin vermesi, eksik değerleri halletmesi gradyan arttırma modelinin en büyük avantajları iken, aşırı öğrenmeye meyilli olması (fakat bu durum regularize edilerek ya da düşük bir öğrenme oranı ile çözülebilir) hesaplama açısından maliyetinin yüksek olabilmesi, özellikle CPU'da eğitiminin uzun sürebilecek olması ve son modeli yorumlamanın zorluğu bu modelin dezavantajları arasında yer alır.

2.2.10 Naive Bayes Sınıflandırıcısı

Naive Bayes sınıflandırıcısı, koşullu olasılıklardan elde edilen Bayes Teoremine dayanan, sınıflandırma problemleri için kullanılan, istatistiki denetimli basit bır makine öğrenme yöntemidir.



Şekil 2.10 Naive Bayes Sınıflandırıcısı Çıktısı Örneği

Naive Bayes sınıflandırıcısının algoritmasının formülü aşağıdaki gibidir

$$P(y|x_1,\ x_2,\ \ldots,\ x_n) = \frac{P(y)\ P(x_1|y)\ P(x_2|y)\ \ldots\ P(x_n|y)}{P(x_1, x_2, \ldots, x_n)} \tag{2.8}$$

Burada

- $P(y|x_1, x_2, ..., x_n)$, bağımlı değişken y'nin, $x_1, x_2, ..., x_n$ verildiğindeki olasılığı,
- $P(y)$, y'nin önceki olasılığı ve
- $P(x_i|y)$, y verildiğinde x_i'nin olasılığını ifade etmektedir.

Hızlı çalışması, her ne kadar basit bir model olsa da çok sınıflı tahmin problemlerini çözmek için uygun olması, özelliklerinin bağımsız olduğu varsayımı doğru olduğu durumlarda diğer modellerden iyi performans gösterebilecek olması ve çok az eğitim verisi ile çalışabilmesi ile kategorik girdi değişkenleri için optimum çalışması Naive Bayes algoritmasının en önemli avantajlarıyken, gerçek hayatta karşımıza çıkması daha az mümkün olsa da özelliklerin bağımsız olduğu varsayımı üzerine olması ve bu nedenle gerçek hayatta uygulanabilirliğinin düşük olması, test veri setindeki kategorisi eğitim veri setinde bulunmayan kategorik bir değişkene sıfır olasılık atadığı sıfır-frekans sorunu, bazı durumlarda tahminlerinin yanlış olması bu modelin en büyük dezavantajlarıdır.

2.3 Sonuç

Yapay zeka, makine öğrenmesi ve sinir ağlarının gelişmesiyle birlikte hemen hemen her sektörde prosedürlerin otomatikleştirmesinden, süreçlerin iyileştirilmesine ve büyük verinin anlamlandırılmasında yardımcı oldu ve olmaya devam etmekte. Makine öğrenmesi ve sinir ağları ile ilgili yapılan çalışmalar ilerlemeye devam ettikçe, yaşamlarımızı asla hayal bile edemeyeceğimiz şekillerde geliştiren, giderek daha karmaşık algoritmalar ve uygulamalar görmek kaçınılmaz olmuştur.

Denetimli ve denetimsiz öğrenmeyi ve bu alanlardaki en popüler algoritmaları kapsayan bu bölümde makine öğrenmesi temelleri ele alınmıştır. Bir sonraki bölümde yaygın kullanılan derin öğrenme algoritmalarından ve bu algoritmaların nasıl çalıştığından, matematiksel arka planından ve derin öğrenme dünyasında yaygın olarak kullanılan terimlerden bahsedilecektir.

3
DERİN ÖĞRENME

3.1 Giriş

Derin öğrenme, 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. Günümüzde birçok teknoloji şirketi, görüntü tanımlama, sahtecilik tespiti, doğal dil işleme, ses tanıma, sürücüsüz araçlar, sağlık, kişiselleştirme, çeviri, sanal asistan gibi birçok uygulamada derin öğrenmeden faydalanmıştır ve faydalanmaya devam etmektedir. Gelecekte, derin öğrenme ve sinir ağlarının çeşitli uygulamalarda daha da büyük bir etkiye sahip olması beklenmektedir.

Bu bölümde yaygın derin öğrenme metotları, derin öğrenme çalışma prensibi ve makine öğrenmesi ile arasındaki farklara yer verilecektir.

3.2 Derin Öğrenme Nasıl Çalışır?

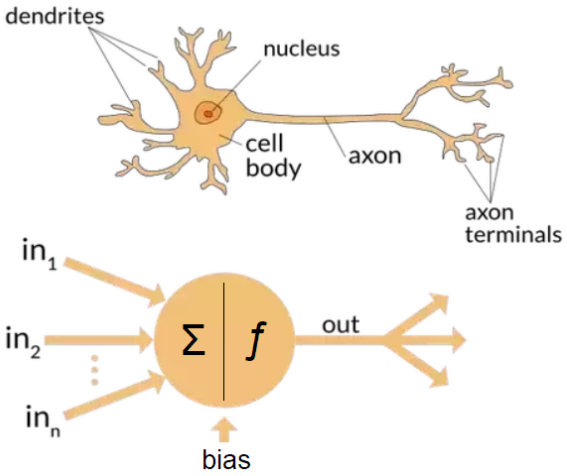
Derin öğrenmenin nasıl çalıştığını anlamak için derin öğrenmenin ilham kaynağı olan hayvan beyninin nasıl çalıştığını anlamak gereklidir. Çünkü yapay sinir ağları, esasında, organik beyin ağlarının temel işlevlerini taklit eder. Yapay sinir ağları, biyolojik öncülleri ile karşılaştırılarak daha iyi öğrenilebilir.

3.2.1 Biyolojik Beyin ve Öğrenme

"Beyin, öğrenme ve problem çözme için en etkili araç olduğu için, makine öğrenmesi algoritmalarının geliştirilmesi için doğal bir ilham kaynağı olmuştur" [34]. Beynin bilgiyi işleme ve analiz etme şeklini taklit eden algoritmalar, deneyimlerden ders çıkarmak, yeni durumlara uyum sağlamak ve tahminlerde bulunmak için tasarlanır. Derin öğrenme de beyinden ilham alan bir hesaplama yöntemidir yani beynin temel yapısını ve işlevselliğini taklit eden hesaplamalı bir yaklaşımdır. Bu nedenle, derin öğrenmenin nasıl çalıştığını anlamak için derin öğrenmenin ilham kaynağı olan hayvan beyninin nasıl çalıştığını anlamak

gereklidir. Yapay sinir ağları, biyolojik öncülleri ile karşılaştırılarak daha iyi öğrenilebilir.

"Biyolojik beynin yapı taşları nöronlardır. Nöronlar, akson ve dendritlerden meydana gelmişlerdir. Dendritler, çok ince olan bir hücrenin dallanan uzantılarıdır. Aksonlar ise hücre gövdesinden büyüyen uzantılardır. Çoğu nöronun dendriti alıcı görevi görürken, aksonlar mesajları taşır. Çoğu sinaps, bir nöronun aksonunda başka bir nöronun dendritine impulsların iletilmesini içerir. Aksiyon potansiyeli, yeterli büyüklük ve süredeki voltaj değişikliğine yanıt olarak bir nöron tarafından üretilen elektrokimyasal bir darbedir. Sinaptik bağlantılar, akson boyunca hızla ilerlerken bu potansiyel tarafından etkinleştirilir" [35].



Şekil 3.1 Biyolojik Beyin ve Yapay Sinir Ağları İlişkisi [36]

Bu bilgiler ışığında yapay sinir ağlarının da genellikle birbirine bağlı birimlerin ve nöronların bir koleksiyonundan oluştuğu söylenebilir. Nöronlar, sinir ağları içerisinde nümerik değerlerin bir temsilidir. İki nöron arasındaki herhangi iki bağlantıya biyolojik beyindeki akson denebilir. Nöronlar arasındaki bağlantılar, yine sayısal değerlerden başka bir şey olmayan sözde ağırlıklar tarafından gerçekleştirilir.

Bir yapay sinir ağı öğrenmeye devam ederken, bağlantının gücü gibi nöronlar arasındaki ağırlıklar da değişir ve optimum ana ulaşıncaya kadar bu işlemi yineler.

3.2.2 **Derin Öğrenmenin Mimarisi**

Derin öğrenme mimarisi birkaç katmandan oluşur; **girdi katmanı, gizli katmanlar, çıktı katmanı** [37].

Girdi katmanı (input layer), eğitim kümesindeki özellikler kadar nöronlara sahiptir ve sinir ağının öğrendiği verileri alan katmandır. Başka bir deyişle, her girdi nöronu eğitim kümesindeki bir özelliği temsil eder [38].

Çıktı katmanı (output layer), sinir ağının sonunda yer alan ve model çıktısını üreten katmandır. Bu katmandaki girdiler, çıktı katmanındaki nöronların değerlerini temsil eder [39]. Örneğin, sınıflandırma modelleri için bir adet çıktı bulunması, bu modelin sonucunun ikili (binary) bir sonuç üreteceğini gösterebileceği gibi, n adet çıktı bulunması bu modelin çok sınıflı bir model olduğunu gösterir.

Bir tahmin vektörü y elde etmek için, ağın giriş ve çıkış katmanları arasındaki katmanlarda gerçekleştirdiği belirli matematiksel işlemleri gerçekleştirmesi gerekir. Bu işlemlerin gerçekleştiği katmanlara **gizli katmanlar** denir [40]. Modelin derinliğine ve istenen sonuçlara bağlı olarak girdi katmanı sayısı yani modelin derinliği ve her bir girdi katmanındaki girdi sayısı değişiklik gösterebilir.

Her iki katman arasında bağlantıyı sağlayan sayısal değerler bulunmaktadır. Bu sayısal gösterimlere **ağırlık** ismi verilir [41]. 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 Wij 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 [37].

İki girdiden oluşan X girdi vektörünün verildiği ve üç çıktının tahminlendiği basit bir model düşünelim.

$$X = (X_1, X_2) \tag{3.1}$$

$$W = (w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}) \tag{3.2}$$

X girdi matrisi (3.1) ile, W ağırlık matrisi ise (3.2) ile ile ifade edilecektir. Şu ana kadarki bilgiler ışığında y bağımlı değişkeninin şu şekilde oluştuğu söylenebilir:

$$y = x_1 w_{11} + b_{11} + x_2 w_{21} + b_{21},$$
$$x_1 w_{12} + b_{12} + x_2 w_{22} + b_{22},$$
$$x_1 w_{13} + b_{13} + x_2 w_{23} + b_{23}$$
$$= (z_1, z_2, z_3) \quad (3.3)$$

Oluşan z vektörü her ne kadar y'yi ifade etmede kullanılabilecek 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 [42]. Genellikle, tanh, relu, sigmoid ve softmax adında dört aktivasyon fonksiyonu kullanılır.

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:

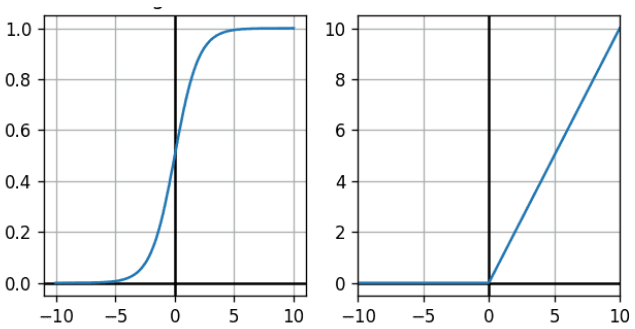
$$h = (f(z_1), f(z_2), f(z_3)) \quad (3.4)$$

İfade (3.4)'te, h, çıktı katmanını, f fonksiyonu aktivasyon fonksiyonunu, ziise girdi değerlerini temsil etmektedir.

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

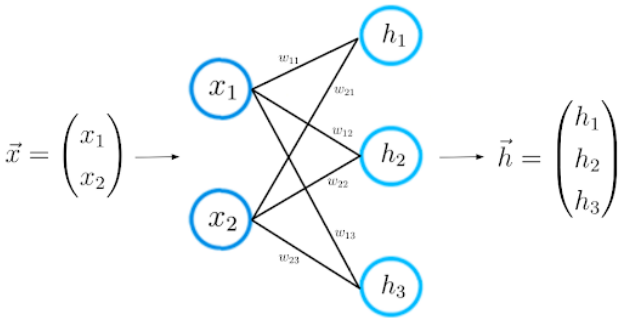
$$h = (f(z_1), f(z_2), f(z_3)) \quad (3.5)$$

İfade (3.5)'te, h, çıktı katmanını, f fonksiyonu aktivasyon fonksiyonunu, eeuler sayısını ve zi girdi değerlerini temsil etmektedir.



Şekil 3.2 Sırasıyla, Sigmoid ve Relu Aktivasyon Fonksiyonları

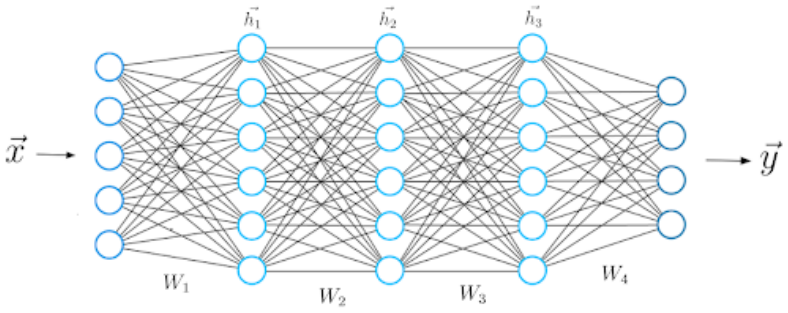
Bir modelde en önemli etkiyi sağlayan ağırlıklardır. Eğitim süresince bazı nöronlar gerçek değere yakınlaşmasını bazılarının ise uzaklaşmasını sağlamaktadır. Bunun sonucunda da ağırlıkları tayin edilmektedir.



