MAKİNE ÖĞRENİMİNİN ARAŞTIRMACILARIN VERİ ANALİZİ BAĞLAMINDA POTANSİYEL ÖNEMİ *

Hasan T. Aytekin**

ÖZ

Bu çalışma, birçok farklı alandaki değişik konularda çalışan uygulamalı araştırmacılar için Makine Öğrenimi hakkında bilgi sağlamayı amaçlamaktadır. Örneğin, ekonomi araştırmacıları tarafından kullanılacak bu tür ham verilerin en yaygın kaynaklarından biri, geliştirme türü verilerdir. Bu tür verilerin en yaygın kaynakları, ilgili kurumlar tarafından ücretsiz ve çevrimiçi olarak sağlanan OECD ve Dünya Bankası veri setleridir. Küresel kurumlar tarafından sağlanan bu tür veri kümeleri ile ilgilenen akademik araştırmacılar, kendi araştırma projelerinde kullanmak için kendi veri kümelerini oluşturmak amacıyla makine öğrenimi tekniklerinin nasıl yardımcı olabileceğini, oluşturdukları kendi veri kümelerinin makine öğreniminde nasıl kullanılabileceğini ve bu veri kümelerini makine öğrenimi teknikleriyle analiz etme konusundaki bilgilerini derinleştirebileceklerdir. Bu amaçla, Dünya Bankası Açık Veri ortamında çevrimiçi olarak sunulan Dünya Gelişim Göstergesi zaman serisi verileri kullanılarak çok değişkenli bir tahmin problemini çözmek için makine öğrenimi teknikleri ile örnek bir vaka geliştirilecektir. Çoğunlukla Ridge, Lasso, Elastic-Net ve LARS gibi doğrusal tekniklere ve yüksek boyutlu verileri işlemek için çok uygun olan diğer bazı yöntemlere odaklanılacaktır. Bu örnek vakada, ilk olarak veriler incelenecek (eksik verilerle başa çıkma ve eksik veri değerlerini değiştirme dahil) ve makine öğrenimi modellerinin eğitimi için kullanılacak veriler hazırlanacaktır. Daha sonra kullanılacak tahmin modellerine karar verilecek ve son olarak bu modelleri değerlendirip elde edilen sonuçlar tartışılacaktır. Bu kapsamda, Makine Öğrenimini kullanan Zaman Serisi Tahmin örneği, Python ortamı kullanılarak sunulacak ve örnek vakanın Jüpiter Not Defteri de Anaconda Cloud ortamında paylaşılacaktır.

Anahtar Sözcükler: Makine Öğrenimi, Veri Analizi, Zaman Serisi Verileri, Çok Değişkenli Tahmin, Düzenlilik

---

* Bu makale, 13.02.2021 tarihinde, Ufuk Üniversitesi, 1. Uluslararası Sosyal Bilimler Kongresi'nde sunulan bildirinin genişletilmiş halidir.

** Öğretim Görevlisi (Yarı-zamanlı), Doktora Öğrencisi, Ufuk Üniversitesi, İ.İ.B.F., Yönetim Bilişim Sistemleri Bölümü, hasan.aytekin@ufuk.edu.tr

Makale Gönderim Tarihi: 08.04.2021
Makale Kabul Tarihi: 15.06.2021

THE POTENTIAL IMPORTANCE OF MACHINE LEARNING IN THE CONTEXT OF RESEARCHERS DATA ANALYSIS

ABSTRACT

This article aims to provide insights on Machine Learning for applied researchers working on topics related to any field. One of the most common sources of such raw data to be used by economic researchers are the development kind of data. The most common sources of such data are OECD and World Bank data sets which are provided by the respective institutions freely and online. The academic researchers in the related fields of such datasets provided by the global institutions may be interested in deepening their knowledge of how machine learning can be useful for the construction of valuable datasets to be used in their research projects and analyze these datasets by machine learning techniques. For this purpose, an example case using machine learning techniques to solve a multivariate forecasting problem will be developed by using World Development Indicator time-series data available online at World Bank Open Data environment. The focus will mainly be on linear regularization techniques such as Ridge, Lasso, Elastic Net, LARS, and some other methods that are well suited for handling high dimensional data. Within this example case, we will initially explore the data (including dealing with missing data and replacing missing data values) and prepare the data to be used for training the machine learning models. Then we will decide the predictive models to be used, and finally evaluate these models and discuss the results obtained. The example case of Time-Series Forecasting using Machine Learning will be presented by utilizing the Python environment and the Jupiter Notebook of the example case will also be shared at Anaconda Cloud environment.

