YBS Ansiklopedi

www.YBSAnsiklopedi.com

Cilt 8 Sayı 1, Ocak 2020

OptiScorer: Otomatik Makine Öğrenmesi ile Skorlama

Şadi Evren ŞEKER[1]

1.OptiWisdom Inc. ve Antalya Bilim Üniversitesi, Bilgisayar Mühendisliği Bölümü

Özet

Makine öğrenmesinde son yıllarda yaşanan baş döndürücü gelişmeler ve hayatın her aşamasında karşılaşılan uygulamalarını 3. nesil yapay zeka yaklaşımı olarak otomatik makine öğrenmesi başlığı altında incelediğimiz bu makalede, otomatik makine öğrenme yaklaşımının işletmeler için kullanım alanları ve veri bilimi dünyasında yapacağı sarsıcı etkiyi ele almaya çalıştık. Yaşanmakta olan yüksek teknoloji dönüşümleri ve ulaşılmaz yapay zeka yaklaşımından, KOBİ ölçeğinde ve her işletmede çalışabilir uygulamalara indirgeme yapılmış, yapay zekanın demokratizasyonu ve herkes için yapay zeka yaklaşımı belirlenmiş ve vatandaş veri bilimci kavramına uygun şekilde konulara açıklık getirilmeye çalışılmıştır. Bir uygulama olarak otomatik makine öğrenmesi ve açıklanabilir yapay zeka yaklaşımlarını kullanan OptiScorer yapay zeka motoru üzerinden sürecin nasıl çalıştığı ve motorun aşamaları açıklanmıştır.

Anahtar Kavramlar: *Otomatik Makine Öğrenmesi, Açıklanaiblir Yapay Zeka, SaaS, Veri Bilimi, 3ncü nesil yapay zeka*

Abstract

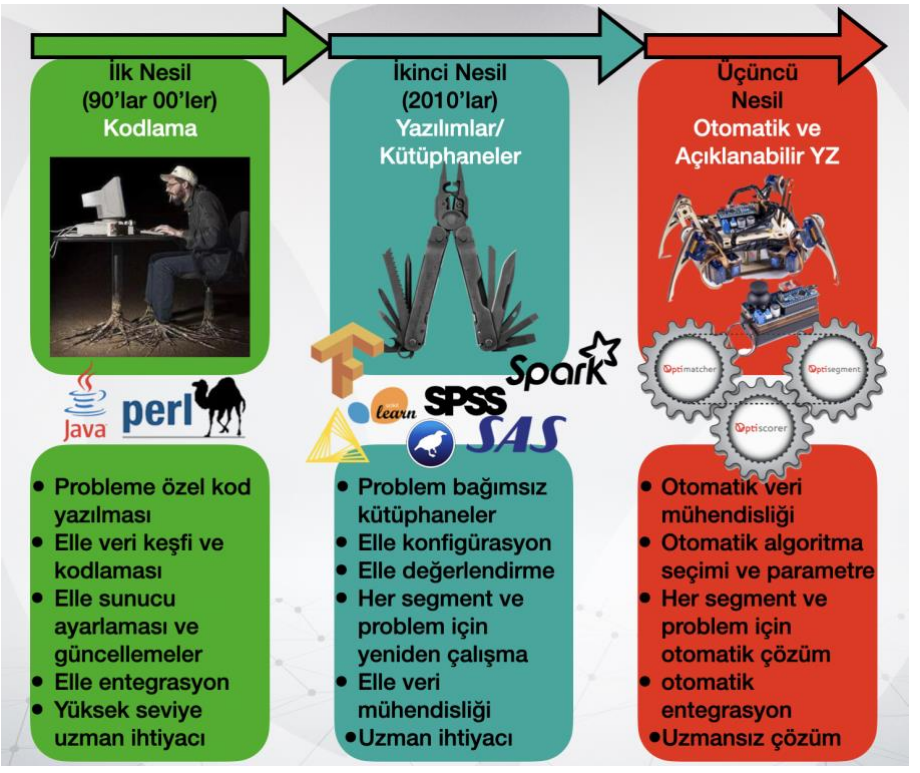
In this study, we have examine the high-edge developments in machine learning in recent years and its applications encountered at every stage of life under the title of automatic machine learning as a 3rd generation artificial intelligence approach, we tried to address the shocking impact that automatic machine learning approach will have on businesses and the world of data science. From the high technology transformations and inaccessible artificial intelligence approach, to applications that can work in SME scale and in every stage of enterprises, the democratization of artificial intelligence and artificial intelligence approach were accessible fo everyone, and the issues were tried to be clarified in accordance with the concept of citizen data scientist. The OptiScorer artificial intelligence engine, which uses automatic machine learning and explainable artificial intelligence approaches as an application, explains how the process works and the stages of the engine.

Keywords: *Automatic Machine Learning (AutoML), Explainable AI (XAI), SaaS, Data Science, 3rd Gen AI (3GAI)*

1.Giriş Ve Tanım

Otomatik makine öğrenmesi kavramı, makine öğrenme literatürüne son yıllarda girmiş ve hızla endüstride uygulama bulmuştur [1-5]. Temel olarak makine öğrenme sürecini 3 aşamada incelemek mümkündür. Bunlardan ilki 90'ların sonu ve iki binlerin başına kadar süren kodlama evresidir. Bu evrede, hemen her türlü proje baştan kodlanmakta, çoğu uygulama için geliştirici ekibin kodun içerisine girerek müdahale etmesi gerekmektedir. İkinci nesil yapay zeka yaklaşımında, makine öğrenme dünyasına çok sayıda kütüphane ve araç girdiği söylenebilir. Örneğin

Python üzerinde çalışan ve bu makalenin yazıldığı tarih itibariyle de çok sayıda uygulaması olan sci-kit learn veya TensorFlow, Keras [6] gibi kütüphaneler artık her proje için yeniden kod yazma ihtiyacını ortadan kaldırmış ve projede yer alan veri bilimcilerin hızlıca projeyi uygulamaya geçirmesinin yolunu açmıştır. Benzer şekilde, kodlamadan çok veriye ve çözülmeye çalışılan probleme odaklanan yeni roller çıkmış ve veri analistliği, iş analistliği gibi meslekler doğmuş, hatta hiç kodlama bilmeden veri bilimi çözümleri geliştirmenin yolları açılmıştır. Örneğin, kodlamasız (no-code) yaklaşımı ile geliştirilen Knime , Rapid Miner, SPSS Modeler [7] gibi çok sayıda araç bu grupta sayılabilir ve bütün bu kütüphane ve araçların ikinci nesil yapay zeka yaklaşımının birer ürünü olduğu söylenebilir.