Şekil 3.3 Basit Bir Sinir Ağı Mimarisi

Bu basit örnekte iki katmanlı sinir ağının nasıl bir matematiğe sahip olduğu anlatıldı. Daha derin bir öğrenme için oluşturulabilecek aşağıdaki gibi bir model için çıktı katmanı genel olarak aşağıdaki gibi hesaplanır:

$$y = f(f(f(f(X^T W_1 + b_1)W_2 + b_2)W_3 + b_3)W_4 + b_4) \quad (3.6)$$



Şekil 3.4 Bir Derin Öğrenme Modeli Mimarisi

Model ileri doğru beslendikten (**ileri besleme** ya da forward propagation [38]) sonra her adım sonrası gerçekte olması gereken değerler ile eğitim sürecinin sonuçların uzaklığının hesaplandığı **kayıp fonksiyonları** (loss functions [39]) 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:

$$y = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \epsilon C_c} log(p_{model}[y_i \epsilon C_c]) \quad (3.7)$$

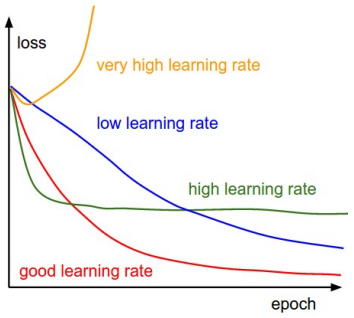
İfade (3.7)'de N gözlem sayısını, C kategorilerini ifade etmekdir: $1_{y_i \epsilon C_c}$, c kategorisine ait i. gözlemin gösterge işlevi ve $p_{model}[y_i \epsilon C_c]$, i. gözlemin c. kategoriye ait olması için model tarafından tahmin edilen olasılığıdır.

Kayıp fonksiyonun sonucuna göre modelde düğümler arasında atanan her bir ağırlık değeri ve bias değeri güncellenir. Bu güncelleme işlemine **geri besleme** (backpropagation [43]) 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 [44]) fonksiyonu kullanılır. En basit haliyle gradyan azalma formülü aşağıdaki gibidir:

$$\theta_{t+1} = \theta_t - \alpha * \nabla J(\theta t) \quad (3.8)$$

İfade (3.8)'de θ_t, t zamanındaki ağırlık vektörünü, α 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**dır (learning rate [45]). Öğrenme oranı, her yinelemede maliyet fonksiyonundaki minimizasyon miktarı olarak tanımlanır. Basit bir ifadeyle, maliyet fonksiyonunun minimum noktasına indiğimiz oran, öğrenme oranıdır ve en iyi sonuca en etkili şekilde ulaşmanın anahtarlarından birisi bu oranı optimize etmektir.



Şekil 3.5 Öğrenme Oranının Uygun Seçilmesinin Önemi [46]

Yüksek öğrenme oranlarını kullanarak öğrenme hızını arttırmak için ise normalizasyon yöntemlerinden **yığın normalizasyonu** (batch normalization [47]) kullanılabilmektedir, formülü şu şekildedir:

$$\hat{y} = \frac{(x - \mu)}{\sqrt{(\sigma\text{^}2 + \varepsilon)}} * \gamma + \beta \tag{3.9}$$

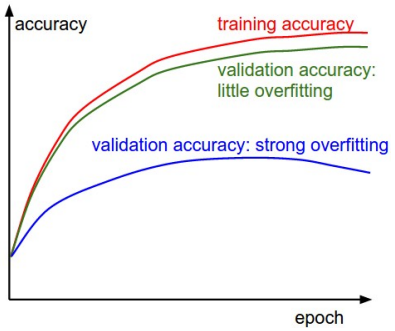
İfade (3.9)'da x katmanın girişi, μ ve σ, giriş yığınının ortalama ve standart sapması, ε, sıfıra bölünmeyi önlemek için kullanılan küçük bir değer, γ ve β, modelin her yığın için en uygun ölçeği ve ortalamayı öğrenmesini sağlayan öğrenilebilir ölçekleme (scaling) ve kaydırma (shifting) parametreleridir.

Modeldeki her ağırlığın gradyanı hesaplamak için model eğitilirken geri besleme kullanılır; hesaplanan gradyanlara göre modelin ağırlıklarını güncellemek için ise bir **optimizasyon algoritması** (optimization function [48]) 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 optimizer daha hızlı yakınsama sağlayabilir ve çeşitli uygulamalar için derin öğrenme modellerinde yaygın olarak kullanılır. Adam'ın formülü şu şekildedir:

$$\theta_{t+1} = \theta_t - \frac{\alpha \, x \, mt}{\sqrt{(vt + \epsilon}} \tag{3.10}$$

İfade (3.10)'da θ_t, t zamanındaki ağırlık vektörü, α öğrenme oranı, mt ve vt, gradyanların birinci ve ikinci moment tahminleri, ϵ ise, sıfıra bölünmeyi önlemek için kullanılan küçük bir değerdir.



Şekil 3.6 Eğitim ve Doğrulamanın Öğrenme Üzerine Etkisi [49]

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:

$$F = \frac{2\,x\,TP}{2\,x\,TP + FP + FN} \tag{3.11}$$

İfade (3.11)'te TP gerçek pozitifler değer sayısı iken, FP ve FN, yanlış pozitif ve negatif sayılarıdır. Gerçek pozitifler (TP), gerçekten pozitif olan ve model tarafından doğru bir şekilde pozitif olarak tanımlanan örnekleri ifade eder. Yanlış pozitifler (FP), aslında negatif olan ancak model tarafından yanlışlıkla pozitif olarak tanımlanan durumları ifade eder. Yanlış negatifler (FN), aslında pozitif olan ancak model tarafından negatif olarak tanımlanan örnekleri ifade eder.

Model için bir diğer önemli husus ise **öğrenme oranı**dır (learning rate [45]). Öğrenme oranı, her yinelemede maliyet fonksiyonundaki minimizasyon miktarı olarak tanımlanır. Basit bir ifadeyle, maliyet fonksiyonunun minimum noktasına indiğimiz oran, öğrenme oranıdır ve en iyi sonuca en etkili şekilde ulaşmanın anahtarlarından birisi bu oranı optimize etmektir.

Son olarak, modelleme esnasında sıklıkla kullanılan şu 3 terimi de açıklamak gereklidir: yığınlar (batch [50]), devir (epoch [51]) ve bırakma (dropout [52]). 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 yığınlara bölerek eğitilmeye müsait olmasıdır. Devir, her bir öğrenme döngüsüne verilen addır. İleri ve geri beslemenin bittiği her yinelemede bir epoch tamamlanmış olmaktadır. Bırakma ise gizli katmanlardaki rastgele nöronların ilgili epochta eğitime dahil edilmeyip bu şekilde modelin aşırı öğrenme konusunda düzenlileştirme yapıldığı tekniktir.

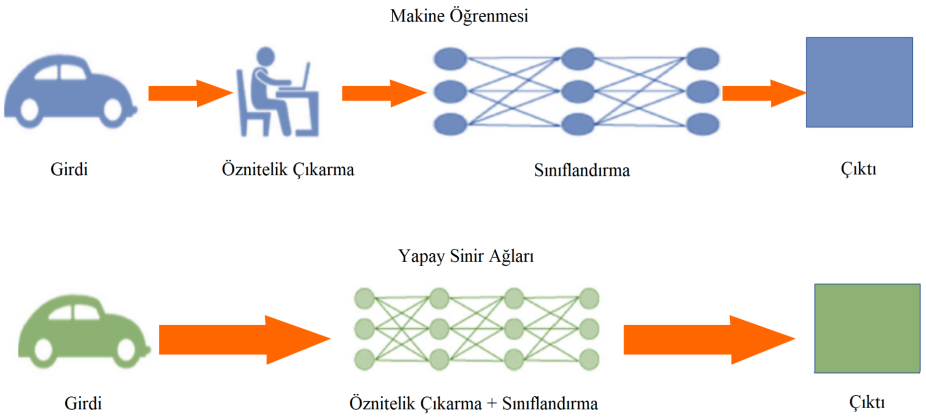
3.3 Makine Öğrenmesi ve Derin Öğrenme Arasındaki Farklar

Derin Öğrenme, makine öğrenmesinin bir alt kümesi olduğu halde, derin öğrenmeyi makine öğrenmesinden ayıran birçok temel fark bulunmaktadır.

Aslında ikisi de makinenin verilen girdileri bir çıktıya dönüştürmesi için istatistiki yöntemleri kullandığı süreçleri kapsamasına rağmen, derin öğrenme, kullandığı veri ve öğrenme teknikleri dolayısıyla geleneksel makine öğrenmesi yöntemlerinden ayrılmaktadır.

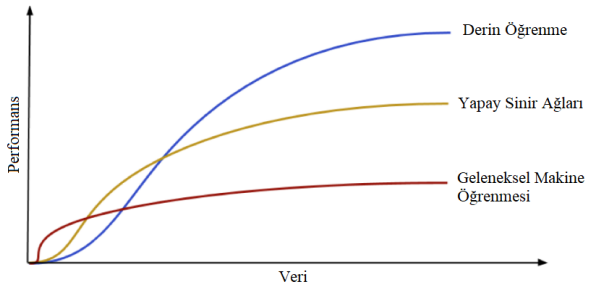
Geleneksel makine öğrenmesinde, kullanılan veri genel itibariyle belirli bir yapıya sahip ve etiketli verilerden oluşmaktadır. Bir başka deyişle, modelin çalışabilmesi için, modelden önceki süreçte, bazı girdilerin, belirli işlemlerden geçtiği bir süreci takip etmesi gerekmektedir. Derin öğrenmede ise makine öğrenmesinde olan bazı tipik ön işleme adımlarını atlayıp modeli oluşturmak mümkündür. Bu algoritmalar, insan müdahalesine bağımlılığı ortadan kaldırarak özellik çıkarmayı otomatik hale getirebilirler ve metin ve görseller gibi yapısal olmayan verileri kolaylıkla işleyebilirler.

Örneğin, sayıların bulunduğu MNIST veri kümesindeki eğitim verilerini kullanarak hangi görselin hangi sayıyı gösterdiğini derin öğrenme algoritmalarından evrişimsel sinir ağları (convolutional neural networks or CNN) ile etiketlemek mümkündür. Derin öğrenme algoritmaları, hangi özelliklerin sayıları belirlemede daha ön plana çıktığını (mesela dairesel bir şekil olup olmaması, kaç tane köşesinin olduğu gibi) belirlemesi mümkün iken, bu durum makine öğrenmesinde uzmanların modelden önce girdileri ön işlemeye tabi tutması ile gerçekleşir.



Şekil 3.7 Makine Öğrenmesi ve Derin Öğrenmede Özellik İşleme

Öte yandan, bir fonksiyonun yerel minimum/maksimum değerini bulmak için kullanılan yinelemeli birinci dereceden optimizasyon algoritması dereceli azalma (gradient descent) ve ağdaki nöronların ağırlıklarını ve yanlılık ayarlayarak sinir ağlarını eğiten geri yayılım (backpropagation) ile yeni görsellerin algoritma tarafından tahmin edilme doğruluğunu arttırmaktadır.



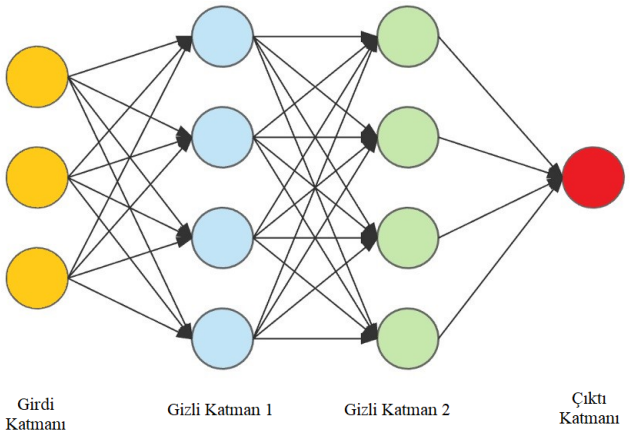
Şekil 3.8 Derin Öğrenmenin Veri Boyutu-Performans İlişkisi

Bunlara ek olarak, derin öğrenme hesaplama gücüne daha çok ihtiyaç duysa da gerek büyük veri ile öğrenmesindeki etkililiği gerekse etiketli verilerde olduğu gibi etiketsiz verilerde de doğruluğu tatmin edici sonuçlar vermesinden dolayı, birçok uzmanın favori modellerinden birisi olmuştur.

3.4 Yaygın Derin Öğrenme Metotları

Derin Öğrenme çatısı altına giren birçok algoritma vardır. Bu başlık altındaki alt bölümlerde kullanılan en yaygın üç algoritmaya ait kısa bir açıklaması yer almaktadır.

3.4.1 İleri Beslemeli Sinir Ağları (Feedforward Neural Networks)



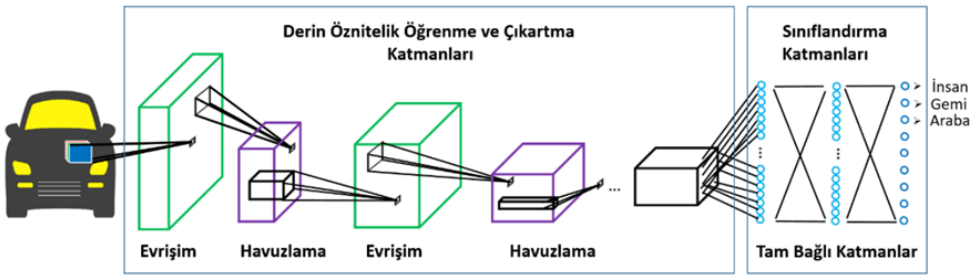
Şekil 3.9 Basit Bir İleri Beslemeli Sinir Ağı Örneği

İleri beslemeli sinir ağları, girdi katmanından çıktı katmanına tek yönlü bir yolda bilgi işlemek için tasarlanmış bir tür yapay sinir ağlarıdır.

Birbirine bağlı çok sayıda düğüm katmanından oluşur. Her katman bir önceki katmandan gelen verileri işler ve çıktı katmanına ulaşana kadar bir sonraki katmana gönderir. Her katmanın düğümleri genellikle vektörler halinde gruplanır ve her vektör, gelen verinin farklı bir özelliğini veya karakteristiğini temsil eder. Ağ, beklenen ve gerçek çıktı arasındaki farkı en aza indirmek için düğüm bağlantılarının ağırlıklarını değiştiren geri yayılım (backpropagation) gibi denetimli bir öğrenme yöntemi kullanılarak eğitilir. İleri beslemeli sinir ağları, sınıflandırma, regresyon ve görüntü ve ses tanıma uygulamalarında yaygın olarak kullanılmaktadır.