Keywords: Machine Learning, Data Analysis, Time-Series Data, Multivariate Forecasting, Regularization.

1. GİRİŞ

Bu çalışmada, yapay zekanın bir altkümesi olan makine öğrenimi tekniklerinin araştırmalarda analiz amacı ile nasıl kullanılabileceği ile ilgili bilgiler tartışılacaktır. Bu amaçla önce yazılım, yapay zekâ ve makine öğreniminin nerede başlayıp nerede bittiği konusu ile aralarındaki ilişkiler irdelenecektir. Daha sonra tanımlayıcı bir biçimde makine öğreniminin ne olduğu ile birlikte makine öğrenimi yöntemlerinden denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenme ile ilgili bilgiler gözden geçirilecektir. Bu çalışmada, denetimli öğrenme teknikleri kullanılarak çok değişkenli bir tahmin probleminin çözümü için makine öğrenimi teknikleri ile problemi en iyi modelleyecek yöntem belirlenecektir. Bu amaçla kullanılacak temel analiz teknikleri ve bu tekniklerin temel taşları olan teoriler kısaca açıklanacaktır ve her bir modelin arka planda çözdüğü optimizasyon probleminin yapısına örnek olacak teorik bilgileri sunulacaktır.

Uygulama aşamasında ise Dünya Bankası Açık Veri ortamından alınan Dünya Gelişim Göstergesi zaman serisi veri setleri Exponential Smoothing, Doğrusal Regresyon, Ridge Regresyonu, Lasso Regresyonu, LARS Regresyonu, Elastic-Net Regresyonu, Zaman-Serisi LassoCV Regresyonu, Naive Predictör, Random Forest ve Gradyan Artırma tahmin modelleri kullanılarak denenecek ve elde edilen sonuçlar her bir model için tahmin hatalarının standart sapması ile değerlendirilecektir. Değerlendirme için Python programlama dili kullanılarak örnek bir program oluşturulmuştur ve Anaconda bulut ortamından Jupyter Notebook olarak paylaşılmıştır. Geliştirilen uygulama yazılımı dört ana sınıftan oluşmaktadır. Bu sınıflardan ikisi dünya bankasından verileri indirme ve işlemeye hazır hale getirmekle ilgilidir. Üçüncü sınıf, ham verilerin işlenerek makine öğreniminde kullanılabilmesi için gereken temizleme, ayıklama gibi veri seti üzerinde yapılan çalışmalardır. Dördüncü ve son sınıf ise işlenmeye hazır verileri kullanarak değişik modeller ile tahminler üretmek için hazırlanmış olan Python program betikleridir. Bu dört sınıf kullanılarak, her bir modelin tahmin performansları çıkarılacaktır.

Son olarak ise çalışmada elde edilen veriler kullanılarak üretilen modellere ait hataların standart sapmaları göz önüne alınarak analiz edilecek, ilgili modellerin eldeki verileri ne kadar temsil ettiklerini tartışılacaktır.

2. KURAMSAL ÇERÇEVE

2.1. Yazılım Perspektifi

Yazılım geliştirmeye başlanan ilk zamanlarda mantıksal kuralları ve koşulları belirleyerek programların akışının kontrol edilebilmek, anahtarları kullanabilmek, döngüler oluşturabilmek ve daha fazlası ile yazılım geliştirmeyi deneyimlemek, kişilerin disiplinlerinden bağımsız olarak bir yazılım geliştirme tutkusuna dönüşmektedir.