Şekil 1: 3. Nesil Yapay Zeka yaklaşımı

Özellikle 2015 sonrası başlayan ve 2020'lere gelindiği dönemde ise endüstride otomatik makine öğrenmesinin kullanımının arttığını söyleyebiliriz. Şekil 2'de de gösterilen bu 3. nesil yeni yaklaşıma göre, bir veri bilimcinin her projede yeniden bütün olası algoritmaları denemesi, bu algoritmaların parametrelerini optimize etmesi veya çoğu durumda birden fazla algoritmayı kombinleyerek (ansamble) kullanması ve bu denemeler süresince vakit kaybetmesi artık gerekmemektedir. Hatta çoğu aracın içerisinde bu özellik sunulmaktadır. Örneğin Rapid Miner veya SPSS Modeler artık bu denemeleri otomatik yapan çözümleri içerisinde barındırmaya başlamıştır. Sektörde bilinen ve pazarı bulunan mevcut oyuncuların dışına gelişen bir akım ile de otomatik makine öğrenme süreçlerine özel araçların geliştirildiğini söyleyebiliriz. Örneğin H2O [8], Tazi, gibi araçlar bu sürecin öncülerinde görülebilir.

Bu makalede, otomatik makine öğrenme süreçlerini bir de OptiScorer örneği üzerinden anlatmanın yanında, otomatik makine öğrenme döngüsünün otomatik veri ön işleme döngüsü ile nasıl birleştiğinden de bahsedilecektir. Basitçe, çoğu otomatik makine öğrenme çözümü, veri ön işleme süreçlerinden bağımsız çalışmakta ve bu süreçleri

hala veri bilimci, veri analisti veya iş analistlerinin çözmesini beklemektedir. Hatta çoğu uygulamada öncesinde veri kalitesini çözecek bir veri ambarı dönüşümü şart görülmektedir. Sektördeki çoğu benzerinden farklı olarak OptiScorer bu aşamada da çözüm önerileri sunmakta ve bunun ötesinde iki döngüyü birbirine bağlayan uçtan uca bir yapıyı çalıştırmaktadır. Sektörde, yine veri ön işleme döngüsüne özel çözümler de göze çarpmaktadır. Örneğin MIT laboratuvarlarında başlatılan, ve deep feature synthesis (DFS) ve daha sonra ismini FeatureTools olarak değiştiren ve daha sonra Alteryx tarafından alınan ön işleme aracının amacı daha çok veriyi otomatik zenginleştirme olarak görülebilir [9]. Buna benzer araçlar, veri ön işleme aşamasındaki problemleri hedeflemekte ve çözmektedir. Ancak yine de çok sayıdaki farklı probleme odaklanan farklı aracın birlikte çalıştırılması (orkestrasyonu) bir uzmanın özel olarak kurulum yapmasını gerektirmektedir.

Bu makale kapsamında, gözetimsiz (denetimsiz, unsupervised) öğrenme veya pekiştirmeli öğrenme gibi bütün alanları kapsayan çözümler üzerinde otomatik makine öğrenmesi süreçlerine girilmeyecek; sadece skorlama olarak ifade edilen problem kümesi incelenecektir. Örneğin müşteri risk skorlama, müşteri değeri skorlama (customer lifetime value), kredi risk skorlama, fiyat skorlama, gibi sayısal sonuçları olan tahmin süreçleri (prediction) ve bazı durumlarda zamana bağlı tahminler ve regresyonlar (regression) bu çalışmanın bir parçası olarak görülebilir. Buna göre, bir müşteriye çarpraz satış yapılabilme ihtimali (cross-sell) veya müşterinin bir üst pakete geçme ihtimali (upsell) veya müşterinin üyeliğini sonlandırma veya kaybedilme ihtimali (customer churn) birer skorlama problemi olarak kurgulanabilir ve bütün bu süreçlerin sonunda bir skor değeri üretilmektedir. Verinin zaman bağlı şekilde işlenmesi ve zamana göre bu skorların değişmesi (örneğin borsadaki hisse senetlerinin fiyatlarının değişmesi) gibi durumlarda ise skorlama süreci zamana bağlı bir hal almaktadır ve algoritmalarda farklı yaklaşımlar gerektirmektedir, ancak yine de skorlamanın bir uygulaması olarak görülebilir.

Verinin uçtan uca aktığı ve gerekli noktalarda kontroller ve güncellemeler yapılarak verinin üzerinde döngüler oluşturulduğu bir skorlama çözümü 2017 yılından beri, OptiScorer ile sunulmaktadır.

2. Otomatik Makine Öğrenmesi ve Problemleri

Giriş bölümünde de anlatıldığı gibi, makine öğrenmesi süreci auto-weka [10] çalışmasını içeren makalede de ilk defa bahsedildiği üzere kombine algoritma seçimi ve hiper parametre optimizasyonu (combined algorithm selection and hyper parameter optimization, CASH) problemi olarak özetlenebilir. Bu problem basitçe, birden fazla algoritmadan oluşan kombine (ensemble) seçimleri ve bu seçimler üzerinde yapılacak olan hiper parametre optimizasyonlarını içermektedir. Örneğin, sektörde yaygın olarak kullanılan müşteri kayıp analizi (customer churn analysis) problemini ele alalım. Bu problemin çözümü sırasında alternatif olarak kullanılabilecek 20 civarı algoritmamızın hazır olarak kütüphanede beklediğini ve bu kütüphanedeki çözümlerin kombinasyonlarının (ensemble) alternatif çözümler oluşturabileceğini düşünelim, ve şimdilik süreci basit tutmak adına en fazla 3 algoritmanın birlikte çalışabileceği alternatifleri düşünelim.

$$\binom{20}{3} = \frac{20!}{3!\,(20-3)!} = 1140 \qquad (1)$$

Basit bir hesaplama ile denklem (1)'de görüleceği üzere 1140 farklı alternatifin denenmesi gerekir. 3 yerine sadece 5 farklı algoritmanın birlikte çalışabileceği alternatiflere bakılırsa da bu sayı bir anda 15504'e çıkacaktır. Demek oluyor ki sadece 20 algoritmadan oluşan bir kütüphanede bile binlerce farklı algoritma alternatifinin birlikte denenmesi gerekmektedir.

Bu sürece, her algoritmanın kendi hiper parametre optimizasyonu da eklenecek olursa problem çok daha karmaşık bir hal almaktadır. Örneğin kütüphanemizdeki makine öğrenme algoritmalarından bir tanesinin de k-en yakın komşu (KNN) algoritması olduğunu düşünelim [11] . Bu durumda KNN algoritmasındaki K değerinin kaç olacağından, kullanılacak olan sayısal mesafe algoritmasın (örneğin, manhattan, öklit, chebyshev gibi) veya

kullanılacak olan dizgi mesafe algoritmasına (örneğin levenstein, jaccard, hamming vs.) [12] kadar çok farklı karar verilmesi gereken nokta bulunmaktadır. Yine basit olarak minkowski mesafesi alınacak olursa bu sefer de minkowsky mesafe algoritmasındaki p değerine karar vermek için denemeler yapılması gerekecektir.