3.4.2 Evrişimsel Sinir Ağları (Convolutional Neural Networks (CNNs))

Evrişimsel Sinir Ağları (CNN), öncelikle görüntü tanımlama ve sınıflandırma gibi görüntü işleme uygulamalarına yönelik derin sinir ağlarıdır.



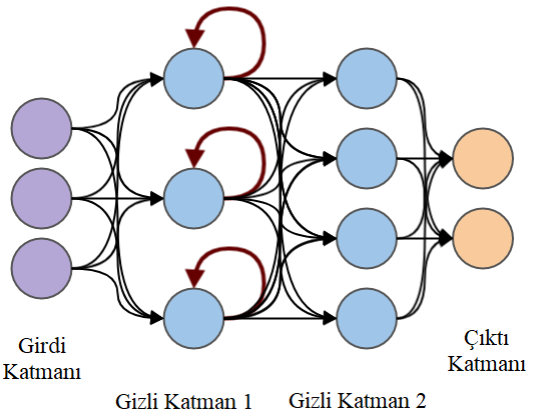
Şekil 3.10 Bir Evrişimsel Sinir Ağı Örneği

Evrişimli katmanlar, havuzlama katmanları ve tamamen bağlantılı katmanlar dahil olmak üzere birkaç nöron katmanına sahiptir. Evrişimli katmanlar, giriş resminden özellik çıkarımından sorumlu olan ağın temel yapı öğeleridir. Birleştirme katmanları, özelliklerin boyutsallığını azaltmak için evrişimli katmanların çıktısını alt örneklerken, tamamen bağlı katmanlar resmi tanımlar. Eğitim sırasında ağ, ağın çıktısının istenen çıktıyla karşılaştırıldığı ve hatanın ağırlıkları ve sapmaları değiştirmek için kullanıldığı geri yayılım (backpropagation) tekniği kullanarak nöronlarının ağırlıklarını ve sapmalarını optimize etmeyi öğrenir. CNN'ler makinenin göz ve kulakları olarak düşünülebilir. CNN'ler bilgisayar görüşü, otonom sürüş, tıbbi görüntüleme ve gerçek zamanlı videoda nesne tanıma gibi çok çeşitli uygulamalarda kullanılmaktadır.

3.4.3 Tekrarlayan Sinir Ağları (Recurrent Neural Networks (RNNs))

Tekrarlayan Sinir Ağları, metin, ses ve zaman serisi verileri gibi sıralı verileri işlemesi amaçlanan sinir ağlarıdır. RNN'ler, ileri beslemeli sinir ağlarının aksine, mimarilerinde bilginin dizideki bir aşamadan diğerine iletilmesine izin veren döngüler içerir. RNN'ler, verilerdeki geçici ilişkileri ve örüntüleri yakalayabilir ve bu da onları dil modelleme, konuşma tanıma ve makine çevirisi gibi uygulamalar için uygun hale getirir.

Derin RNN'lerde meydana gelebilecek *yok olan gradyan sorununu* (vanishing gradient problem) ele almak için Uzun Kısa Süreli Bellek (LSTM) ve Kapılı Tekrarlayan Birim (GRU) gibi varyasyonlarla desteklenen RNN'ler, doğal dil işlemede ve konuşma tanımada yaygın kullanım görmektedir.



Şekil 3.11 Basit Bir Tekrarlayan Sinir Ağı Örneği

3.5 Sonuç

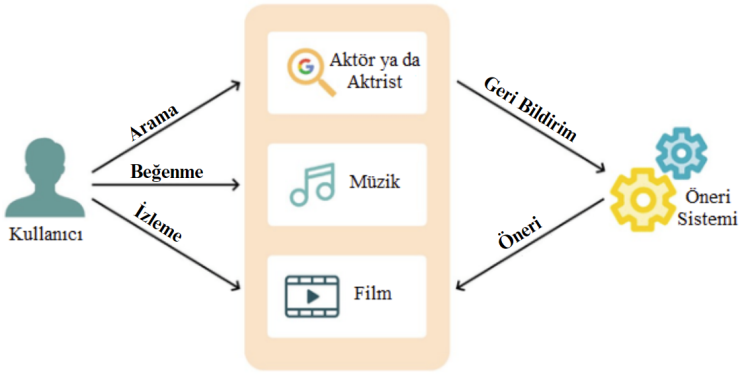
Hayvan beynini ilham alarak tasarlanarak hayatımıza giren ve birçok sektörde geniş kapsamlı olarak kullanılmaya başlanan derin öğrenme, insanların teknolojiyle etkileşim biçimini derinden değiştiren bir teknoloji oldu. Büyük hacimli verilerden içgörüler çıkarma kapasitesi, her alanda önemli ilerlemeler sağladı. Her ne kadar modellerin yorumlanabilirliği (gizli katmanlar ve kara kutu problemi) az olsa ve eğitim için büyük veriye ihtiyaç duyulsa da, derin öğrenmenin günlük yaşamlarımız üzerindeki etkilerinin teknoloji ilerledikçe artması kaçınılmazdır.

4
ÖNERİ SİSTEMLERİ VE SEYAHAT ENDÜSTRİSİ

4.1 Giriş

Her geçen gün veri yoğunluğunun arttığı ve günlük yaklaşık 118 zettabayt (129,742,372,077,568 gigabyte) verinin üretildiği günümüzde [53], 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.

Öneri sistemleri günümüzün en popüler konusu olmasına rağmen mazisi 70'li yıllara dayanmaktadır [54]. O zamanlarda internetin yaygınlaşmaması ve veri miktarının yetersiz olması sebebiyle teoriler pratiğe başarıyla uygulanmaktan çok uzaktaydı. Yapılan öneriler kişiselleştirilmemiş, yetersiz ve yanlış olması sebebiyle pek tercih edilmemekteydi. Ancak 1990'ların ortasından itibaren öneri sistemleri ayrı bir araştırma dalı haline gelmiştir [55]. Günümüzde ise internet ve sosyal medyanın ortaya çıkışı ile yaşanan teknolojik devrim sayesinde birçok firma, makine öğrenmesi ve anlamsal yöntemler ile metin analizi gibi farklı teknikler ile kompleks öneri sistemleri oluşturdu ve kullanıcılara, geçmiş davranışları veya kullanıcıların tercihlerine göre çeşitli başarılı öneriler sunarak katma değer elde etti ve etmeye de devam etmekteler.



Şekil 4.1 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 seyahat öneri sistemlerinin de seyahat endüstrisi üzerinde önemli bir etkisi olmuştur. Çevrimiçi seyahat acentelerinin ve inceleme sitelerinin yükselişiyle, gezginler her zamankinden daha fazla bilgi sahibi oluyor. Bu, her ne kadar müşterileri çekmek ve elde tutmak için yüksek kaliteli deneyimler sağlamak için oteller ve destinasyonlar üzerinde baskı oluştursa da müşterilerin deneyimlerini doğrudan artıran bir unsur olmuştur.

Makine öğrenmesi ve derin öğrenmede devam eden ilerlemelerle birlikte seyahat öneri sistemlerinin geleceği umut verici görünmektedir. Günümüzde doğal dil işlemeden resim ve videolar da dahil olmak üzere büyük verinin analiz edilmek için kullanıldığı tüm alanlarda makine öğrenmesi ve derin öğrenme algoritmaları oldukça başarılı öneriler oluşturmakta ve kullanıcılara iyi bir deneyim hazırlamaktadır.

Bu bölümde öneri sistemlerinin temelleri, türleri ve uygulamaları ile seyahat öneri sistemleri incelenecektir.

4.2 Öneri Sistemleri Süreçleri

Bu bölümde öneri sistemleri için ne tür verilerin hangi yollarla toplandığından, öğrenme ve tahminleme süreçlerinden bahsedilmektedir.

4.2.1 Verilerin Toplanması

Sistemlerin düzgün çalışabilmesi için probleme uygun verilerin toplanması ve profil/model yapılanmasının iyi oluşturulması gereklidir [56]. Makul bir öğrenme gerçekleştirebilmek için sistemin bildiği bilgilerin yeterli olması gerekmektedir, yani modelin mümkün olduğunca fazla bilgiyle beslenmesi gereklidir.

Öneri sistemleri kullanıcılardan aldıkları açık geri bildirimler ile kullanıcıların kapalı tercihlerinin gözlenmesi sonucunda oluşan veriler ile beslenir. Açıkça kullanıcıdan alınan geri bildirimlerin sisteme etkisi, kullanıcının verdiği geri bildirimin kalitesi ile ilişkilidir. Kullanıcıdan efor beklemesi ve kullanıcının istenen geri bildirimin hepsini açıkça vermeyebilecek olması bu geri bildirim sisteminin negatif kısmı olmasına rağmen kullanıcının üstü kapalı geri bildirimlerinden daha tutarlı bir sonuç ortaya koyar. Bu iki yöntemin negatifliklerini yok etmek için genelde bu iki yöntemin harmanlandığı hibrit yöntemler tercih edilmektedir.



Şekil 4.2 Öneri Sistemlerinde Müşteriden Toplanabilecek Örnek Veri Kümesi

4.2.2 **Öğrenme ve Tahminleme**

Öneri sistemleri söz konusu olduğunda, öğrenme ve tahmin, kullanıcılara sağlanan önerilerin kalitesini belirleyen temel bileşenlerdir. Öğrenme aşamasında, kişiselleştirilmiş önerilerde bulunmak için, bilgi toplama aşamasında sağladıkları örtülü ve açık geri bildirime dayalı olarak kullanıcının özelliklerini filtrelemek ve analiz etmek için bir öğrenme algoritması kullanılır. Bu, yeni toplanan bilgileri tahmin edebilen ve sistemin kullanıcının tercihlerine, davranışlarına ve ilgi alanlarına göre uyarlanmış öneriler sunmasına yardımcı olan bir model oluşturur.

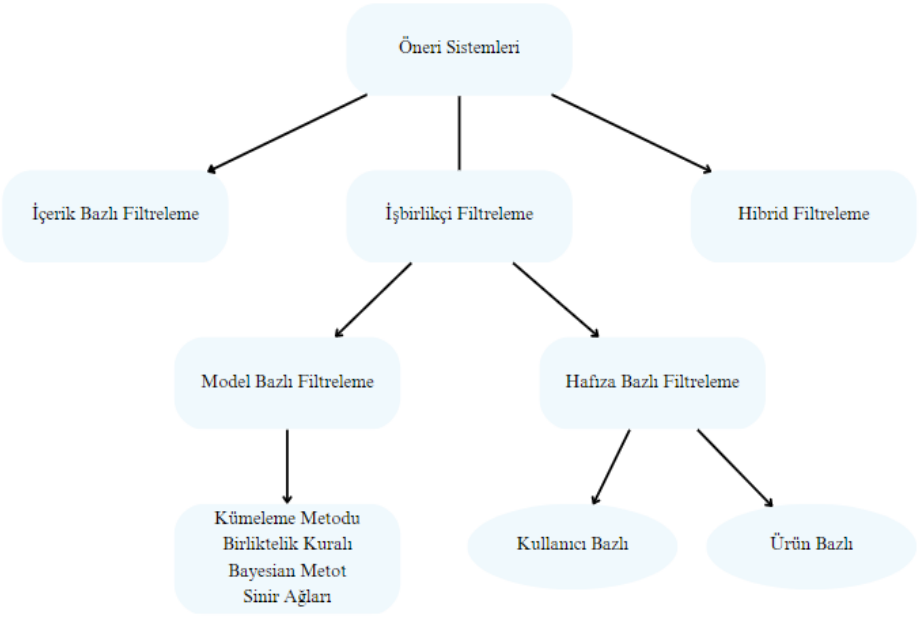
Ancak, öğrenme algoritması ve modelinin doğruluğu, bilgi toplama aşamasında toplanan verilerin kalitesine ve miktarına bağlıdır. Bu nedenle öneri sistemleri, öğrenme ve tahmin modellerinin doğruluğunu artıran ilgili verileri toplamak için genellikle işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit filtreleme gibi farklı teknikler kullanır.

4.3 Seyahat Öneri Sistemleri

İster iş ister eğlence için seyahat etmek hayatımızın vazgeçilmez bir parçası haline geldi. Seyahat planlaması yaparken dikkate alınabilecek birçok husus vardır: gidilecek yerler, tarih, uygunluk, fiyat vb. Bu kadar çok seçenek mevcutken, kişisel seyahat öneri sistemleri bir ihtiyaç haline gelmiştir. Bu sistemler, yolculara özel öneriler sunmak için makine öğrenmesi algoritmalarını kullanarak ulaşım, konum, otel ve etkinlikler hakkındaki verileri değerlendirmektedir.

4.4 Temel Yaklaşımlar

Öneri sistemlerinin birçoğunda, sistemin bulunduğu ortama kullanıcı veya ortam ile ilgili eklenebilecek verilerin sağlanmasının ardından filtreleme algoritmaları kullanmaktadır. 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. Bu yöntemlerin her birinin avantajları, dezavantajları, sınırlamaları vardır ve bu bölümde bu konulardan bahsedilecektir.



Şekil 4.3 Öneri Sistemlerinde Filtreleme Yöntemleri

Bunlara ek olarak, öneri sistemlerinde bir gruplama da öneri sistemleri nesli için yapılabilir. Öneri sistemleri üç nesilde incelenebilir.

İlk nesil Öneri Sistemleri Bilgi Tabanlı, İçerik Tabanlı, İşbirliğine Dayalı Filtreleme ve Hibrit Öneri Sistemleri tekniklerini kullanarak kişiselleştirilmiş öneriler kavramını tanıttı. Bu teknikler, her bir kullanıcının benzersiz tercihlerine hitap eden kişiselleştirilmiş önerilerin geliştirilmesini mümkün kıldığı için önemliydi. Örneğin, Netflix'in öneri motoru, kullanıcının izleme geçmişini, derecelendirmelerini ve farklı filmlerin veya TV şovlarının içeriği arasındaki benzerliği dikkate alan bir Hibrit Öneri Sistemi kullanır.