Daha sonraki yıllarda, modüller oluşturarak ve kod parçalarını işlevlere ve sınıflara soyutlayarak geliştirilen kişisel kodlardaki dağınıklıklar giderilmeye başlanmaktadır. Bir sonraki adımda ise yazılım geliştirme becerileri, nesne yönelimli analiz ve tasarım (OOA / D) ile daha ileri bir aşamaya taşınmaktadır. Kodun yeniden kullanımını ve değişik tasarım modelleri bu aşamada öğrenilmektedir. Geliştirilen programları Birleştirilmiş Modelleme Dili (UML - Unified Modeling Language) çizelgeleri ve diyagramlarında ifade etmeyi öğrenmek bundan sonra gelen adımdır. Sonrasında ise bu ilkeler çeşitli gereksinimleri karşılayabilmek amacı ile ortalama sekiz farklı programlama dilinde uygulanmaktadır.

Ancak programlamanın temel kuralı her zaman aynı kalmaktadır: Kuralları ve mantığı tanımlamak. Gerisi sadece bu kuralların uygulanmasını ve sürdürülmesini kolaylaştıran çeşitli yöntemler ve felsefelerdir.

Metodik programlamanın başladığı ilk günden beri, kural tabanlı kodlama, yazılım oluşturmanın tek yöntemi olmuştur. Bir veya bir dizi problem analiz edilip, sınırları, varlıkları, süreçleri ve ilişkileri belirlenmekte ve bunlar ise geliştirilecek yazılımın çalışma şeklini tanımlayan kurallara dönüştürülmektedir.

2.2. Yapay Zeka Perspektifi

Metodik programlama yöntemleri bugüne kadar çok işe yaramıştır ve bundan sonra da işe yaramaya devam edecektir. Ancak elde edilecek sonuç, her bir farklı yazılım projesinde kullanılan program mantığının bir şekilde güncellenmedikçe davranışlarını asla değiştirmeyecek olan "statik" bir yazılımdır. Bu tür sadece uygulanan mantık çerçevesinde statik olarak çalışan yazılım programlarıyla görüntülerdeki nesneleri tanımak, ağ trafiğinde kötü amaçlı bir etkinlik bulmak veya engebeli arazide bir robotu gezdirmek gibi kuralların kesin olmadığı senaryoları hayata geçirmek olası değildir.

Modern yapay zekanın temel taşı olan makine öğrenimi, geleneksel programlama modelini alt üst eden bir bilim alanıdır. Makine öğrenimi, herhangi bir uzmanın görevlerin nasıl yerine getirileceğini açıklamasına gerek kalmadan performansını değiştirebilen ve geliştirebilen yazılımlar oluşturmaya yardımcı olmaktadır. Web sitelerinde, dijital asistanlarda, sürücüsüz arabalarda, analiz yazılımlarında ve daha fazlasında görülen akıllı öneriler dahil olmak üzere, bugün doğrudan kullandığımız ve ufukta gördüğümüz pek çok yeniliğin arkasında yatan teknoloji yapay zeka teknolojisi olacaktır.

2.3. Makine Öğrenimi

Makine öğrenimi, örneklerden öğrenen bir yazılımdır ve algoritmaları klasik yazılım geliştirmede olduğu gibi kodlanmaz, ancak büyük miktarda ilgili veriler ile eğitilmektedir. Örneğin, bir kedinin bir makine öğrenimi algoritmasına nasıl göründüğünü açıklamaya çalışmak yerine, ilgili makine öğrenimi algoritmasına milyonlarca kedi resmi sağlanmalıdır. Algoritma, bu görüntülerde yinelenen kalıplar bularak, bir kedinin görünüşünün nasıl tanımlanacağını kendisi belirlemektedir. Daha sonra, programa yeni bir resim gösterildiğinde, bir kedi içerip içermediğini ilgili algoritma ayırt edebilmektedir.