Hemen her algoritmanın başarısını etkileyen bu hiper-parametre değerleri, makine öğrenmesi algoritmaları ile çözülememektedir. Zaten çözülebilecek olsa, algoritmanın bir parçası haline getirilir ve çözümün parçası olarak sunulur. Bu aşamada optimizasyon algoritmalarından faydalanmak, literatürde sık rastlanan bir çözüm olarak karşımıza çıkıyor. Ancak problemi daha net görebilmek adına, örneğin 100.000 satırlı bir veri kümesinde, ki günümüzdeki büyük veri çalışmaları göz önüne alındığında bu oldukça mütevazi bir veri kümesidir, K-NN gibi basit bir algoritmada bile, veri kümesindeki eleman sayısının karekköüne kadar farklı K değerlerinin denenmesi gerekir.

$$\sqrt{100000} \cong 316$$

Özetle, algoritma seçildikten sonra bile, küçük bir veri kümesi için bile tek bir algoritmanın yüzlerce kere çalışması gerekmektedir. Ayrıca bu hesaplamanın içerisinde kullanılacak olan mesafe algoritmaları ve bu algoritmaların parametre optimizasyonları veya K-NN algoritmasının hafıza optimizasyonu için kullanılabilecek olan KD Ağaçları, Centroid yaklaşımı veya BallTree algoritması gibi yaklaşımlardan bahsedilmedi bile.

Kütüphanemize geri dönecek olursak, 20 algoritmalık mütevazi kütüphanemizdeki her algoritmanın farklı parametrelerinin her biri için yüzlerce deneme, dolayısıyla her algoritma için binlerce denemeden oluşan ve bu binelrce denemenin birlikte çalışacak olan algoritma kombinasyonları için on binlerce denemeyle çarpımından milyonlarca algoritma denemesine ulaşmak oldukça mümkün.

Veri bilimcilerin elle yaptığı ve tecrübelerine dayalı, sezgisel olarak yapılan bu denemelerin çoğunda gözden kaçan denemeler olduğu kesindir. Yine de sektörde ortalama 8 hafta süren uçtan uca veri bilimi çalışmalarını, hem daha kısa sürede hem de bütün denemelerin hepsini makineye yaptırarak bulmak mümkün olsa da en iyi alternatifi çoğu zaman gerçek dünya uygulamalarında otomatik makine öğrenmesi kullanan paydaşların bu kadar bile beklemek istemediği söylenebilir.

Bu durum, bütün denemeleri yapmak yerine, daha hızlı çözüme ulaşmamızı sağlayacak alternatif yaklaşımları gerekli kılmaktadır. Örneğin, yukarıda bahsedilen ve bütün alternatiflerin bir tablo ile denenerek sonuçlarının karşılaştırıldığı grid-search yaklaşımına bir alternatif olarak rasgele arama (random-search) sunulabilir, ancak bu durumda da çözüm olabilecek en iyi alternatifi gözden kaçırma riski bulunmaktadır. Bu iki problem, yani hem en iyi çözümü bulmak hem de bütün ihtimalleri denememek için optimizasyon algoritmalarına başvurulmuş ve parametre optimizasyonu çözümleri üretilmiştir. Günümüzde farklı parametre optimizasyon yöntemleri olmakla birlikte OptiScorer içeisinde Bayes optimizasyon algoritmalarından faydalanılmaktadır.

3. Otomatik Veri Ön İşleme ve Problemleri

Veri bilimi sürecinin önemli bir aşaması da veri ön işleme sürecidir ve çoğu veri bilimi projesinin en önemli maliyetini ve en uzun süren aşamasını oluşturur. Doğru veri kaynaklarının bulunması ve verilerin taranarak işe yarar olanlarının seçilmesi; veriler üzerindeki eksik, kirli veya gürültülü veri problemlerinin çözülmesi gibi çok sayıda veri ön işleme sürecinden bahsedilebilir ancak otomatik veri ön işleme aşamasının en önemli gereksinimlerinden birisi de verinin otomatik olarak zenginleştirilmesidir. Otomatik Makine Öğrenmesi ve Problemleri bölümünde bahsettiğimiz müşteri yayılım problemine dönecek olursak, bu problemdeki veri kaynağında çok büyük ihtimalle müşterinin yaptığı alışverişler ve tekrarları yer alacaktır. Yani, bir müşteri bir yıl içerisinde 10 kere firmadan alışveriş yaptıysa bütün bu bilgiler alışverişlerin tutulduğu veri kaynağında geçecektir.

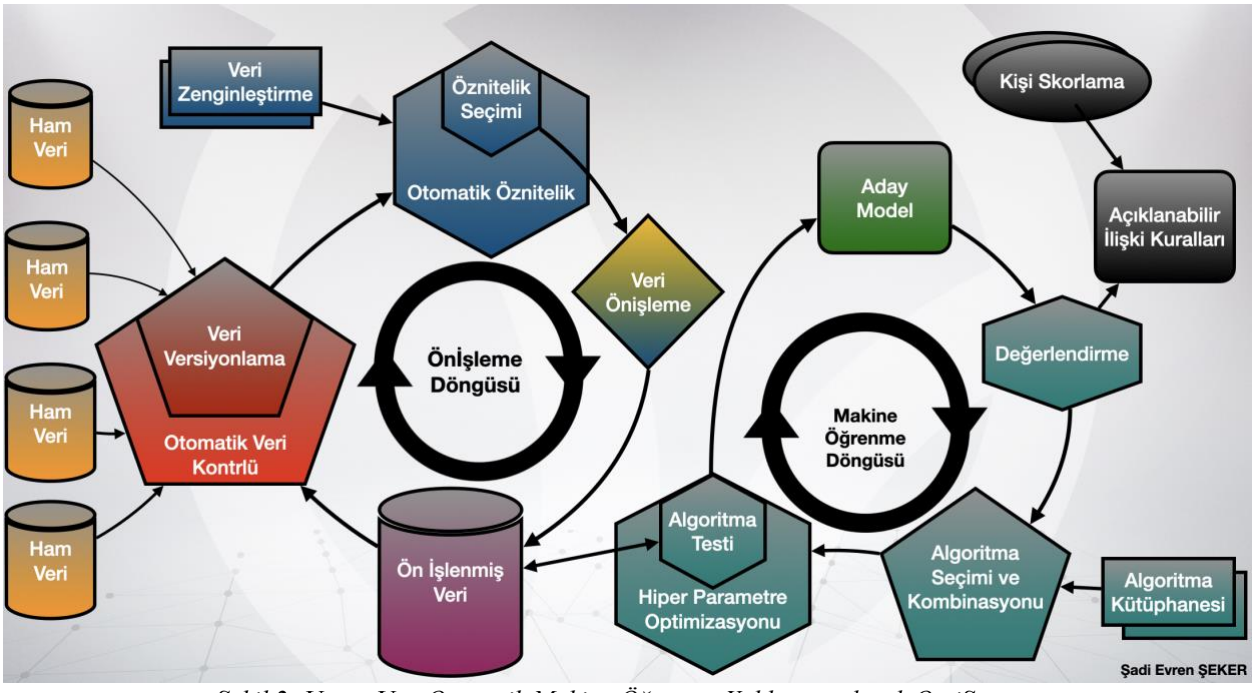
Veri zenginleştirme sürecinde kullanılan yöntemlerden birisi de, aynı eşsiz belirleyici (unique identifier, ID) için tekrarlı bilgilerin zenginleştirilerek tekilleştirilmesidir. Buna göre tekrarlı veri önce tek bir ID değerine indirgenecek ve bu tekrarlı veri için ilave kolonlar açılarak bu kolonlarda, müşterinin yaptığı maksimum, minimum, ortalama veya toplam alışverişlere yer verilebilecek, yine tarih alanı için müşterinin ilk veya son alışveriş tarihi, dolayısıyla müşterinin ne kadar yeni olduğu veya müşterinin ne kadar sıklıkla geldiği gibi bilgilere ulaşılabilecektir.