İkinci nesil Öneri Sistemleri, önerilerin doğruluğunu artıran yeni teknikler getirdi. Matris Çarpanlarına Ayırma, Web Kullanım Madenciliği ve Kişiliğe Dayalı Öneri Sistemleri bu nesilde kullanılan tekniklerden bazılarıdır.. Bu teknikler, karmaşık kullanıcı davranışını ve tercihlerini yakalayamama gibi birinci nesil tekniklerin bazı sınırlamalarının ele alınmasına yardımcı oldukları için önemliydi. Örneğin, Amazon'un öneri motoru, müşterilerine kişiselleştirilmiş ürün önerileri yapmak için işbirlikçi filtreleme ve matris çarpanlarına ayırmanın bir kombinasyonunu kullanır [5].

Üçüncü nesil Öneri Sistemleri, Yapay Zeka alanında Derin Öğrenmenin devrimi ile ortaya çıkmışlardır. Bu nesil, Kısıtlı Boltzmann Makineleri, Derin Öğrenmeyi Kullanan İşbirlikçi Filtreleme, Derin İçerik Tabanlı Öneri Sistemleri, CoNN Kullanarak İncelemeleri Kullanan Kullanıcıları ve Öğeleri Birleştiren Modelleme ve Ürün Tabanlı Öneri Sistemleri gibi yeni teknikleri tanıtmıştır. Bu teknikler, Önerici Sistemlerin kullanıcı verilerinden daha karmaşık ve soyut özellikler öğrenmesini sağlayarak daha doğru ve kişiselleştirilmiş önerilere yol açtığı için önemlidir. Örneğin, Spotify'ın öneri motoru, kullanıcılarına şarkı ve çalma listeleri önermek için kullanıcı davranışını, müzik zevkini ve bağlamı analiz etmek için Derin Öğrenme tekniklerinin bir kombinasyonunu kullanılır.

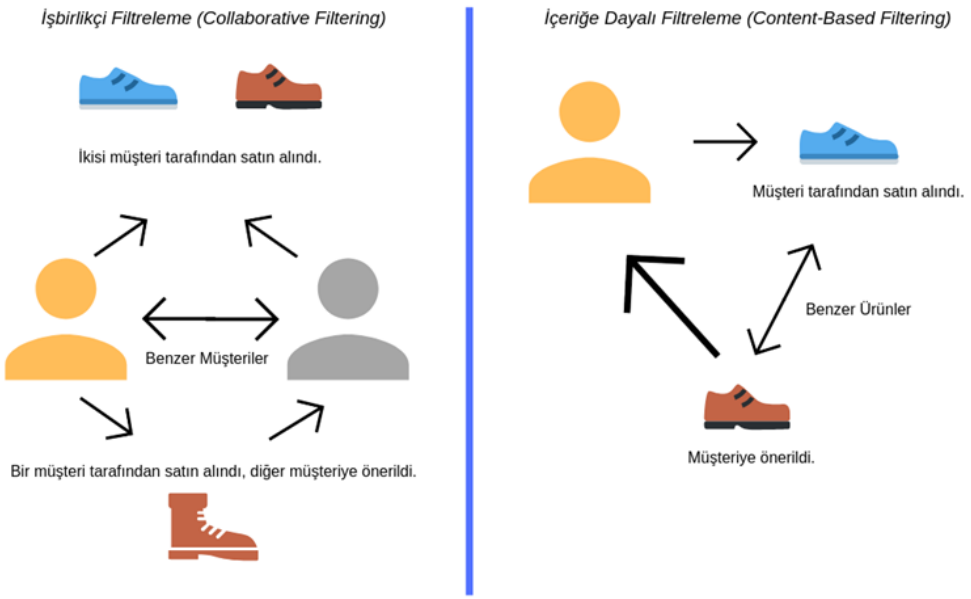
Sonuç olarak, Öneri Sistemlerinin her nesli, daha doğru ve kişiselleştirilmiş önerilere yol açan yeni teknikler ve ilerlemeler getirmiştir. Bu ilerlemeler, genel kullanıcı deneyimini ve dijital ürün ve hizmetlerle ilgili memnuniyeti iyileştirmede çok önemli olmuştur.

4.4.1 İçerik Tabanlı Öneri Sistemleri (Content-based Filtering)

Kullanıcının ön planda tutularak objelerin filtrelendiği öneri biçimidir. Bu sistemde önceden yapılan işlemler göz önünde bulundurularak nesnenin özelliklerine göre tahminleme yapılmaktadır. İnternet sayfaları, yayınlar, haberler gibi dokümanların önerilmesinde yaygın olarak kullanılan başarılı bir tekniktir.

Bu teknikte, en değerli içeriği bulmaya yarayan TFIDF (terim frekansına ters belge frekansı) gibi uzay-vektör modelleri kullanılabileceği gibi Naïve Bayes Sınıflandırıcısı, Karar Ağaçları veya Yapay Sinir Ağları gibi olasılıksal metotlar da kullanılabilir. Bu teknikte kullanıcıların sisteme herhangi bir etkisi olmazken, en büyük sınırlaması ise nesnelerin özellikleri hakkında derin bilgi birikimine ihtiyaç duymasıdır. Google, Wikipedia, IMDB gibi siteler arama motorlarında bu yöntemi kullanmaktadır.

Yeterli verinin olmadığı durumlarda modelin başlamasında meydana gelen soğuk başlangıç probleminin görülmemesi, kullanıcı bağımsız olması ve açıklanabilirliği içerik tabanlı öneri sistemlerinin en büyük avantajları iken, aşırı öğrenmeye sebep olabilmesi ve içerik özelliklerini özellik işlemeye tabi tutmadaki zorluklar bu modelin en büyük dezavantajlarıdır.



Şekil 4.4 İşbirlikçi ve İçerik Tabanlı Öneri Sistemleri

4.4.2 İşbirlikçi Öneri Sistemleri (Collaborating Filtering)

İşbirlikçi öneri sistemleri, benzer kullanıcı davranışlarının modellenerek öneri sunulmasını sağlayan sistemlerdir. Temelinde aynı davranışlara sahip insan gruplarınının eşleşmesi ile önerme vardır. Ortak zevk, ihtiyaç ve çıkarlara sahip kişilerin benzer istekleri olacağı varsayımına dayanmaktadır. Kullanıcı grubu - nesne ilişkisi temellidir. Bu sistemde üretilen çıktılar öneri olabileceği gibi tahminleme de verebilmektedir. Model tabanlı ve hafıza tabanlı olmak üzere iki başlık altında incelenebilirler.

- Hafıza Tabanlı Öneri Sistemi; önceden kullanıcı tarafından değerlendirilen nesnelerin, kullanıcıyla beğeniler benzeşen bir komşunun aranmasında kullanılmasını içeren öneri sistemidir.
- Model Tabanlı Öneri Sistemi; işbirlikçi öneri performansını arttırmak için önceki oylamaları kullanarak makine öğrenmesi modeli ile tahminleme yapılmasını içeren sistemdir. Modellemede makine öğrenmesi ve veri madenciliği yöntemlerini temel alır.

Amazon, Netflix gibi siteler e-ticaret kısımlarında ürünlerinin önerilmesinde bu sistemi kullanmaktadır.

Alan bilgisi (domain knowledge) gerektirmemesi, yeni ürün öneri kabiliyeti ile aşırı öğrenmeyi engelleyen sisteme sahip oluşu ve esnekliği, işbirlikçi öğrenmenin en güçlü yanları iken, soğuk başlangıç ve veri seyrekliği bu yöntemin dezavantajları arasında yer alır.

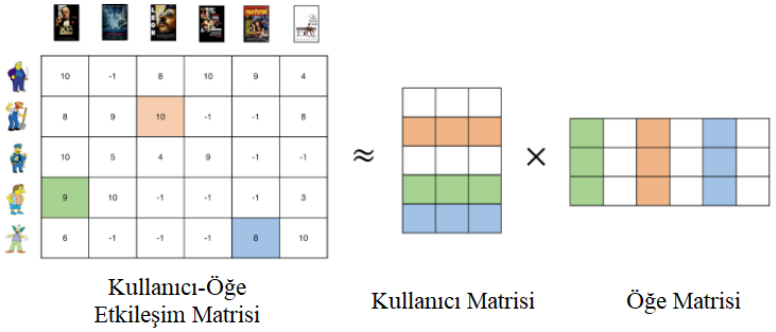
4.4.2.1 Matris Çarpanlara Ayırma (Matrix Factorization)

Matris çarpanlarına ayırma, öneri sistemlerinde kullanılan işbirlikçi bir filtreleme tekniğidir. Kullanıcı-öğe etkileşim matrisini, çarpanlarına ayırma yöntemleri matrisi kapsayan iki küçük matrise ayırmayı içerir. Öğe popülaritesine ve kullanıcı performansına dayalı olarak ağırlıkları düzenleyerek, ortaya çıkan tahminler iyileştirilebilir. Matris çarpanlara ayırmanın formülü aşağıdaki gibidir:

$$R_{ij} = P_i^T \, x \, Q_j \tag{4.1}$$

Burada

- R_{ij} öneri matrisini,
- P_i^T öğe matrisinin transpozesini, ve
- Q_j kullanıcı matrisini ifade etmektedir.



Şekil 4.5 Matris Çarpanlara Ayırma Denklemi

Matris çarpanlarına ayırma, kullanıcı-öğe etkileşimlerini düşük boyutlu bir gizli uzayda modelleyerek, geleneksel öneri yöntemlerindeki seyreklik ve ölçeklenebilirlik sorunlarına etkili bir çözüm getirmiştir.

Matris çarpanlarına ayırma model tabanlı ve bellek tabanlı olmak üzere iki kategoride incelenebilir. Buna ek olarak, matris çarpanlarına ayırma, öneri doğruluğunu artırmak için kullanıcı demografisi, öğe öznitelikleri ve zamansal dinamikler gibi bağlamsal bilgileri entegre edebilir. Bu teknik, bağlama duyarlı matris çarpanlara ayırma olarak bilinir ve çeşitli senaryolarda geleneksel matris çarpanlara ayırmaya göre üstün performans göstermiştir.

Öneri sistemlerinin yanı sıra, matris çarpanlarına ayırmanın görüntü ve ses işleme, doğal dil işleme ve sosyal ağ analizi gibi başka alanlarda da uygulamaları vardır. Temel bir makine öğrenmesi tekniği olarak matris çarpanlara ayırma, Derin Sinir Ağları Sinir Ağları gibi gelişmiş modellerin geliştirilmesine ilham vermiştir.

4.4.2.2 **Kümeleme Yöntemi (Clustering)**

Nesneleri benzerliklerine göre gruplandıran önerilerde bulunma yöntemidir. Kümeleme, basitçe, nesnelerin özelliklerini inceler ve benzerliklerine göre

kümelere koyar. Kullanıcının beğendiği ürünlerle aynı kümedeki ürünler daha sonra önerilerde bulunmak için kullanılır.

4.4.2.3 **Karar Ağaçları (Decision Trees)**

Ağaç benzeri bir paradigma kullanan bir dizi yargıya dayalı olarak önerilerde bulunmak için bir yöntemdir. Karar ağaçları, bütçe ve konum gibi bir dizi kullanıcı tercihine dayalı olarak kullanıcılara konum, otel veya etkinlik önermek için kullanılır.

4.4.2.4 **Birliktelik Kuralı (Association Rule Learning)**

Birliktelik kuralı öğrenimi, bir veri kümesindeki değişkenler arasındaki ilginç ilişkileri ve örüntüleri belirlemeyi amaçlayan bir denetimsiz makine öğrenmesi türüdür. Bu teknik, farklı veri öğeleri arasındaki bağımlılıkları kontrol etmeyi ve bunları daha karlı bir şekilde haritalamayı içerir.

Birliktelik kuralı öğrenmenin uygulamadaki bir örneği, perakendeciler tarafından hangi öğelerin sıklıkla birlikte satın alındığını belirlemek için yaygın olarak kullanılan pazar sepeti analizidir. Satıcı, müşteri satın alma verilerini analiz ederek ürünler arasındaki örüntüleri ve ilişkileri belirleyebilir ve ardından bu bilgileri ürün yerleştirmeyi ve promosyonları optimize etmek için kullanabilir.

Örneğin, bir müşteri ekmek satın alırsa, birliktelik kuralı öğrenme algoritması, müşterinin aynı zamanda tereyağı, yumurta veya süt satın alma olasılığının da yüksek olduğunu önerebilir. Bu bilgilere dayanarak perakendeci, bu ürünleri birlikte mağazaya yerleştirebilir veya müşteri dört ürünü de satın alırsa özel bir promosyon sunabilir.

4.4.2.5 **Yapay Sinir Ağları (Neural Networks)**

Yapay sinir ağlarında öneri sistemleri, insan beyninin işleyişini taklit eden önerilerde bulunmak için bir yöntemdir. Sinir ağları, çok sayıda bağlantılı düğüm katmanını kullanarak verileri değerlendirir ve verilerdeki örüntülere ve korelasyonlara dayalı öneriler oluşturur. Sinir ağları, büyük hacimli kullanıcı

verilerini değerlendirmek ve ardından kişiselleştirilmiş öneriler sağlamak için öneri sistemlerinde kullanılmaktadır.

4.4.3 Hibrit Öneri Sistemleri

Hibrit öneri sistemleri, diğer yöntemlerin dezavantajlarını yok ederek en iyi performansı almak için birlikte kullanılması sonucu oluşan öneri sistemleridir. Hibrit öneri sistemleri, içerik tabanlı filtreleme, işbirlikçi filtreleme ve matris çarpanlara ayırma gibi farklı öneri algoritmaları birleştirilerek oluşturulabilir. Hibrit bir yaklaşım kullanmanın temel amacı, her bir yöntemin güçlü yönlerinden yararlanmak ve zayıf yönlerini azaltmaktır. Sistem, birden çok algoritmayı birleştirerek daha geniş bir yelpazede öneriler sunabilir, kullanıcının tercihlerini karşılayabilir ve genel kullanıcı deneyimini iyileştirebilir.

Hibrit öneri sistemleri, kişiselleştirilmiş önerilerin kullanıcı deneyimini geliştirmede çok önemli bir rol oynadığı e-ticaret, sosyal medya ve diğer sektörlerde yaygın olarak kullanılmaktadır. Spotify tarafından haftalık müzik önerisinde bu sistem kullanılmaktadır.