Makine öğreniminin ilk tanımlamalarından birisi Arthur Samuel'in 1959 yılında yaptığı tanımlamadır. Bu tanımlamada Samuel "Makine Öğrenimi, deneyimlerden öğrenmek için bilgisayarları programlama veya bilgisayarlara açıkça programlanmadan öğrenme yeteneği veren çalışma alanıdır" demektedir (Samuel, 1959). Daha teknik bir mühendislik tanımını ise 1977 yılında Tom Mitchell yapmıştır: "Görevin T, performansın P ile gösterildiği durumda eğer P ile ölçülen T performansı E deneyimi ile iyileşiyorsa, ilgili programın E deneyiminden öğrendiği söylenir" (Mitchell, 1997). En yeni tanımlarından birisi ise 2019 yılında Géron (2019) tarafından "Makine Öğrenimi, bilgisayarların verilerden öğrenebilmeleri için geliştirilmiş olan programlama bilimidir ve aynı zamanda bir sanattır" olarak yapılmıştır.

Güncel pek çok yazıda makine öğrenimi yapay zeka ile özdeşleştirilmektedir. Ancak yapay zeka, karmaşık kural tabanlı yazılımlardan henüz icat edilmemiş insan düzeyindeki zekaya kadar her şeye uygulanabilen genel bir terimdir. Gerçekte, makine öğrenimi, kuralların aksine verilere dayalı programlar oluşturmakla ilgili özel bir yapay zeka alt kümesidir.

2.4. Makine Öğrenimi Yöntemleri

Birçok makine öğrenimi sistemi vardır ve bunlar 3 farklı kategoride sınıflandırılmaktadır (Géron, 2019). Bu kategoriler aşağıdaki şekilde sınıflandırılmaktadır:

1. İnsan gözetimi ile eğitilip eğitilmedikleri (denetimli, denetimsiz, yarı denetimsiz ve pekiştirmeli öğrenme).
2. Anında aşamalı olarak öğrenip öğrenemeyecekleri (çevrimiçi öğrenmeye karşılık gelen toplu öğrenme).
3. Veri noktalarını bilinen veri noktalarıyla basitçe karşılaştırarak veya eğitim verilerindeki kalıpları tespit edip, tıpkı bilim adamlarının yaptığı gibi (örnek tabanlı ve model tabanlı öğrenme) bir tahmine dayalı model oluşturarak çalışıyor olması.

Bu çalışmada makine öğreniminin birinci kategori yöntemleri kullanılmaktadır. Bu nedenle denetimli, denetimsiz, yarı denetimsiz ve pekiştirmeli öğrenme yöntemleri kısaca gözden geçirilecektir.

2.5. Denetimli Öğrenme

Makine öğrenimi algoritmalarının birkaç çeşidi vardır. En yaygın olanlardan biri, algoritmanın etiketli verilerle eğitildiği ve bir dizi girdinin bir dizi çıktıyla eşleştirdiği *denetimli öğrenme*dir. Yüze takılan maskeyi algılamak, denetimli öğrenmeye bir örnektir. Başka bir örnek, algoritmaya ses dalga biçimleri ve bunlara karşılık gelen yazılı sözcüklerin sağlandığı konuşma tanımadır.

Denetimli bir öğrenme algoritmasına ne kadar çok örnek sağlanırsa, yeni verileri sınıflandırmada o kadar hassas olmaktadır. Denetimli öğrenmenin temel sorunu da burada yatmaktadır. Etiketli örnekler ile büyük veri kümelerini oluşturmak çok zaman almaktadır ve kapsamlı bir insan çabası gerekmektedir. Bu amaçla sadece veri etiketleme hizmeti sağlayan platformlar oluşmuştur. Bu platformların en çok bilineni Amazon'un *Mechanical Turk* platformudur.

2.6. Denetimsiz Öğrenme

Makine öğreniminin başka bir dalı olan *denetimsiz öğrenme*de referans verisi yoktur. Sadece girdi için kullanılacak veri gereklidir. Algoritma, etiketlenmemiş verileri alarak çıkarımlarda bulunur ve ilgili veri dizilerindeki kalıpları bulur. Denetimsiz öğrenme, özellikle insanların tanımlayamayacağı gizli kalıpların olduğu durumlarda kullanışlıdır.