Benzer şekilde tarih alanı içeren bir veri kümesinde, verinin tarih bilgisinin, haftanın günü, ayın günü, yılın kaçıncı haftası olduğu, veya yılın kaçıncı ayı olduğu gibi bilgilerin çıkarılmasıyla daha da zenginleştirilmesi mümkündür. Bu otomatik veri zenginleştirmeleri, bazı algoritmaların veriye daha farklı açılardan bakmasını ve daha yüksek başarı göstermesini sağlayacaktır.

Veri zenginleştirmenin sebep olduğu önemli bir problem de verinin gereğinden fazla büyümesidir. Büyüklük sorunu, çalışan bütün algoritmalar ve verinin saklanması aşamasında problem olarak karşımıza çıkar ve çözümü için verini küçültülmesinin yolları aranabilir. Bir çözüm olarak veride gerçekten sonuca etki eden özniteliklerin seçilmesi ve gereksiz olan verinin temizlenmesi, diğer bir çözüm olarak da verinin farklı bir boyuta dönüştürülmesi veya boyut indirgeme operasyonları düşünülebilir.

4. OptiScorer: Uçtan Uca bir Çözüm

OptiScorer, uçtan uca otomatik veri ön işleme ve makine öğrenme süreçlerini birlikte sunan ve bu konuda öncü rol oynayan yapay zeka motorudur. Moturun genel akışı Şekil 2'de gösterilmiştir:



Şekil 2: Uçtan Uca Otomatik Makine Öğrenme Yaklaşımı olarak OptiScorer

Soldan sağa doğru akışın ifade edildiği Şekil 2'de, veri öncelikle ham hali ile farklı veri kaynaklarından alınmakta ve otomatik veri kontrolü aşamasında verinin güncelliğinin ve eşleşmesinin kontrolü sağlanmaktadır. Bu aşamada verinin tekilleştirilmesi, ve güncellenen veri akışının takip edilerek verilerin güncel tutulması süreçleri

işletilmektedir. Aynı zamanda verinin ön işleme döngüsü içerisinde birden fazla kere farklı versiyonlarına ihtiyaç duyulacağı veya değişen veri kaynaklarına göre verinin güncelleneceği düşünülürse, verinin versiyonlaması ve gerekli durumlarda eski versiyonlara dönülerek işlem yapılması da mümkündür.

Ön işleme döngüsü, Veri Zenginleştirme kütüphanesine yüklenmiş çok sayıda veri zenginleştirme fonksiyonu ve algoritmasını veri üzerinde çalıştırır. 3. bölümde bahsedilen veri zenginleştirme algoritmalarına benzer şekilde, tekrarlı verilerin tekilleştirilmesi sırasında verinin tekrar hızı (frekansı), ilk ve son veri, maksimum, minimum, toplam veya ortalama gibi toparlayıcı fonksiyonlar (aggregate functions), kullanılır. Sonuçta orijinale göre çok daya büyük boyutlara erişen verinin küçültülmesi için patent başvurusu yapılan verinin seçimi (feature elimination) ve boyut dönüştürme yöntemlerinden harmanlanan yöntemler kullanılarak veri daha verimli işlenebilecek ve gerçekten sonuca etki eden bir hale indirgenir. Veri ön işleme aşamasının, ilk geçişte, daha önceki aşamalardan çıkan veri ile ilgili kirli, gürültülü veya marjinal verilerin tespit edilmesi için kullanılan bir aşama olmakla birlikte, esas fonksiyonu makine öğrenme aşamasından gelen bazı sinyallere göre veri üzerinde normalleştirme, standartlaştırma veya kodlama (encoding) gibi işleri de yapmaktır.

Makine öğrenme döngüsü ise ön işlemesi tamamlanmış veri üzerinde çalışır. Makine öğrenme döngüsüne geçmeden önce bazı durumlarda zamansal analizler (temporal) çalıştırılmaktadır [14]. Örneğin veri üzerinde sezonsallık analizleri veya trend analizleri yapılarak bazı durumlarda veri farklı şekillerde de zenginleştirilmektedir.

İkinci aşamada makine öğrenme döngüsü başlatılır ve bu aşamada, daha önceden tanımlanmış olan makine öğrenme ve istatistiksel modellerden oluşan kütüphanede bulunan algoritmalar veri üzerinde CASH probleminin çözümüne uygun olarak çalıştırılır. OptiScorer, bu aşamada algoritma seçimi, seçilen algoritmaların kombinasyonlarının denenmesi ve hiper parametre optimizasyonları için Bayes optimizasyon yöntemlerini kullanmakta ve arama uzayını azaltarak yüksek oranda başarı elde etmektedir.

Makine öğrenme algoritmasından çıkan aday çözümler, değerlendirme aşamasında incelenmekte ve yeterli başarı sağlanıyorsa sisteme alınmaktadır. Buradaki değerlendirme kriterleri 3 farklı şekilde incelenebilir:

- İç kriterler
- Dış kriterler
- Saha Testleri

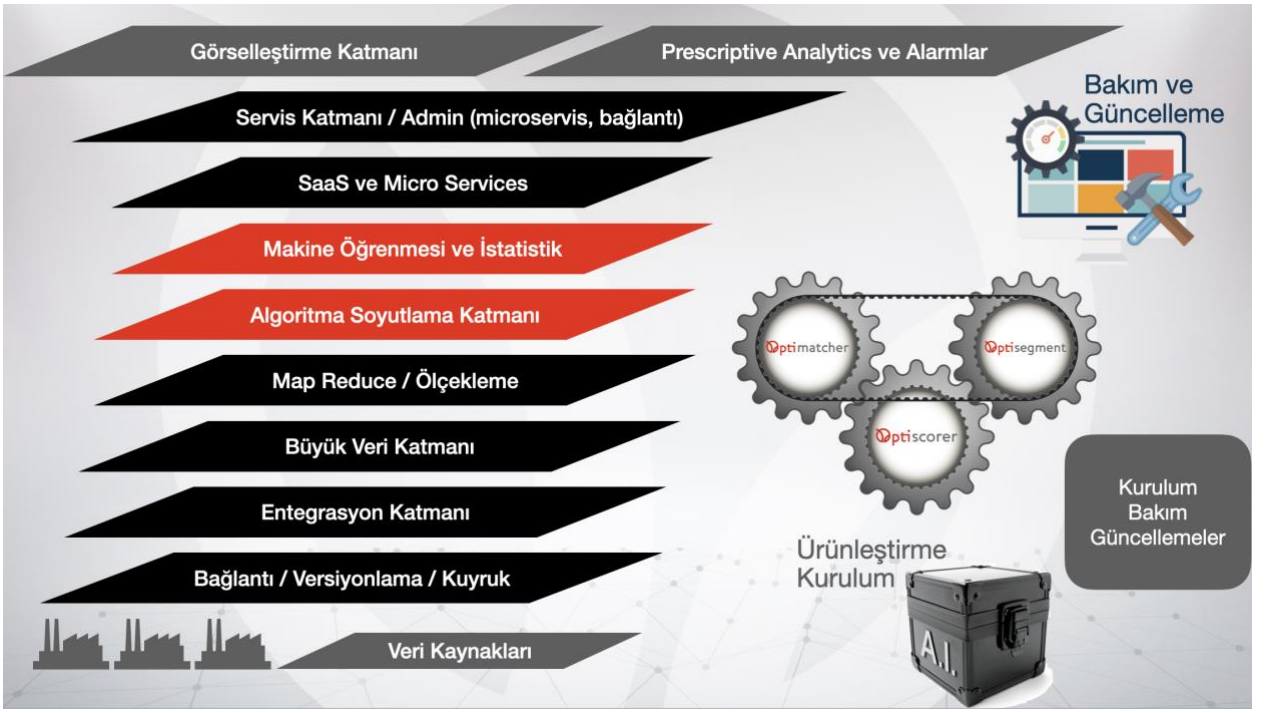
İç kriterler, daha çok matematiksel fonksiyonlarla yazılabilen hata karaleri ortalamasının karekökü (RMSE), ortalama mutlak hata (MAE), r^2 gibi ölçülebilir değerlerdir [13]. Dış kriterler ise makine öğrenme sürecinden sonra çalıştırılabilecek dış fonksiyonlardır, geçmiş verilere bağlı back test, veya simülasyon çalışmaları bu grupta düşünülebilir. Son olarak saha testleri, gerçek veri ile gerçek kullanım üzerinde yapılan testlerdir. Gerek analizine göre bazı durumlarda sadece iç kriterler test edilirken bazı durumlarda bu testler saha testlerine kadar ilerletilmekte ve sistem saha testlerini canlı olarak çalıştırarak sonuçlardan emin bir sistem inşa edene kadar eğitim sürecine devam etmektedir.

Sistemin en kritik ve bir patent başvuru konusu olan bağlantı noktası, makine öğrenme ve veri önişleme döngüleri arasındaki köprüdür. Bu köprü iki taraflı çalışmakta ve seçilen algoritmaların ön işleme kısmında farklı beklentilere yol açmasından dolayı gerekli durumlarda ön işlemeyi tekrar tetiklemektedir. Örneğin seçilen algoritmaya göre verinin kodlanması veya normalize edilmesi gibi ihtiyaçlar bu aradaki köprü tarafından otomatik olarak çözülmektedir.

Aradaki köprünün diğer bir görevi de veri ön işlemedeki veri seçimi ve seçilen verilerin algoritmalardaki etkisi arasında bağlantı kurarak, açıklanabilir yapay zeka sonuçları çıkarmak ve girdi olan veriler ile sonuçtaki veriler arasındaki açıklama sürecini başlatmaktır. Bu sayede OptiScorer motoru, çıktı olarak öznitelik önemlerini de sunabilmektedir.

5. Büyük Veri ve Otomatik Makine Öğrenmesi

Otomatik makine öğrenmesi çözümlerinin karşılaştığı önemli problemlerden birisi de çözümlerin paralel olarak çalıştırılmasıdır.



Şekil 3. Ölçeklenebilir ve Yönetilebilir Otomatik Makine Öğrenmesi

Daha önce şekil 2'de bahsedilen uçtan uca çözüm, şekil 3'teki makine öğrenmesi ve istatistik katmanına karşılık gelmektedir. Ancak gerçek dünyada çalışan bir uçtan uca otomatik makine öğrenmesi çözümü, büyük veri ortamında ölçeklenebilir olmayı, kolay kurulumu, bakım ve güncellemelerin yapılabildiği yönetim bağlantılarını da gerektirmektedir. Bu problemi gören araştırmacılar tarafından oluşturulan çalışma grupları ile 10 katmanlı bir çözüm yaklaşımı sunulmuştur. Daha önce bu çalışmaların toparlandığı Amerikan Ulusal Bilim ve Teknoloji Enstitüsü (NIST) tarafından da sunulan bu çalışma [17] OptiScorer için yeniden yorumlanmış ve aşağıdaki katmanlar kodlanarak motorun çözüm süreçlerine eklenmiştir.

En alt katmanda veri kaynakları ve bu veri kaynaklarına bağlanan çözüm teknolojilerine ihtiyaç vardır. Bağlantı katmanındaki en büyük sorunlarında birisi de büyük veri ortamında akan verinin kuyruk yapısı içerisinde işlenmesidir. Bu amaca yönelik Apache Kafka veya Rabbit MQ gibi teknolojilerin çözüme entegre edilmesi gerekmiştir.

Veri bağlantı katmanının üzerinde farklı veri kaynaklarından gelen verinin entegrasyon problemi bulunmaktadır. Veri tekilleştirme ve verinin tek bir işlenebilir büyük tablo haline getirilmesi olarak görülebilecek bu problem, yine dağıtık ortamda çalıştırılmakta ve verinin yerelleştirilmesi (localization) problemi çözülmektedir.

Büyük veri katmanı tam olarak bu aşamada devreye girmekte ve yatay ölçekleme gereği olarak çok sayıda farklı sunucu üzerinde veri ve problemler dağıtılmaktadır. Bu dağıtım bir üst katmanda bulunan map-reduce fonksiyonlarının sunucular üzerinde dağıtılmasını gerekli kılmakta bu yüzden otomatik makine öğrenme çözümlerinde sunulan algoritmaların dağıtılabilir olması şartı da beraberinde gelmektedir. Farklı paralelleştirme teknolojilerinin zorlandığı bu aşamada, veri replikasyonu, algoritma paralelleştirmesi veya veri lokalizasyonu çözümleri kullanılabilmektedir.