4.4.3.1 Çarpanlara Ayırma Makineleri (Factorization Machines)

Çarpanlara Ayırma Makineleri, matris çarpanlarına ayırma ve regresyon modellerini birleştiren önerilerde bulunmak için hibrit bir yaklaşımdır. Çarpanlara ayırma makineleri, kullanıcı tercihlerini ve ürün niteliklerini hesaba katan kullanıcı-ürün etkileşimlerine dayalı olarak öğe önermek için kullanılır. Çarpanlara Ayırma Makineleri, öneri sistemleri oluşturmak için güçlü bir araçtır. Karmaşık özellik etkileşimlerini modellemede etkilidirler ve oldukça doğru öneriler sağlayabilirler.

4.4.4 Takviyeli Öğrenme (Reinforcement Learning)

Takviyeli öğrenme, kullanıcı davranışından öğrenmek ve deneme yanılmaya dayanan önerilerde bulunmak için bir yöntemdir. Takviyeli öğrenme, kullanıcıların sistemle zaman içinde nasıl etkileşime girdiğine dayalı olarak öğe önermek için kullanılan bir tekniktir.

4.5 Geniş ve Derin Öğrenme Sistemleri (Wide and Deep Recommendation Systems)

Öneri sistemleri, e-ticaret, medya ve sosyal ağlar dahil olmak üzere çeşitli sektörlerde yaygın olarak kullanılmaktadır. Doğru ve alakalı öneriler sağlamak için işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımlar gibi çeşitli algoritmalara ve tekniklere güvenirler.

4.5.1 Geniş ve Derin Sinir Ağı

Google tarafından öneri sistemleri için geliştirilen bir makine öğrenme algoritmasıdır. Geniş ve Derin Sinir Ağı [31], 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 öğe özelliklerinin ana etkilerini yakalarken derin model, bunlar arasındaki karmaşık etkileşimleri yakalar.

4.5.2 DLRM Algoritması

Derin Öğrenme Öneri Modeli (DLRM) [32], Facebook tarafından öneri görevleri için oluşturulan bir makine öğrenmesi algoritmasıdır. Öneri puanları üretmek için kullanıcı ve öğe 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 öğe gömmeleri arasındaki etkileşimleri öğrenerek yüksek doğruluk elde eder.

4.6 Sonuç

Öneri sistemleri, e-ticaret, medya ve sosyal ağlar dahil olmak üzere çeşitli sektörlerde yaygın olarak kullanılmaktadır. Doğru ve alakalı öneriler sağlamak için işbirlikçi filtreleme, içerik tabanlı filtreleme ve hibrit yaklaşımlar gibi çeşitli algoritmalara ve tekniklere güvenirler.

Facebook tarafından geliştirilen DLRM algoritması ve Google tarafından geliştirilen geniş ve derin sinir ağı, öneri görevleri için kullanılan iki popüler makine öğrenme algoritmasıdır. 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.

Bu bölümde öneri sistemlerinin temelleri, türleri ve uygulamaları incelenmiştir.

5
DERİN ÖĞRENME TABANLI SEYAHAT ÖNERİ SİSTEMİ GELİŞTİRME

5.1 Giriş: Önerilen Yöntem

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.

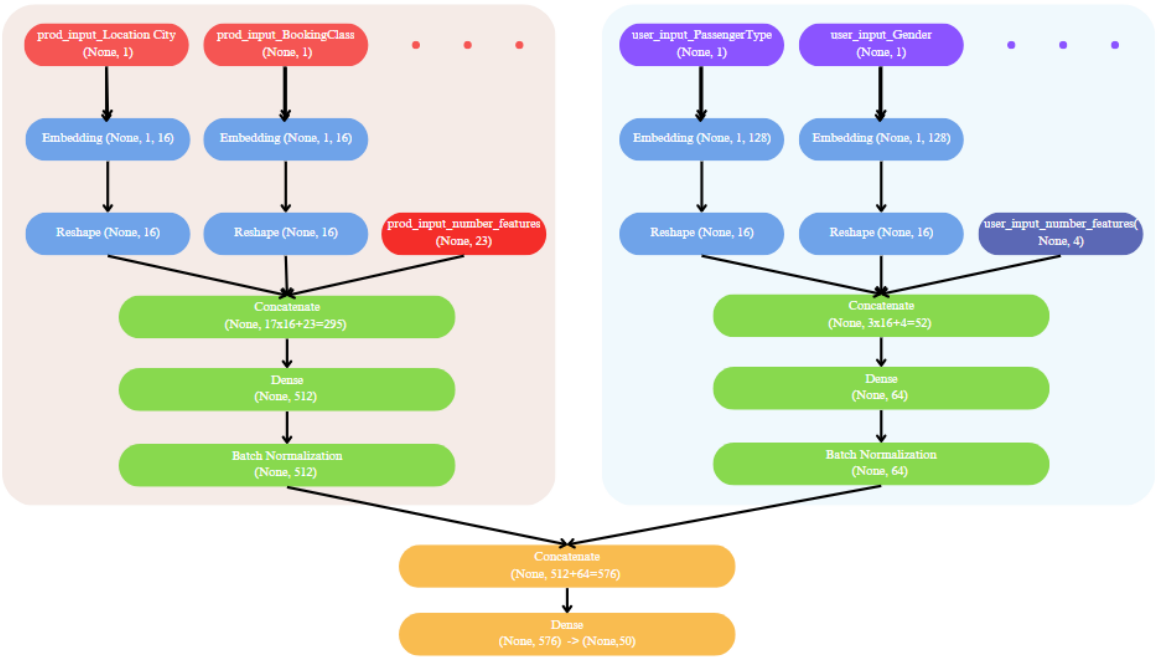
Çalışma kapsamında oluşturulan model, kullanıcı modeli, ürün modeli ve iki modeli birleştiren birleştirme katmanı olmak üzere üç ana bölümden oluşur. Kullanıcı ve ürün modellerinin her biri, yoğun (dense) vektörleri sabit boyuta dönüştüren yerleştirmelerden (embedding) ve tamamen bağlantılı (fully connected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

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 embeding_size parametresi, her kategori özelliği 16 boyuttan oluşan bir vektörle temsil edilecek şekilde ayarlanır.



Şekil 5.1 Derin ve Geniş Seyahat Öneri Sistemi Mimarisi

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 tur 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.

5.2 Veri Kümesi

Derin öğrenme modelleri, sistemlerin bireysel müşteriler için doğru tahminler yapmasını sağlayarak, büyük veri kümelerinden karmaşık örüntüleri öğrenme yeteneğine sahiptir. Bu nedenle bu çalışma kapsamında, 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 [57-60] çalışmalarında da kullanılmıştır. [57]'te Spark büyük veri platformu üzerinde koşturulan işbirlikçi algoritmalar kullanılarak bir uçuş öneri sistemi geliştirilmiştir. [58]'te uçuş için satılan ek hizmetler değerlendirilerek ek hizmetler öneri sistemi geliştirilmiştir. [59, 60]'da acentelerin sattıkları uçak biletleri ve davranışları incelenerek müşteri kayıp analizi üzerine çalışmalar yapılmıştır.

Bu çalışma kapsamında, 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 5.1 Çalışmada Kullanılan Veri Kümesindeki Kolonların Açıklamaları

Özellik	**Açıklama**	**Özellik**	**Açıklama**
PassengerId	*Yolcu ID*	SeatCount	*Biletlenen koltuk sayısı*
PassengerType	*Yolcu Tipi*	MarketingAirline	*Satan havayolu*

Type	*PassengerType sayısal değeri*	OperatingAirline	*Uçan havayolu*
Gender	*Cinsiyet*	FareBasis	*Sınıf ve ücret tipinin kodu*
BirthDate	*Doğum Tarihi*	FareType	*Ücret tipi kodu*
WheelChairServiceType	*Tekerlekli sandalye isteniyor mu?*	OD_Origin	*Başlangıç havalimanı (kısaltma)*
AgencyId	*Satışı yapan Acente ID*	OD_Destination	*Varış havalimanı (kısaltma)*
BookingCode	*PNR*	IsRefund	*Geri iade edilmiş mi etiketi*
VatFlag	*Yurt içi, Yurt dışı*	LoyaltyProgram	*Sadakat programına dahil mi?*
SellingDate	*Satış tarihi*	TicketTotal	*Toplam satış tutarı*
Flight_Baggage	*Bagaj boyutu*	PaymentType	*Ödeme türü*
TripType	*Yolculuk tipi*	IsScheduleChange	*Uçuş saati değişti mi etiketi*
DepartureDay	*Kalkış günü*	IsExpirationChange	*Opsiyon değişikliği etiketi*
DepartureTime	*Kalkış saati*	Location_City	*Acentenin bulunduğu şehir*
ArrivalDay	*Varış günü*	Location_Country	*Acentenin bulunduğu ülke*
ArrivalTime	*Varış saati*	Status_IfActive	*Aktif mi*
Origin	*Başlangıç havalimanı*	T_CreationDate	*Acente kayıt oluşturulma tarihi*
Destination	*Varış havalimanı*	AgencyType	*Acente türü*
BookingClass	*Uçuş sınıfı*	DefaultCurrency	*Varsayılan para birimi*

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 5.1’de tüm öznitelikler ve açıklamaları sunulmaktadır.

Ham veri kümesi kolonlarının veri türleri Tablo 5.2'de, boş değer sayıları (#null), benzersiz veri sayıları ve nümerik kolonlar için yüzdelik değerleri Tablo 5.3'te verilmiştir.

Tablo 5.2 Çalışmada Kullanılan Veri Kümesindeki Kolonların Veri Tipleri

Özellik	**Veri Tipi**	**Özellik**	**Veri Tipi**
AgencyId	*int32*	Location_Country	*category*
AgencyType	*category*	LoyaltyProgram	*int8*
ArrivalDay	*object*	MarketingAirline	*object*
ArrivalTime	*object*	OD_Destination	*object*
BirthDate	*int16*	OD_Origin	*object*
BookingClass	*category*	OperatingAirline	*object*
BookingCode	*object*	Origin	*object*
DefaultCurrency	*category*	PassengerId	*int32*
DepartureDay	*object*	PassengerType	*category*
DepartureTime	*object*	PaymentType	*category*
Destination	*object*	SeatCount	*int8*
FareBasis	*object*	SellingDate	*object*
FareType	*category*	Status_IfActive	*float64*
Flight_Baggage	*category*	T_CreationDate	*object*
Gender	*category*	TicketTotal	*float16*
IsExpirationChange	*int8*	TripType	*category*
IsRefund	*int8*	Type	*int8*
IsScheduleChange	*int8*	VatFlag	*category*
Location_City	*object*	WheelChairServiceType	*int8*

Tablo 5.3 Çalışmada Kullanılan Veri Kümesindeki Kolonların Boş Değer ve Benzersiz Değer Sayıları

Özellik	#NULL	#Unique	Özellik	#NULL	#Unique
AgencyId	*0*	*3342*	Location_Country	*2*	*31*
AgencyType	*13406*	*5*	LoyaltyProgram	*0*	*2*
ArrivalDay	*0*	*2406*	MarketingAirline	*0*	*321*
ArrivalTime	*0*	*1082*	OD_Destination	*0*	*1059*
BirthDate	*0*	*251*	OD_Origin	*0*	*1012*
BookingClass	*94*	*108*	OperatingAirline	*0*	*393*
BookingCode	*11*	*2359497*	Origin	*0*	*1014*
DefaultCurrency	*2*	*4*	PassengerId	*0*	*2175911*
DepartureDay	*0*	*2405*	PassengerType	*0*	*5*
DepartureTime	*0*	*780*	PaymentType	*5515*	*8*
Destination	*0*	*1060*	SeatCount	*0*	*12*
FareBasis	*151760*	*21172*	SellingDate	*0*	*2311087*
FareType	*3855213*	*30*	Status_IfActive	*2*	*2*
Flight_Baggage	*41193*	*161*	T_CreationDate	*2*	*2699*
Gender	*0*	*2*	TicketTotal	*0*	*7861*
IsExpirationChange	*0*	*2*	TripType	*0*	*3*
IsRefund	*0*	*2*	Type	*0*	*5*
IsScheduleChange	*0*	*2*	VatFlag	*0*	*2*
Location_City	*2*	*294*	WheelChairServiceType	*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 5.2'de görülmektedir ve bu kolonlara tip dönüşüm işlemleri, veri dönüşüm kısmında yapılmıştır.

Tablo 5.4 Çalışmada Kullanılan Veri Kümesindeki Nümerik Kolonların Yüzdelik Değerleri

Özellik	**0**	**0.01**	**0.05**	**0.5**	**0.95**	**0.99**	**1**
AgencyId	*10088*	*1545013*	*1545960*	*1547653*	*1553033*	*1553825*	*1554048*
BirthDate	*1*	*1944*	*1956*	*1986*	*2007*	*2015*	*2017*
IsExpirationChange	*0*	*0*	*0*	*0*	*0*	*0*	*1*
IsRefund	*0*	*0*	*0*	*0*	*0*	*1*	*1*
IsScheduleChange	*0*	*0*	*0*	*0*	*0*	*1*	*1*
LoyaltyProgram	*0*	*0*	*0*	*0*	*0*	*0*	*1*
PassengerId	*1*	*56638.3*	*271454*	*2591078*	*4956012*	*5164284*	*5218779*
SeatCount	*0*	*1*	*1*	*1*	*4*	*6*	*11*
Status_IfActive	*0*	*0*	*0*	*1*	*1*	*1*	*1*
TicketTotal	*0*	*32*	*65*	*168*	*864*	*2244*	*29408*
Type	*1*	*1*	*1*	*1*	*4*	*5*	*5*
WheelChairService Type	*0*	*0*	*0*	*0*	*0*	*0*	*3*

Tablo 5.4'te görüldüğü üzere bazı kolonlar düşük kardinaliteye sahip olmasına rağmen nümerik veri tipinde bulunmaktadır ve bu veri tiplerinin dönüşümleri veri dönüşüm aşamasında yapılmıştır.

5.3 Veri Hazırlama

Veri hazırlama kısmında, derin bir öneri sistemi oluşturmak için gerekli ön işleme adımları uygulanmıştır.