Bilgisayar ağlarındaki etkinliklerin izlenmesi bu tür makine öğrenimi algoritmalarının bir örneğidir. Bu algoritmalar ağ etkinliklerindeki kalıpları gözlemleyerek normal ağ etkinliği için bir temel oluşturmakta ve elde edilen kalıpları baz alarak aykırı etkinlikleri belirleyerek uyarıda bulunmaktadır.

Denetimli öğrenmeye kıyasla, denetimsiz öğrenme, makinelerin kendi kendilerine öğrenebilmelerine bir adım daha yakındır. Ancak, denetimsiz öğrenmeyle ilgili sorun, sonucun genellikle tahmin edilemez olmasıdır. Bu nedenle, kendi kendine öğrenirken onu doğru yöne yönlendirmek için genellikle insan sezgisiyle birleştirilmektedir. Örneğin, yukarıda açıklanan ağ güvenliği örneğinde, ağ etkinliğinin herhangi bir kötü niyet olmadan normal kalıptan sapması için birçok neden vardır. Ancak bir makine öğrenimi algoritması, istisnaları öğrenene ve daha iyi kararlar alana kadar kararlarını bir insan analistin düzeltmesi gerekeceğini bilmemektedir.

2.6.1. Yarı Denetimli Öğrenme

Bazı algoritmalar, kısmen etiketlenmiş eğitim verileri (genellikle çok sayıda etiketlenmemiş veri ve biraz etiketlenmiş veri) ile çalışmaktadır. Buna yarı denetimli öğrenme denir. Google Fotoğraflar gibi bazı fotoğraf barındırma hizmetleri yarı denetimli öğrenme algoritmasının en güncel örneklerindendir.

Yarı denetimli öğrenme algoritmalarının çoğu, denetimsiz ve denetimli algoritmaların birleşimidir. En çok kullanılan örnek, derin inanç ağlarıdır ve birbiri üzerine yığılmış kısıtlı Boltzmann makineleri adı verilen denetimsiz bileşenlere dayanır. Kısıtlı Boltzmann makineleri denetimsiz bir şekilde sırayla eğitilir ve ardından tüm sisteme denetimli öğrenme teknikleri kullanılarak ince ayar yapılır.

2.6.2. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme, toplam ödülü maksimize etmek için akıllı aracıların bir ortamda nasıl harekete geçmesi gerektiğiyle ilgili bir makine öğrenimi alanıdır (Hu, Niu, Carrasco, Lennox, & Arvin, 2020). Pekiştirmeli öğrenme, denetimli öğrenim ve denetimsiz öğrenmenin yanı sıra üç temel makine öğrenimi paradigmasından biridir.

Pekiştirmeli öğrenme, etiketli girdi/çıktı çiftlerinin sunulması gerekmemesi ve açıkça düzeltilmesi için optimale yakın olan eylemlerin gerekmemesi bakımından denetimli öğrenmeden farklıdır. Bunun yerine odak, keşif (keşfedilmemiş bölgenin) ve sömürü (mevcut bilginin) arasında bir denge bulmaktır (Hu, Niu, Carrasco, Lennox, & Arvin, 2020).

Problemi çözmeye odaklanmış bir temsilci (yapay zeka), belirsiz, potansiyel olarak karmaşık bir ortamda bir hedefe ulaşmayı öğrenmektedir. Temsilci olarak davranan yapay zeka, oyun benzeri bir durumla karşı karşıyadır ve soruna bir çözüm bulmak için deneme yanılma yöntemini kullanmaktadır. Yapay zeka, geliştirilen çözümün programcının istediğini yapmasını sağlamak amacıyla, gerçekleştirilen eylemler için ya ödül ya da ceza alır. Amacı, toplam ödülü maksimize etmektir.