Algoritma soyutlama katmanı genelde veri katmanı ile algoritma katmanını bir birinden ayırmak için kullanılır. Bu katmanı basitçe teknoloji bağımsız bir yapay zeka motoru oluşturma kaygısına yönelik olarak geliştirdiğimizi söyleyebiliriz. Bu sayede verinin tutulduğu teknolojilerden bağımsız olarak paralelleştirme ve yatay ölçekleme mümkün hale getirilecektir. Örneğin Hadoop veya Spark üzerinde yapılan bir parallelleştirme yapay zeka motoru OptiScorer için artık önemsiz hale gelmekte ve problemin soyutlama katmanında yapay zeka çözümü ile tekonoloji arasındaki eşleştirme ile çözülmesi mümkün olmaktadır.

Genelde üretilen çözümler daha üst seviyelerde farklı hizmetlere veya mikro servislere bağlanabilmektedir. Örneğin muhasebe departmanı ve insan kaynakları departmanlarının yapay zeka motorundan beklentileri farklı olabilmektedir. Bütün departmanların farklı çözüm taleplerine aynı motor, aynı veri kaynakları üzerinden bu sayede çözüm sunabilmekte ve servis katmanında temsili durum transferi (representation state trasnfer, REST) veya uygulama programlamabilir arayüzü (application programmable interface, API) gibi bağlantı protokolleri tanımlanabilmektedir.

Elbette bütün bu katmanların paralelinde her katman için ayrı ayrı güncelleme, teknoloji değişimi ve bakım prosedürlerinin de yaşayan sistemler için tasarımları yapılarak uygulamaya alınması gerekmiş ve bu konularda farklı geliştirme ve canlıya alma stratejileri iş geliştirme süreçlerinin parçası olarak hayata geçirilmiştir.

6. Otomatik Makine Öğrenmesinin Kullanım Alanları

Otomatik makine öğrenmesi farklı açılardan avantaj sağlamaktadır. Öncelikle, veri bilimi projelerinin başarısızı olma sebepleri incelendiğinde, aşağıdaki sebepler, sırasıyla en büyük proje başarısızlık sebebi olarak görülmektedir. Her ne kadar otomatik makine öğrenmesi günümüzde OptiSegment veya OptiMatcher gibi yapay zeka motorları üzerinden gözetimsiz öğrenme gibi alanlarda da mümkün hale gelmiş olsa da skorlama özelinde aşağıdaki kullanım alanlarını ele almak mümkündür.

Müşteri Skorlama: müşterilerin veya paydaşların hizmet veya ürün ile ilgili görüşlerini skorlamak için kullanılan bu özellikle, müşteri davranışları ve toplanan veriler üzerinden müşterinin memnuniyetinin skorlanması ve tahmin edilmesi mümkündür.

Müşteri Kayıp Skorlama (Customer Churn Scoring): yeni bir müşteri kazanmanın, eldeki mevcut müşterileri tutmaya göre daha maliyetli olduğu düşünülürse müşterilerin elde tutulması (retention) oldukça kritik bir süreçtir. Müşteri tutundurmanın ilk adımı ise, bırakma ihtimali yüksek müşterilerin tespit edilerek uygun aksiyonların alınmasını kapsar. Bu açıdan müşteri kayıp ihtimalinin skorlanması için otomatik makine öğrenmesi bazlı kullanım oldukça önemlidir.

Yönlendirme Skorlama (Lead Scoring): Potansiyel müşterilerin skorlanması ve farklı kazanım (on boarding) süreçleri için skorlanarak en uygun yaklaşma yöntemi ile müşterinin kazanılması amacıyla skorlama algoritmalarından faydalanılabilir.

Periyodik Satış İhtimalinin Skorlanması (Sales Frequency): OptiScorer, tekrar alış veriş yapan müşterilerin alış veriş yaptıkları ürünlerin tekrarlanması durumunda, her ürün için tekrarlanan satış frekanslarını çıkarabilmekte ve bu sayede tekrar satış frekansı gelen müşteriler için uyarı yapabilmektedir. Örneğin emlak sektöründe kiraların yıllık yapılması her yıl aynı dönemlerde kiracının tekrar ev arama ihtimalini arttırmakta, sigorta poliçeleri veya kredi kartlarının yıllık satılması benzer durumları ortaya çıkarmaktadır. Takvim frekansları nispeten bilinen bu ürünlere nazaran satılan her ürünün de satış frekansını OptiScorer ile bulmak ve müşterilere doğru zamanda benzer tavsiyelerde bulunmak mümkün hale gelmektedir.

Müşteri Yaşam Değeri (Customer Lifetime Value, LTV): OptiScorer, müşterilerin yapacağı toplam alış verişin değerini önceden tahmin edebilmektedir. Örneğin parekende sektöründe, bir müşterinin yapacağı toplam alım miktarı yaptığı alış veriş sıklığına ve alışveriş büyüklüğüne bağlıdır. LTV değeri, bir müşterinin kazandırabileceği tahmini değeri verdiği için, bir müşteriye yapılabilecek harcama (müşteri kazanma harcaması) çıkarılabilmekte ve buna göre reklam, kampanya, indirim gibi hesaplar yapılabilmektedir. Benzer şekilde LTV değerine ulaşmadan olan müşteri kayıplarının sebebi de incelenebilmekte ve gerekli durumlarda müdahale edilebilmektedir.

Müşteri Yaşam Süresi (Customer Life time): Müşterinin tahmini sistemde kalma süresini hesaplamakta ve hem müşterinin sistemde geçireceği sürenin en verimli şekilde değerlendirilmesi hem müşteri deneyiminin arttırılması hem de müşterinin daha yüksek değere dönüşmesi açısından önem taşımaktadır.

Ödeme Riski Skoru : Borçların tahsili açısından müşterilerin ödeme ihtimalleri ve risklerinin skorlanması, yine OptiScorer tarafından yapılabilmektedir. Buradaki kritik nokta, müşterinin ödeme ihtimali yanında OptiScorer'ın, ödemeyi etkileyen faktörleri de bulması (açıklanabilir yapay zeka) ve her müşteri için ödeme veya ödememesine sebep olacak en önemli faktörleri çıkarabilmesidir.

Potansiyel Müşteri Skorlama: Potansiyel müşteriler arasından hangisinin müşteri olma ihtimalinin yüksek olduğunu ve hatta hangi ürünler için yüksek potansiyel içerdiğini skorlamaya yarar ve OptiScorer, diğer kullanım senaryolarında olduğu gibi, burada da açıklanabilir sonuçlar sunar.

Finansal ve Satış Anomalileri: Firmaların satış kanallarında veya finansal akışlarında olan anomalilerin bulunması için kullanılır. Buna göre OptiScorer belirli periyotlar için tahminleme yapar ve firmanın satış değerleri veya finansal akışı belirli bir güven koridorunda kaldığı sürece herhangi bir alarm vermez. Ancak bu güven koridoru aşağıya veya yukarıya doğru kırıldığında alarm vererek uyarı yapabilir ve hatta daha önceden tanımlı aksiyonlar bulunuyorsa, bu aksiyonları da icra edebilir.