5.3.1 Kullanılan Kütüphaneler

Çalışma aşağıdaki listelenen gerekli kitaplıkları içe aktararak başlar:

NumPy: NumPy, çok büyük, çok boyutlu diziler ve matrislerle uğraşmanıza izin veren bir Python bilimsel bilgi işlem araç setidir. Sayısal hesaplamalar için temel bir Python modülüdür ve veri bilimi, makine öğrenmesi ve bilimsel araştırmalarda yaygın olarak kullanılır.

Pandas: Pandas, bir Python veri işleme ve analiz araç takımıdır. Devasa veri kümelerini etkili bir şekilde depolamak ve işlemek için veri yapılarının yanı sıra veri temizleme, birleştirme, yeniden şekillendirme ve filtreleme araçları içerir. Pandas, veri bilimi ve makine öğrenmesi uygulamalarında yaygın olarak kullanılan bir programlama dilidir.

Matplotlib: Matplotlib, statik, animasyonlu ve etkileşimli görüntüler oluşturmak için kullanılabilen bir Python veri görselleştirme paketidir. Çok çeşitli görselleştirme işleri için son derece esnek ve uygundur.

Seaborn: Seaborn, Matplotlib üzerinde geliştirilmiş bir Python veri görselleştirme araç takımıdır. Yararlı ve görsel olarak çekici istatistik görselleri üretmek için üst düzey bir arayüz sunar. Seaborn, keşif amaçlı veri analizinin yanı sıra büyük veri kümelerinin görselleştirmelerini üretmek için sıklıkla kullanılır.

Datetime: Datetime, tarihler ve saatlerle etkileşime girmenizi sağlayan bir Python paketidir. Tarihleri, saatleri ve zaman aralıklarını temsil eden sınıfların yanı sıra tarih ve saatleri biçimlendiren ve ayrıştıran yöntemler içerir. Zaman serisi verileriyle çalışmak için, veri analizinde ve makine öğrenmesi uygulamalarında genellikle tarih saat kullanılır.

Sklearn (scikit-learn): Çeşitli yöntemler kullanarak sınıflandırma, regresyon, kümeleme ve boyut azaltma gibi görevler için araçlar sağlayan, NumPy ve SciPy kütüphanelerinden üretilen, popüler bir Python makine öğrenmesi araç setidir. Makine öğrenmesi algoritmaları oluşturmak için basit arabirimler sunar.

Tensorflow: Google tarafından tasarlanmış, açık kaynaklı bir derin öğrenme çerçevesidir. TensorFlow, derin öğrenme araştırmaları ve uygulamaları için değerli bir araçtır çünkü kullanıcıların hem CPU'larda hem de GPU'larda model eğitilmesine olanak tanır.

Keras: TensorFlow üzerinde çalışan üst düzey bir sinir ağı API'dır. Derin sinir ağları oluşturmak ve eğitmek için kullanımı kolay bir arayüz sağlar ve kullanıcıların çeşitli mimarileri hızlı bir şekilde prototiplemesine olanak tanır.

5.3.2 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 5.5'te verilen dağılıma sahip bir veri kümesi oluşturulmuştur. Tablo 5.6'da özelliklerin veri tipleri ve Tablo 5.7'de benzersiz değer sayıları verilmiştir.

Tablo 5.5 Çalışmada Kullanılan Veri Kümesinin Özellik-Veri Tipi Dağılımı

Veri Tipi	**Adet**	**Özellikler**
ID	*3*	*PassengerId, AgencyId, BookingCode*
Nümerik	*8*	*CustomerAge, SellingDateDayofyear, SellingDateWeekofyear, ArrivalDateDayofyear …*
Düşük Kardinalite, Kategorik	*12*	*TicketTotal, Type, PassengerType, Gender, VatFlag, TripType, PaymentType, DefaultCurrency, …*

Yüksek Kardinalite, Kategorik	*11*	*Origin, Destination, BookingClass, MarketingAirline, OperatingAirline, FareBasis, Location_City ...*
Düşük Kardinalite, Kategorik Nümerik	*10*	*WheelChairServiceType, IsRefund, LoyaltyProgram, IsScheduleChange, IsExpirationChange, ...*
Yüksek Kardinalite, Kategorik Nümerik	*6*	*SeatCount, Flight_Baggage_Dep, Flight_Baggage_Arr, SellingDateMonth, ...*

Tablo 5.6 Veri Kümesindeki Kolonların Ön İşleme Sonrası Veri Tipleri

	Veri Tipi		**Veri Tipi**
AgencyId	int32	Location_City	object
AgencyType	category	Location_Country	category
ArrivalDayofyear	int32	LoyaltyProgram	int8
ArrivalHourDayParts	object	MarketingAirline	object
ArrivalMonth	int32	OD_Destination	object
ArrivalWeekofyear	int32	OD_Origin	object
ArrivalYear	int32	OperatingAirline	object
BookingClass	category	Origin	object
BookingCode	object	PassengerId	int32
CustomerAge	int64	PassengerType	category
DefaultCurrency	category	PaymentType	category
DepartureDayofyear	int32	SeatCount	int8
DepartureHourDayParts	object	SellingDayofyear	int32
DepartureMonth	int32	SellingHourDayParts	object
DepartureWeekofyear	int32	SellingMonth	int32
DepartureYear	int32	SellingWeekofyear	int32
DepartureArrivalDuration	float64	SellingYear	int32
Destination	object	SellingToDepartureDuration	float64
FareBasis	object	Status_IfActive	float64
Flight_Baggage_Arr	int32	T_CreationDate	object

Flight_Baggage_Dep	int32	TicketTotal	category
Gender	category	TripType	category
IsExpirationChange	int8	Type	category
IsRefund	int8	VatFlag	category
IsScheduleChange	int8	WheelChairServiceType	int32

Tablo 5.7 Veri Kümesindeki Kolonların Ön İşleme Sonrası Benzersiz Değer Sayıları

	#Unique		**#Unique**
AgencyId	3339	Location_City	294
AgencyType	5	Location_Country	31
ArrivalDayofyear	366	LoyaltyProgram	2
ArrivalHourDayParts	4	MarketingAirline	314
ArrivalMonth	12	OD_Destination	1055
ArrivalWeekofyear	53	OD_Origin	1006
ArrivalYear	8	OperatingAirline	388
BookingClass	107	Origin	982
BookingCode	2355592	PassengerId	2174194
CustomerAge	97	PassengerType	5
DefaultCurrency	4	PaymentType	3
DepartureDayofyear	366	SeatCount	12
DepartureHourDayParts	4	SellingDayofyear	366
DepartureMonth	12	SellingHourDayParts	4
DepartureWeekofyear	53	SellingMonth	12
DepartureYear	8	SellingWeekofyear	53
DeparturetoArrivalDuration	3	SellingYear	6
Destination	884	SellingToDepartureDuration	172243
FareBasis	20689	Status_IfActive	2

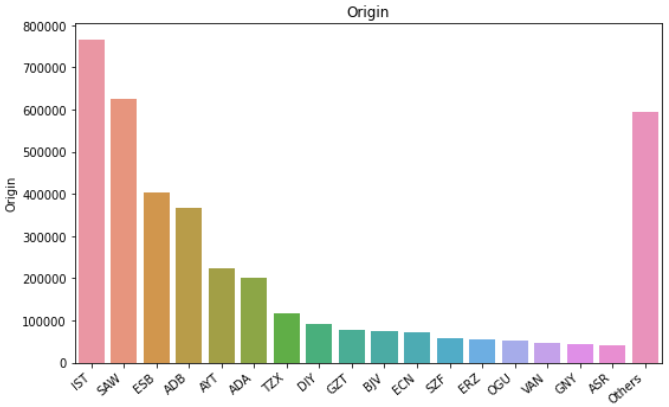
Flight_Baggage_Arr	19	T_CreationDate	2696
Flight_Baggage_Dep	21	TicketTotal	7843
Gender	2	TripType	2
IsExpirationChange	2	Type	5
IsRefund	2	VatFlag	2
IsScheduleChange	2	WheelChairServiceType	2

5.3.3 Veri Görselleştirme

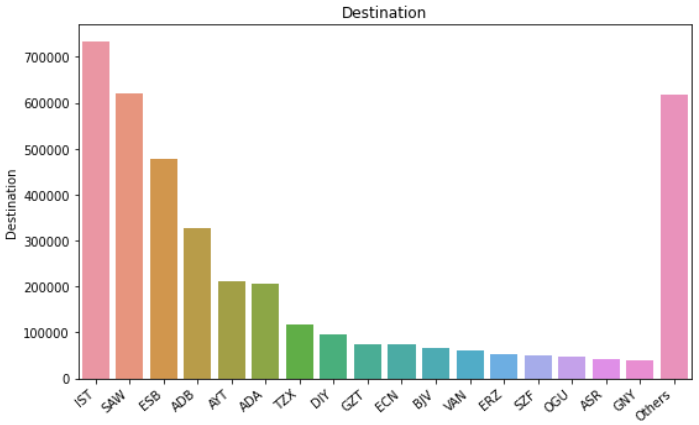
Veri görselleştirme, büyük ve karmaşık veri kümelerinden değerli içgörüler ve modeller çıkarmamızı sağladığı için modern veri analizinin çok önemli bir yönüdür. Görselleştirme, verilerin görsel bir temsilini sağlayarak veri ilişkilerini anlamayı ve analiz etmeyi, eğilimleri belirlemeyi ve diğer analiz biçimleriyle belirgin olmayabilecek önemli örüntüleri vurgulamayı kolaylaştırır.

Ayrıca, veri görselleştirme, görsel bir ekran olmadan yorumlanması veya anlaşılması zor olabilecek karmaşık verilerle çalışırken özellikle yararlıdır. Verilerin altında yatan yapının daha derinlemesine anlaşılmasını sağlar ve değişkenler arasındaki ilişkiye ilişkin içgörüler sağlayabilir

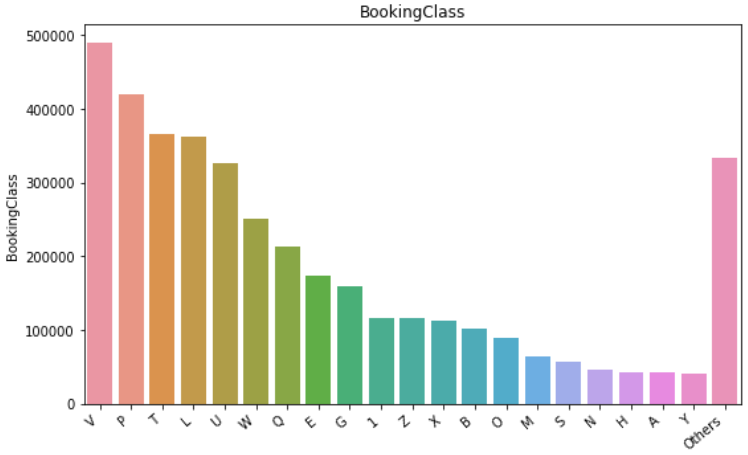
Bu bölümde modelde kullanılan önemli verilerin dağılımlarına ait görselleştirmelere yer verilmiştir.



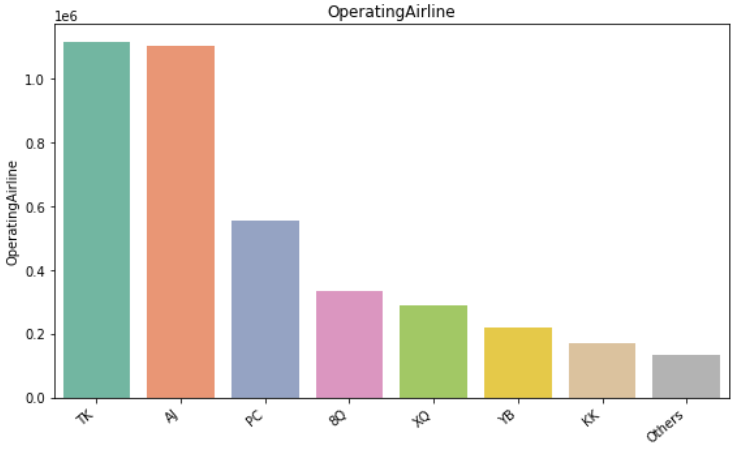
Şekil 5.2 Origin Kolonunun Dağılımı



Şekil 5.3 Destination Kolonunun Dağılımı



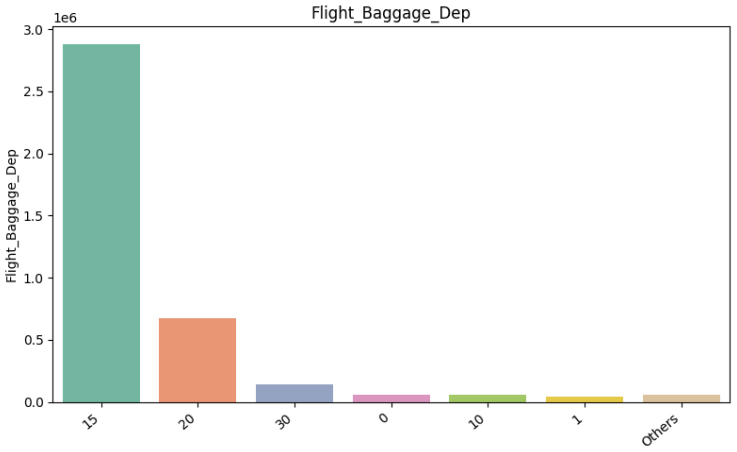
Şekil 5.4 Booking Kolonunun Dağılımı



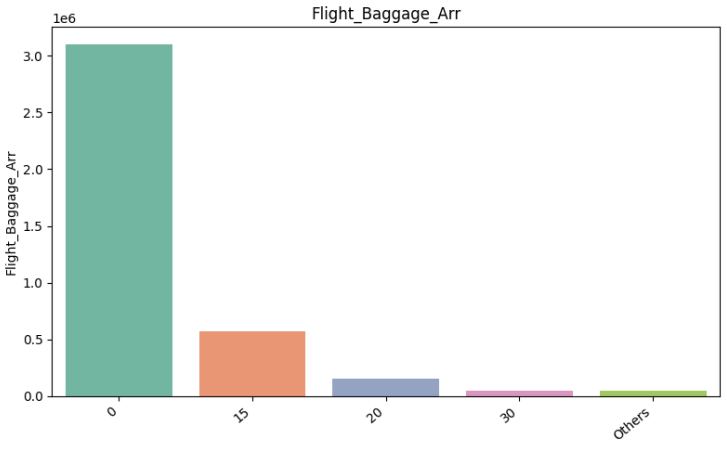
Şekil 5.5 OperatingAirline Kolonunun Dağılımı