Tasarımcı, ödül politikasını, yani oyunun kurallarını belirlese de, modele oyunun nasıl çözüleceğine dair hiçbir ipucu veya öneri vermemektedir. Tamamen rastgele denemelerden başlayıp gelişmiş taktikler ve insanüstü becerilerle oynadığı oyunu bitirerek ödülü en üst düzeye çıkarmak için görevin nasıl yerine getireceğini anlamak modelin görevidir. Aramanın gücünden ve birçok denemeden yararlanarak, pekiştirmeli öğrenme şu anda makinenin yaratıcılığına ipucu vermenin en etkili yoludur. İnsanların aksine, yeterince güçlü bir bilgisayar altyapısında bir pekiştirmeli öğrenme algoritması çalıştırılırsa, yapay zekâ binlerce paralel oyundan deneyim toplayabilmektedir.

Otonom arabaları kontrol eden modelleri eğitmek, pekiştirmeli öğrenmenin potansiyel bir uygulamasına mükemmel bir örnektir. İdeal bir durumda bilgisayar, arabayı sürmekle ilgili hiçbir talimat almamalıdır. Programcı, görevle bağlantılı herhangi bir şeyi programa kodlamamalı ve makinenin kendi hatalarından öğrenmesine izin vermelidir. İdeal bir durumda, programın içine fiziksel olarak kodlanacak tek unsur ödül fonksiyonu olmalıdır.

Normal olarak kullanılması düşünülen otonom araçla, yarış için kullanılabilecek bir otonom araç arasında birtakım farklılıklar olacaktır. Örneğin, olağan koşullarda güvenliği ön planda tutacak, sürüş süresini en aza indirecek, kirliliği azaltacak, yolculara konfor sunacak ve hukuk kurallarına uyacak otonom bir araca ihtiyaç duyulmaktadır. Otonom bir yarış arabasında ise, sürücünün konforundan çok hıza önem verilmektedir. Programcı, yolda karşılaşabileceği her olayı tahmin edemez ve uzun "eğer öyleyse" talimatları oluşturmak yerine, pekiştirmeli öğrenme aracısını ödül ve cezalar sisteminden öğrenebilecek şekilde hazırlamaktadır. Temsilci

(görevi yerine getiren pekiştirmeli öğrenme algoritmalarının başka bir adı) belirli hedeflere ulaşmak için ödüller almakta ve verilen görevi maksimum ödül ile yerine getirmeye çalışmaktadır.

3. YÖNTEM

Bu çalışmada, denetimli öğrenme teknikleri kullanılarak çok değişkenli bir tahmin probleminin çözümü için makine öğrenimi teknikleri ile problemi en iyi modelleyecek yöntem belirlenecektir. Araştırmada kullanılacak her bir model, tahmin hatalarının standart sapması ile değerlendirilecektir [Kök Ortalama Kare Hatası (RMSE)].

Örnek araştırmada Dünya Bankası Açık Veri ortamından alınan Dünya Gelişim Göstergesi zaman serisi veri setleri kullanılacaktır. Dünya Bankası Dünya Gelişim Göstergesi zaman serisi verileri çok değişkenli bir tahmin problemi verisi olarak kullanılarak aşağıdaki tahmin modelleri ile denenecek ve ilgili seriyi en iyi tahmin eden model seçilecektir:

- Exponential Smoothing (Holt)
- Doğrusal Regresyon
- Ridge Regresyonu
- Lasso Regresyonu
- LARS Regresyonu
- Elastic-Net Regresyonu
- Zaman-Serisi LassoCV Regresyonu
- Naive Predictör
- Random Forest
- Gradyan Artırma (Gradient Boosting)

Araştırmada ağırlıklı olarak değişik regresyon modelleri kullanılmıştır. Bu modeller, standart modeli daha basit hale getirmek ve aşırı uyum (overfitting) riskini azaltmak için değişik kriterlerle sınırlandırılarak düzenlileştirilmiş regresyon modelleri olarak denemeye alınmıştır.

Düzenli hale getirmek, işleri düzenli veya kabul edilebilir kılmak anlamına gelmektedir. Matematik, istatistik, finans (Kratsios, 2020), bilgisayar bilimi, özellikle makine öğrenimi ve ters problemlerde, düzenlileştirme, yanlış bir problemi çözmek veya aşırı uyumu önlemek için bilgi ekleme sürecidir (Wikipedia, Regularization (mathematics), 2021). Bu yöntem, makine öğrenim algoritmasına verilen eğitim setine bir işlevi uygun şekilde yerleştirerek hatayı azaltmak ve aşırı uyumu önlemek için kullanılan tekniklerdendir.