Rakip İndeks Analizi: OptiScorer bir skorlama motoru olmakla birlikte, birden fazla firmanın beraber yer aldığı durumlarda, her firmayı ayrı ayrı skorlayarak bu skorlardan oluşan indeks oluşturması ve bu firmaların birbirine görece olarak sıralamasını (indeks) sunması mümkündür. Bu indeksleri düzenli olarak hazırlayarak firmaların zaman içerisindeki ilerleme ve gerileme sebeplerini analiz edebilir. Skor indeksleri sadece firmalar için değil, aynı zamanda siyasetten sivil toplum kuruluşlarına, devlet kurumlarından bireylere kadar geniş bir kullanım yelpazesine sahiptir.

OptiScorer rakip analizi ve firmanın fiyat bilgilerini içeren veri kümelerini tarayarak fiyat ve satış arasındaki dengeyi tahminleyebilir. Örneğin bir ürünün hemen satılmak istenmesi durumunda satılabileceği azami fiyatı tahmin etmek veya satış fiyatı verildiği durumlarda satılabileceği süreyi tahmin etmek OptiScorer'ın skorlama amaçları arasında yer almaktadır.

Yukarıdaki örneklerin de gösterdiği üzere, sektörde çok farklı alanlarda OptiScorer, skorlama yapay zeka motorunun kullanımı mümkündür. Bu kullanım veri kümesinin istenen senaryoya göre hazırlanması ve motora yüklenmesi ile sağlanabilir.

7. Otomatik Makine Öğrenmesi ve SaaS hizmetlerin Avantajları

Klasik makine öğrenme süreçleri ile çıkarılan modellerin kullanımı, otomatik makine öğrenmesine göre daha az tercih edilmesinin sebebi ise, aşağıdaki şekilde sıralanabilir:

1. Üretilen modelin eskiyor olması ve ilk başlarda elde edilen başarının zaman içerisinde düşüyor olması. Bu sorunun en büyük sebebi, zaman içerisinde değişen ortam koşulları ve veriler üzerinde oluşan farklılıklardır. Ayrıca zaman içerisinde biriken verinin ilerleyen zamanlarda daha başarılı modeller çıkarması mümkün hale gelecektir. Çözüm olarak elle üretilen modelin belirli aralıklarla tekrar çalışılarak, veri kaynağının yeniden taranması, üretilen modelin belki tamamen değiştirilmesi veya parametrelerinin düzenlenmesi gerekmektedir. Bunun yerine bütün bu süreci otomatik olarak yapan bir çözüm çok daha hızlı ve sürekli çalışarak değişiklikleri takip edebilir ve güncellemeler yapabilir.

2. Verinin mahremiyeti ve güvenliği. Bu problemin en büyük sebebi veri bilimi çalışmalarının elle yapılması durumunda verinin görülmesi gerektiği ve veri üzerinde madencilik çalışması yapan kişinin veri ile ilgili bilgilere erişiyor olmasıdır. Sorun sadece veri üzerindeki, isim, telefon, adres gibi mahrem bilgiler değil aynı zamanda veri üzerinden çıkarılabilecek gözle görülmeyen bağlantıların da çalışan araştırmacı tarafından görülebiliyor olmasıdır. Örneğin twitter verisi üzerinde çalışan bir araştırmacı, atılan tweet noktalarını birleştirerek bir coğrafyadaki hareketlilik ve yoğun noktalar hakkında bilgi sahibi olabilir, bu bilgi üzerinden çok farklı çalışmalar yürütebilir. Bu problemin çözümü olarak veri bilimi çalışmasının elle yapılması yerine otomatik makine öğrenmesi ile yapılması çok daha doğru bir yaklaşım olacaktır.

3. Tek bir model ile bütün sistemin çalıştırılmasının zor olması: Genelde bütün sistemi tek bir modelin çalışması ya çok zor, ya başarısı düşük ya da performansı düşüktür. Bu problemi çözümü için mesela müşlteri segmentasyonu gibi yöntemlere başvurulabilir. Segmentasyonun kullanıldığı durumlarda her segment için ayrı bir modelin çalıştırılması, yeniden veri ön işleme aşamasında verilerin çalışılması ve analiz edilmesi, sonuca etkisi olan verilerin bulunarak çalışılması gerekmektedir. Her segment için etkili olan veri farklı olabileceği gibi her segmentte kullanılan model de farklı olabilir. Çözüm olarak bütün sürecin her segment için elle yapılması yerine otomatik olarak yapılması önerilebilir.

4. Veri bilimi konusunda uzman bulmanın zor olması: Sektörde her firmada onlarca veri bilimine konu olabilecek problem bulunuyorken bu problemlerin tamamı ile uğraşacak uzman bulmak ve bu uzmanların değerli vakitlerini aynı problemlerin tekrarı şeklindeki uygulamalarla harcamak verimli görülmemektedir. Çözüm olarak problemlerin toparlandığı bir araç üzerinden kullanım senaryolarını bilen ve iş analizi yapabilen kişiler tarafından problemlerin araca aktarılması ve çözüm sunulmasının daha verimli olacağı anlaşılmaktadır.

5. Üretilen veri bilimi çözümlerinin standartlarının olmaması. Bireysel olarak sunulan veri bilimi çözümlerinin çözümü sunan kişinin yetenekleri ile sınırlı olması ve elle yapılan bu veri bilimi çözümlerinin sunulabilecek en iyi çözüm olduğunun bir garantisi olmaması, sektörde de farklı problemlere yol açmaktadır. Örneğin bir veri bilimi uzmanı, bir problem için çözüm ortaya atmakta ve bu atılan çözümü işletmelere hayata geçirerek kullanmaktadır. Ancak kimse, bu çözümden daha iyi bir çözüm olup olmadığını bilmemektedir. 2. Bölümde de bahsi geçtiği üzere onbinlerce farklı alternatif arasından deneme

yapmak, çoğu kişi ve proje için mümkün olamamaktadır ancak bu durum her zaman için bazı daha iyi çözüm alternatiflerini kaçırma riskini de beraberinde getirmektedir.

6. Üst yönetimlerin veri bilimi konusunda yeterli kültürde olmaması: Çoğu işletme ve organizasyonda, yeni kavram ve kültürel dönüşümlerin, işletmenin bütün seviyelerine yansıması yeterince hızlı olamamaktadır. Veri bilimi gibi görece olarak yeni kavramların üst yönetimlerin yoğun gündeminde anlaşılması ve organizasyonun dönüşümü gibi stratejik kararların hızlı alınması çoğu zaman mümkün olamamakta veya öncelik sırasında gerilerde kalmaktadır. Problemin en büyük sebeplerinden birisi de bu tip dönüşüm denemelerinin yüksek arge ve personel/teknoloji yatırımları gerektirmesidir. OptiScorer gibi SaaS üzerinden çözüm sunan motorların hızlı şekilde veriler üzerinde ön bazı sonuçlar göstermesi ve hatta işletmenin bütün veri bilimi dönüşümünü hızlıca sağlaması bu açıdan önemlidir ve üst yönetimlerin kültürel dönüşümlerini hızlandırmaktadır.