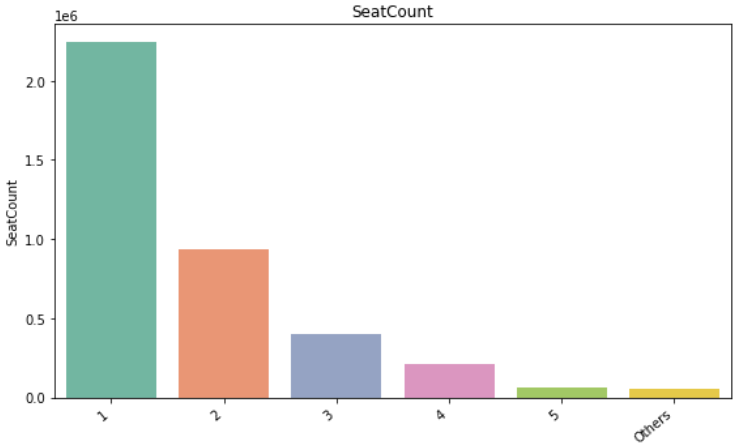
Şekil 5.6 Location_City Kolonunun Dağılımı



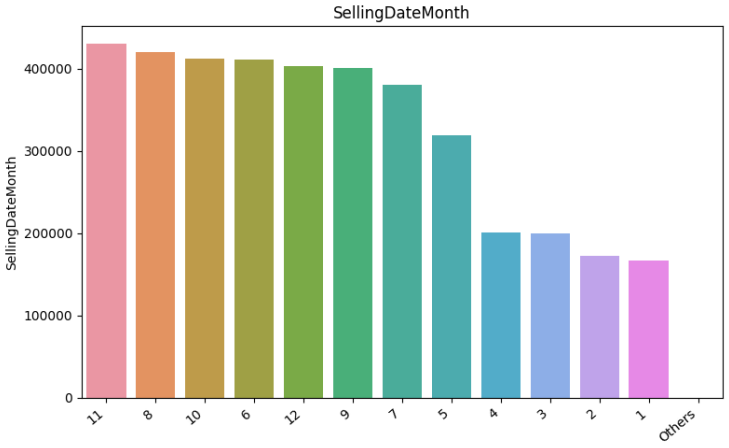
Şekil 5.7 Flight_Baggage_Dep Kolonunun Dağılımı



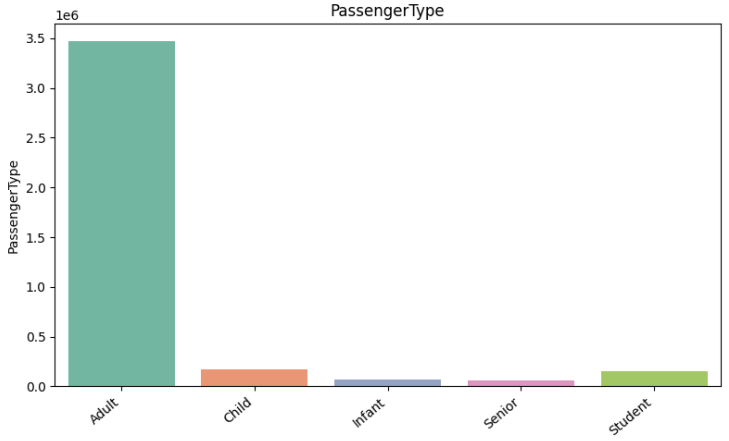
Şekil 5.8 Flight_Baggage_Arr Kolonunun Dağılımı



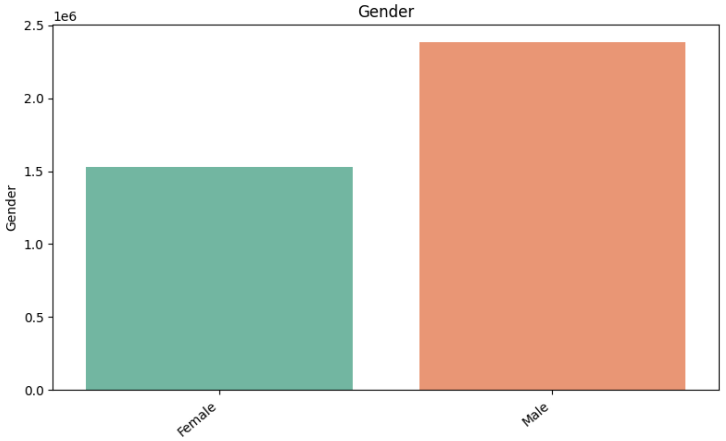
Şekil 5.9 SeatCount Kolonunun Dağılımı



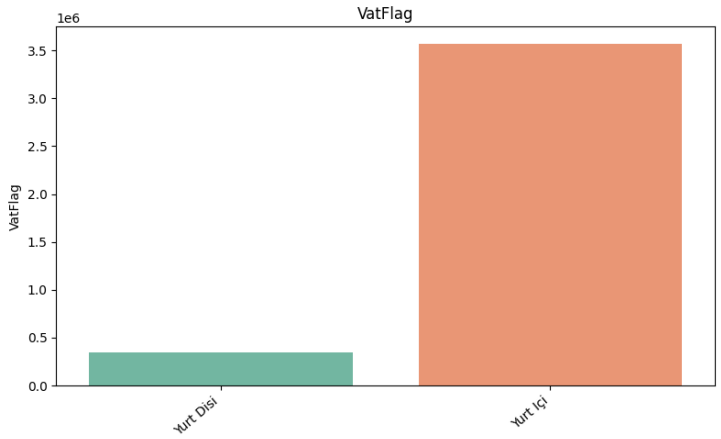
Şekil 5.10 SellingMonth Kolonunun Dağılımı



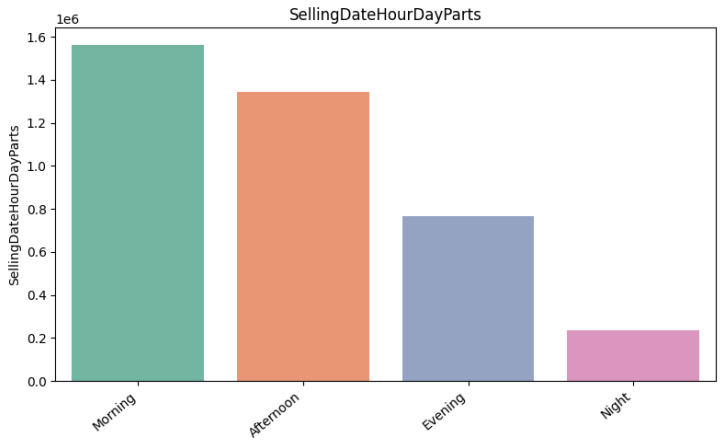
Şekil 5.11 PassengerType Kolonunun Dağılımı



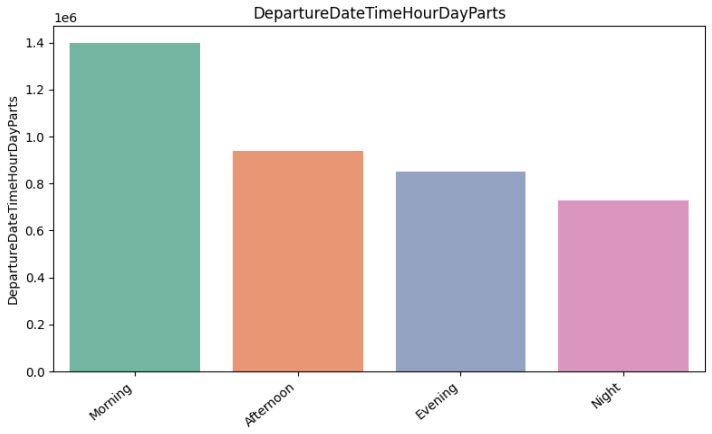
Şekil 5.12 Gender Kolonunun Dağılımı



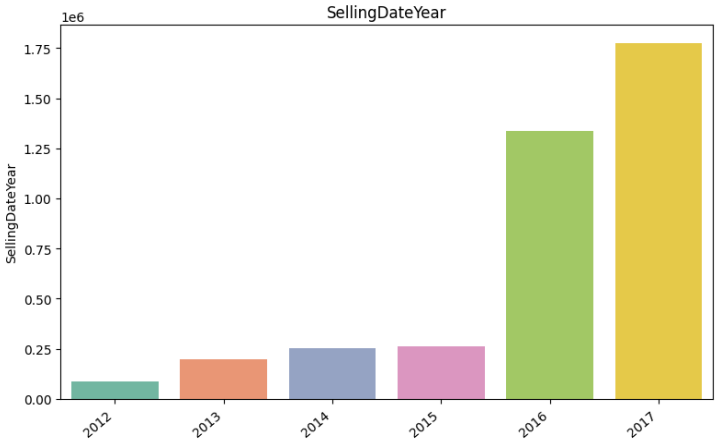
Şekil 5.13 VatFlag Kolonunun Dağılımı



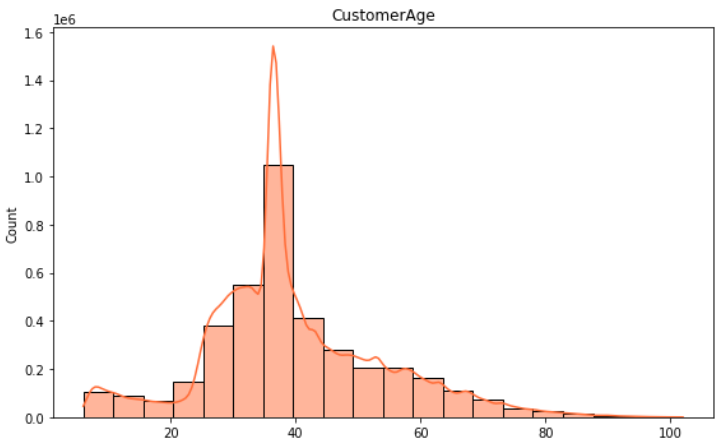
Şekil 5.14 SellingDayParts Kolonunun Dağılımı



Şekil 5.15 DepartureHourDayParts Kolonunun Dağılımı



Şekil 5.16 SellingYear Kolonunun Dağılımı



Şekil 5.17 CustomerAge Kolonunun Dağılımı

5.4 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 yanlılık 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 5.8'deki gibi ayrıldı.

Tablo 5.8 Kullanıcı-Öğe, Nümerik-Kategorik Özellik Dağılımı

	Özellikler
Kategorik Öğe Özellikleri	Origin, BookingClass, MarketingAirline, OperatingAirline, Location_City, Location_Country, VatFlag, TripType, PaymentType, AgencyType, DefaultCurrency, SellingDateHourDayParts, ArrivalDateTimeHourDayParts, DepartureDateTimeHourDayParts
Kategorik Kullanıcı Özellikleri	PassengerType, Gender

Nümerik Öğe Özellikleri	SellingDateDayofyear, SellingDateWeekofyear, ArrivalDateTimeDayofyear, ArrivalDateTimeWeekofyear, DepartureDateTimeDayofyear, DepartureDateTimeWeekofyear, SellingToDepartureDuration, SellingDateYear, ArrivalDateTimeYear, DepartureDateTimeYear ,DepartureToArrivalDuration, SeatCount, Flight_Baggage_Dep, Flight_Baggage_Arr, SellingDateMonth, ArrivalDateTimeMonth, DepartureDateTimeMonth, AgencyId
Nümerik Kullanıcı Özellikleri	TicketTotal, CustomerAge, LoyaltyProgram, Affinity

5.5 Model

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 ağırlıklarını 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 katmanıdır 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 çağırmayı 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 accuracy (top k accuracy) ile değerlendirilmiştir.

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.

5.6 Sonuçlar ve Öneriler

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 5.9'daki sonuçlar elde edilmiştir.

Tablo 5.9 Model Sonuçları

	kategorik çapraz entropi	En iyi 5 Doğruluk	En iyi 10 Doğruluk	Doğruluk	F1
Eğitim (onuncu devir)	1.1129	0.8833	0.9541	0.6173	0.6456
Test	1.0913	0.8871	0.9561	0.6250	0.6486

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 çalışma kapsamında 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.

6
SONUÇ

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

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

Bu çalışma kapsamında, 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 ogrenme tabanli bir seyahat öneri sistemi geliştirilmiştir. Bu yöntem ile kullanıcı tercihleri ve davranışları daha incelikli bir şekilde anlaşılması sağlanarak daha doğru ve kişiselleştirilmiş öneriler oluşturulmuştur.

Bu çalışma kapsamında, gerçek bir seyahat acentesi verisi kullanılmış ve çalışma sonunda 50 hedef noktasından en iyi 5'i, eğitim verisinde %88.33, test verisinde %88.71 doğrulukla tahminlenmiştir.

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.