7. Algoritma güncellemeleri: Veri bilimi alanının dayandığı istatistik ve makine öğrenmesi çalışmalarında her yıl onlarca yeni algoritma bulunmakta ve bu algoritmaların sektörde kullanım alternatifleri denenmektedir. On binlerce farklı işletmenin her birinde ayrıca geliştirilen veri bilimi çözümlerinde, yeni çıkan algoritmaların güncellenmesi ve daha başarılı bulunan algoritmalara geçiş, hem süreç açısından zamansal maliyetler hem de insan kaynağı açısından sayısız zorluklar oluşturmaktadır. Çözüm olarak merkezi bir teknoloji kullanımı ve merkezde yapılan tek bir güncelleme ile veri bilimi projelerinde yer alan çok sayıda işeltmenin yeni algoritmalara geçmesi mümkün hale gelmektedir. Ayrıca geçiş, tek bir algoritmanın zorunlu geçişi şeklinde değil, otomatik makine öğrenme süreçlerinin içerisinde yeni algoritmaların denenmesi (şekil 2'deki algoritma kütüphanesine yeni algoritmaların eklenmesi) ve bu sayede sadece gerekli olan durumlarda ve gerçekten başarı sağlayan durumlarda yeni algoritmaların canlıya alınması ve çalıştırılması şeklinde yapılmaktadır. Son yıllarda SaaS üzerinden sunulan bu hizmetlerin yapay zeka seviyesine indirilmesi ile ortaya çıkan MLaaS (Machine Learning as a Service, bir servis olarak makine öğrenmes) [15] veya AIaaS (Artificial Intelligence as a Service, bir servis olarak yapay zeka) [16] yaklaşımları bu anlamda OptiScorer gibi 3. nesil yapay zeka motorlarının da sağladığı bir hizmet olarak görülebilir.

8. Paralel olarak veri bilimi çözümlerinin sunulması: Örneğin otellerin gecelik oda fiyatlarının hesaplandığı bir uygulamada, her otelin kendi veri dünyasında bir model geliştirilmesi gerekecektir. Bir otelde geçen sene kaç çocuğun konakladığı, diğer bir otelde, o sene şehirde kaç konferans, sergi veya konser olduğu etkili olacaktır. Veri kaynaklarının bile farklı olduğu böyle bir dünyada, her otel için çalışacak olan modelin farklı olabileceği anlaşılmaktadır. On binlerce farklı otelin olduğu bir ekosistemde, her otelin ayrı ayrı veri bilimci bulundurması, her otel için sonucu etkileyen verilerin ayrı ayrı çalışılması ve her otel için ayrı bir modelin sistemde kurgulanması oldukça güç olacaktır. Çözüm olarak, merkezi ve otomatik makine öğrenmesi çözümleri ile her otel için farklı çözümler üretilebilmekte ve bütün bu üretim süreci paralel çalışan onlarca bilgisayar sayesinde saatler mertebesinde sağlanabilmektedir.

8. Sonuç

Bu makalede, yapay zeka konusunda yaşanan 3. nesil dönüşüm ve bu dönüşümün bir parçası olarak otomatik makine öğrenmesi ve açıklanabilir yapay zeka çalışmalarına yer verilmiş, bu çalışmaların sahadaki uygulama örneklerine değinilmiş ve uçtan uca bir örnek olarak OptiScorer yapya zeka motorunun iç tasarımı açıklanmıştır. Kavramlara giriş yapılan bu makalede, ayrıca otomatik makine öğrenmesi ve açıklanabilir yapay zeka alanında yaşanan problemlere de değinilmiştir. Çok sayıdaki gerçek uygulamada bu yaklaşımlar kullanılır hale gelmiş ve dönüşüm süreçleri başlamıştır. 3. nesil yaklaşımın sağladığı en önemli katkılardan birisi, yapay zeka çözümlerinin herkes için ulaşılabilir hale getirilmesi ve servis olarak bu çözümlerden iş analitiği seviyesinde faydalanılabilmesidir.

Kaynakça

[1] Agrapetidou, Anna, et al. "An AutoML application to forecasting bank failures." Applied Economics Letters (2020): 1-5.

[2] Stamoulis, Dimitrios. Hardware-Aware AutoML for Efficient Deep Learning Applications. Diss. Carnegie Mellon University, 2020.

[3] Barreiro, Enrique, et al. "Net-Net auto machine learning (AutoML) prediction of complex ecosystems." Scientific reports8.1 (2018): 1-9.

[4] Bhat, Gautam Shreedhar, Nikhil Shankar, and Issa MS Panahi. "Automated machine learning based speech classification for hearing aid applications and its real-time implementation on smartphone." 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2020.

[5] He, Xin, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A Survey of the State-of-the-Art." arXiv preprint arXiv:1908.00709(2019).

[6] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016.

[7] Yıldız, Merve, and Şadi Evren ŞEKER. "Veri Madenciliği Araçları (Data Mining Tools)." YBS Ansiklopedi 3.4 (2016).

[8] Hall, Patrick, Megan Kurka, and Angela Bartz. "Using H2O driverless ai." (2017).

[9] Kanter, James Max, and Kalyan Veeramachaneni. "Deep feature synthesis: Towards automating data science endeavors." 2015 IEEE international conference on data science and advanced analytics (DSAA). IEEE, 2015.

[10] Thornton, Chris, et al. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.

[11] Seker, Sadi Evren, and Khaled Al-Naami. "Sentimental analysis on Turkish blogs via ensemble classifier." Proc. the 2013 International Conference on Data Mining. 2013.

[12] Seker, Sadi Evren, et al. "A novel string distance function based on most frequent K characters." arXiv preprint arXiv:1401.6596 (2014).

[13] Seker, Sadi Evren, and Ibrahim Ocak. "Performance prediction of roadheaders using ensemble machine learning techniques." Neural Computing and Applications 31.4 (2019): 1103-1116

[14] Seker, Sadi Evren, and Banu Diri. "TimeML and Turkish Temporal Logic." IC-AI. Vol. 10. 2010.

[15] Ribeiro, Mauro, Katarina Grolinger, and Miriam AM Capretz. "Mlaas: Machine learning as a service." 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015.

[16] JEFIMOVA, ZENJA, and SOFIE NABSETH. "A Pricing Model for AIaaS: An analysis of a new AI personalization product within the edtech space." (2018).

[17] Grady, Nancy W., et al. "Big data: Challenges, practices and technologies: NIST big data public working group workshop at IEEE big data 2014." 2014 IEEE International Conference on Big Data (Big Data). IEEE, 2014.