KAYNAKÇA

[1] W. Fan vd., "A Comprehensive Survey on Trustworthy Recommender Systems". arXiv, 2022. doi: 10.48550/ARXIV.2209.10117. Available: https://arxiv.org/abs/2209.10117

[2] F. Karimova, "A Survey of e-Commerce Recommender Systems", European Scientific Journal, ESJ, c. 12, sy 34. European Scientific Institute, ESI, s. 75, Ara. 31, 2016. doi: 10.19044/esj.2016.v12n34p75. Available: http://dx.doi.org/10.19044/esj.2016.v12n34p75

[3] T. Belluf, L. Xavier, ve R. Giglio, "Case study on the business value impact of personalized recommendations on a large online retailer", Proceedings of the sixth ACM conference on Recommender systems. ACM, Eyl. 09, 2012. doi: 10.1145/2365952.2366014. Available: http://dx.doi.org/10.1145/2365952.2366014

[4] X. Amatriain ve J. Basilico, "Past, Present, and Future of Recommender Systems", Proceedings of the 10th ACM Conference on Recommender Systems. ACM, Eyl. 07, 2016. doi: 10.1145/2959100.2959144. Available: http://dx.doi.org/10.1145/2959100.2959144

[5] Y. Koren, "Factorization meets the neighborhood", Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, Ağu. 24, 2008. doi: 10.1145/1401890.1401944. Available: http://dx.doi.org/10.1145/1401890.1401944

[6] S. Rendle, "Factorization Machines", 2010 IEEE International Conference on Data Mining. IEEE, Ara. 2010. doi: 10.1109/icdm.2010.127. Available: http://dx.doi.org/10.1109/icdm.2010.127

[7] I. Paparrizos, B. B. Cambazoglu, ve A. Gionis, "Machine learned job recommendation", Proceedings of the fifth ACM conference on Recommender systems. ACM, Eki. 23, 2011. doi: 10.1145/2043932.2043994. Available: http://dx.doi.org/10.1145/2043932.2043994

[8] S. T. Al-Otaibi, "A survey of job recommender systems", International Journal of the Physical Sciences, c. 7, sy 29. Academic Journals, Tem. 26, 2012. doi: 10.5897/ijps12.482. Available: http://dx.doi.org/10.5897/ijps12.482

[9] A. Albadvi ve M. Shahbazi, "A hybrid recommendation technique based on product category attributes", Expert Systems with Applications, c. 36, sy 9. Elsevier BV, ss. 11480-11488, Kas. 2009. doi: 10.1016/j.eswa.2009.03.046. Available: http://dx.doi.org/10.1016/j.eswa.2009.03.046

[10] M. Balabanović ve Y. Shoham, "Fab", Communications of the ACM, c. 40, sy 3. Association for Computing Machinery (ACM), ss. 66-72, Mar. 1997. doi: 10.1145/245108.245124. Available: http://dx.doi.org/10.1145/245108.245124

[11] L. Pizzato, T. Rej, T. Chung, I. Koprinska, ve J. Kay, "RECON", Proceedings of the fourth ACM conference on Recommender systems. ACM, Eyl. 26,

2010. doi: 10.1145/1864708.1864747. Available: http://dx.doi.org/10.1145/1864708.1864747

[12] Y. Bengio, Y. Lecun, ve G. Hinton, “Deep learning for AI”, Communications of the ACM, c. 64, sy 7. Association for Computing Machinery (ACM), ss. 58-65, Haz. 21, 2021. doi: 10.1145/3448250. Available: http://dx.doi.org/10.1145/3448250

[13] Y. Peng vd., “Cross-media analysis and reasoning: advances and directions”, Frontiers of Information Technology & Electronic Engineering, c. 18, sy 1. Zhejiang University Press, ss. 44-57, Oca. 2017. doi: 10.1631/fitee.1601787. Available: http://dx.doi.org/10.1631/fitee.1601787

[14] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, ve H. Jing, “Recurrent Recommender Networks”, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, Şub. 02, 2017. doi: 10.1145/3018661.3018689. Available: http://dx.doi.org/10.1145/3018661.3018689

[15] A. M. Elkahky, Y. Song, ve X. He, “A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems”, Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, May. 18, 2015. doi: 10.1145/2736277.2741667. Available: http://dx.doi.org/10.1145/2736277.2741667

[16] Y. Song, A. M. Elkahky, ve X. He, “Multi-Rate Deep Learning for Temporal Recommendation”, Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, Tem. 07, 2016. doi: 10.1145/2911451.2914726. Available: http://dx.doi.org/10.1145/2911451.2914726

[17] F. Vasile, E. Smirnova, ve A. Conneau, “Meta-Prod2Vec”, Proceedings of the 10th ACM Conference on Recommender Systems. ACM, Eyl. 07, 2016. doi: 10.1145/2959100.2959160. Available: http://dx.doi.org/10.1145/2959100.2959160

[18] M. Grbovic vd., “E-commerce in Your Inbox”, Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Ağu. 10, 2015. doi: 10.1145/2783258.2788627. Available: http://dx.doi.org/10.1145/2783258.2788627

[19] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, ve D. Estrin, “Collaborative Metric Learning”, Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, Nis. 03, 2017. doi: 10.1145/3038912.3052639. Available: http://dx.doi.org/10.1145/3038912.3052639

[20] S. Roy ve S. C. Guntuku, “Latent Factor Representations for Cold-Start Video Recommendation”, Proceedings of the 10th ACM Conference on Recommender Systems. ACM, Eyl. 07, 2016. doi: 10.1145/2959100.2959172. Available: http://dx.doi.org/10.1145/2959100.2959172

[21] S. Li, J. Kawale, ve Y. Fu, “Deep Collaborative Filtering via Marginalized Denoising Auto-encoder”, Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, Eki. 17, 2015. doi: 10.1145/2806416.2806527. Available: http://dx.doi.org/10.1145/2806416.2806527

[22] L. Zheng, V. Noroozi, ve P. S. Yu, “Joint Deep Modeling of Users and Items Using Reviews for Recommendation”, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, Şub. 02, 2017. doi: 10.1145/3018661.3018665. Available: http://dx.doi.org/10.1145/3018661.3018665

[23] S. Zhang, L. Yao, A. Sun, ve Y. Tay, “Deep Learning Based Recommender System”, ACM Computing Surveys, c. 52, sy 1. Association for Computing Machinery (ACM), ss. 1-38, Şub. 25, 2019. doi: 10.1145/3285029. Available: http://dx.doi.org/10.1145/3285029

[24] T. Bansal, D. Belanger, ve A. McCallum, “Ask the GRU”, Proceedings of the 10th ACM Conference on Recommender Systems. ACM, Eyl. 07, 2016. doi: 10.1145/2959100.2959180. Available: http://dx.doi.org/10.1145/2959100.2959180

[25] Y. Wu, C. DuBois, A. X. Zheng, ve M. Ester, “Collaborative Denoising Auto-Encoders for Top-N Recommender Systems”, Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, Şub. 08, 2016. doi: 10.1145/2835776.2835837. Available: http://dx.doi.org/10.1145/2835776.2835837

[26] J. Wang, K. Kenthapadi, K. Rangadurai, ve D. Hardtke, “Dionysius: A Framework for Modeling Hierarchical User Interactions in Recommender Systems”. arXiv, 2017. doi: 10.48550/ARXIV.1706.03849. Available: https://arxiv.org/abs/1706.03849

[27] S. Zhang, L. Yao, A. Sun, ve Y. Tay, “Deep Learning Based Recommender System”, ACM Computing Surveys, c. 52, sy 1. Association for Computing Machinery (ACM), ss. 1-38, Şub. 25, 2019. doi: 10.1145/3285029. Available: http://dx.doi.org/10.1145/3285029

[28] M. Wang, “Applying Internet information technology combined with deep learning to tourism collaborative recommendation system”, PLOS ONE, c. 15, sy 12. Public Library of Science (PLoS), s. e0240656, Ara. 03, 2020. doi: 10.1371/journal.pone.0240656. Available: http://dx.doi.org/10.1371/journal.pone.0240656

[29] M. Çakır, Ş. G. Öğüdücü, ve R. Tugay, “A Deep Hybrid Model for Recommendation Systems”, Lecture Notes in Computer Science. Springer International Publishing, ss. 321-335, 2019. doi: 10.1007/978-3-030-35166-3_23. Available: http://dx.doi.org/10.1007/978-3-030-35166-3_23

[30] P. Covington, J. Adams, ve E. Sargin, “Deep Neural Networks for YouTube Recommendations”, Proceedings of the 10th ACM Conference on

Recommender Systems. ACM, Eyl. 07, 2016. doi: 10.1145/2959100.2959190. Available: http://dx.doi.org/10.1145/2959100.2959190

[31] H.-T. Cheng vd., “Wide & Deep Learning for Recommender Systems”, Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, Eyl. 15, 2016. doi: 10.1145/2988450.2988454. Available: http://dx.doi.org/10.1145/2988450.2988454

[32] M. Naumov vd., “Deep Learning Recommendation Model for Personalization and Recommendation Systems”. arXiv, 2019. doi: 10.48550/ARXIV.1906.00091. Available: https://arxiv.org/abs/1906.00091

[33] L. Breiman, Machine Learning, c. 45, sy 1. Springer Science and Business Media LLC, ss. 5-32, 2001. doi: 10.1023/a:1010933404324. Available: http://dx.doi.org/10.1023/a:1010933404324

[34] V. Sze, Y.-H. Chen, T.-J. Yang, ve J. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey”. arXiv, 2017. doi: 10.48550/ARXIV.1703.09039. Available: https://arxiv.org/abs/1703.09039

[35] S. K. Schultz, “Principles of Neural Science, 4th ed.,” American Journal of Psychiatry, vol. 158, no. 4, p. 662, Apr. 2001, doi: 10.1176/appi.ajp.158.4.662. Available: https://doi.org/10.1176/appi.ajp.158.4.662

[36] “What is the differences between artificial neural network (computer science) and biological neural network?,” Quora. Available: https://www.quora.com/What-is-the-differences-between-artificial-neural-network-computer-science-and-biological-neural-network

[37] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. “O’Reilly Media, Inc.,” 2019..

[38] F. Chollet, Deep Learning with Python. Manning Publications, 2017.

[39] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016.

[40] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning: Data Mining, Inference, and Prediction, Second Edition. Springer Science & Business Media, 2009.

[41] C. M. . Bishop, Pattern recognition and machine learning: Cram101 Textbook Outlines to Accompany. 2011.

[42] P. Ramachandran, B. Zoph, ve Q. V. Le, “Searching for Activation Functions”. arXiv, 2017. doi: 10.48550/ARXIV.1710.05941. Available: https://arxiv.org/abs/1710.05941

[43] D. E. Rumelhart, G. E. Hinton, ve R. J. Williams, “Learning representations by back-propagating errors”, Nature, c. 323, sy 6088. Springer Science and Business Media LLC, ss. 533-536, Eki. 1986. doi: 10.1038/323533a0. Available: http://dx.doi.org/10.1038/323533a0

[44] L. Devroye, “Complexity Questions in Non-Uniform Random Variate Generation”, Proceedings of COMPSTAT’2010. Physica-Verlag HD, ss. 3-18,

2010. doi: 10.1007/978-3-7908-2604-3_1. Available: http://dx.doi.org/10.1007/978-3-7908-2604-3_1

[45] S. Ruder, “An overview of gradient descent optimization algorithms”. arXiv, 2016. doi: 10.48550/ARXIV.1609.04747. Available: https://arxiv.org/abs/1609.04747

[46] R.-Y. Sun, “Optimization for Deep Learning: An Overview”, Journal of the Operations Research Society of China, c. 8, sy 2. Springer Science and Business Media LLC, ss. 249-294, Haz. 2020. doi: 10.1007/s40305-020-00309-6. Available: http://dx.doi.org/10.1007/s40305-020-00309-6

[47] 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

[48] S. Ioffe ve C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. arXiv, 2015. doi: 10.48550/ARXIV.1502.03167. Available: https://arxiv.org/abs/1502.03167

[49] G. V. Jose, “Useful Plots to Diagnose your Neural Network - Towards Data Science,” Medium, Dec. 12, 2021. Available: https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45

[50] Y. Lecun, L. Bottou, Y. Bengio, ve P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, c. 86, sy 11. Institute of Electrical and Electronics Engineers (IEEE), ss. 2278-2324, 1998. doi: 10.1109/5.726791. Available: http://dx.doi.org/10.1109/5.726791

[51] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines”, Lecture Notes in Computer Science. Springer Berlin Heidelberg, ss. 599-619, 2012. doi: 10.1007/978-3-642-35289-8_32. Available: http://dx.doi.org/10.1007/978-3-642-35289-8_32

[52] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, Jan. 2014, Available: https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf

[53] “Total data volume worldwide 2010-2025 | Statista,” Statista, Sep. 08, 2022. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

[54] A. Ansari, S. Essegaier, ve R. Kohli, “Internet Recommendation Systems”, Journal of Marketing Research, c. 37, sy 3. SAGE Publications, ss. 363-375, Ağu. 2000. doi: 10.1509/jmkr.37.3.363.18779. Available: http://dx.doi.org/10.1509/jmkr.37.3.363.18779

[55] G. Adomavicius ve A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, IEEE Transactions on Knowledge and Data Engineering, c. 17, sy 6. Institute of Electrical and Electronics Engineers (IEEE), ss. 734-749,

Haz. 2005. doi: 10.1109/tkde.2005.99. Available: http://dx.doi.org/10.1109/tkde.2005.99

[56] F. O. Isinkaye, Y. O. Folajimi, ve B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, c. 16, sy 3. Elsevier BV, ss. 261-273, Kas. 2015. doi: 10.1016/j.eij.2015.06.005. Available: http://dx.doi.org/10.1016/j.eij.2015.06.005

[57] M. Uzun-Per, A. B. Can, A. Volkan Gurel, ve M. S. Aktas, "Big Data Testing Framework for Recommendation Systems in e-Science and e-Commerce Domains", 2021 IEEE International Conference on Big Data (Big Data). IEEE, Ara. 15, 2021. doi: 10.1109/bigdata52589.2021.9672082. Available: http://dx.doi.org/10.1109/bigdata52589.2021.9672082

[58] M. Uzun‐Per, A. V. Gurel, A. B. Can, ve M. S. Aktas, "Scalable recommendation systems based on finding similar items and sequences", Concurrency and Computation: Practice and Experience, c. 34, sy 20. Wiley, Oca. 20, 2022. doi: 10.1002/cpe.6841. Available: http://dx.doi.org/10.1002/cpe.6841

[59] S. Sancar ve M. Uzun-Per, "Testing the Performance of Feature Selection Methods for Customer Churn Analysis: Case Study in B2B Business", Computational Intelligence, Data Analytics and Applications. Springer International Publishing, ss. 509-519, 2023. doi: 10.1007/978-3-031-27099-4_39. Available: http://dx.doi.org/10.1007/978-3-031-27099-4_39

[60] S. Sancar ve M. Uzun-Per, "Testing the Performance of Feature Selection Methods for Customer Churn Analysis: Case Study in B2B Business", Computational Intelligence, Data Analytics and Applications. Springer International Publishing, ss. 509-519, 2023. doi: 10.1007/978-3-031-27099-4_39. Available: http://dx.doi.org/10.1007/978-3-031-27099-4_39

TEZDEN ÜRETİLMİŞ YAYINLAR

Makaleler

1. Alihan Ö., Meryem U. & Mert B. (2023) *Developing a Deep Learning Based Travel Recommendation System (Versiyon 1). Avrupa Bilim ve Teknoloji Dergisi.*