Düzenlileştirme, kötü niyetli optimizasyon problemlerindeki amaç fonksiyonları yoluyla gerçekleştirilir. Düzenlileştirme terimi veya ceza, optimizasyon fonksiyonunun aşırı uyumlu olması veya optimal bir sonuç bulmak amacıyla ek bir maliyet olarak eklenmektedir (Neumaier, 1998).

Makine öğrenimi modelini eğitmenin en önemli yönlerinden biri, kullanılan modelin aşırı uyum göstermesinden kaçınmaktır. Aşırı uyum gözlemlemek, modelin doğruluğunun az olduğunu göstermektedir. Bunun nedeni, modelin eğitim veri kümesindeki gürültüyü yakalamak için çok uğraşmasıdır. Gürültü ile, verilerin gerçek özelliklerini kesin olarak temsil etmeyen, ancak rastgele tesadüfi olan veri noktaları kastedilmektedir. Bu tür veri noktalarını belirlemek, modelin aşırı uyumdan kaynaklanacak risklerden etkilenmesini önleyecek, daha esnek hale getirecek ve uygun uyumu yakalamasına olanak sağlayacaktır.

Bu aşamada düzenlileştirilmiş doğrusal modelleri kısaca gözden geçireceğiz.

3.1. Düzenlileştirilmiş Doğrusal Modeller

Aşırı uyumu azaltmanın iyi bir yolu, modeli düzenli hale getirmek, yani, sınırlamaktır. Model ne kadar az serbestlik derecesine sahip olursa, verilere aşırı uyması o kadar zor olacaktır. Örneğin, polinom derecelerinin sayısını azaltmak, bir polinom modelini düzenlemenin basit bir yoludur. (Géron, 2019)

Doğrusal bir model için, düzenlileştirme tipik olarak modelin ağırlıklarının sınırlandırılması yoluyla elde edilir. Aşağıda, ağırlıkları farklı yollarla sınırlandıran Ridge, Lasso, Elastik Net ve LARS regresyon modelleri sırasıyla incelenecektir. Bütün regresyon modellerinin temelinde yer alan doğrusal regresyon modeli, Ridge regresyon modelinin tanımı altında incelenecektir.

3.1.1. Ridge Regresyonu

3.1.1.1. Genel Sınıflandırma Özelliği

Ridge regresyonu L2 düzenliliği kullanır. L2 düzenliliği ağırlık (özellik ağırlıkları) girişlerinin (ceza terimleri) karelerinin en aza indirilmesidir (minimizasyon).

3.1.1.2. Tanım

Ridge regresyonu, verilerdeki ortak değişkenlerin çoklu bağlantı (multicollinearite) problemini çözmek için klasik bir veri modelleme yöntemidir (Hoerl & Kennard, 1970). Burada çoklu bağlantı, çoklu regresyon modelinde birden fazla öngörücü değişkenin yüksek oranda korelasyonlu olduğu bir durumu ifade etmektedir. Çoklu bağlantı mükemmel ise, regresyon katsayıları belirsizdir ve standart hataları sonsuzdur. Mükemmelden düşükse, regresyon katsayıları belirli olmakla birlikte büyük standart hatalara sahiptir ve bu da katsayıların büyük bir doğrulukla tahmin edilemeyeceği anlamına gelmektedir (Gujarati, N., & Madsen, 1998).

Çoklu bağlantı oluştuğunda, en küçük kareler tahminleri tarafsızdır, ancak varyansları büyük olduğundan gerçek değerden uzak olabilirler. Regresyon tahminlerine bir derece yanlılık ekleyerek, Ridge regresyonu standart hataları azaltılabilir (NCSS, 2020).

Ridge Regresyonu, Doğrusal (Lineer) Regresyonun özel bir hali olduğu için, bu aşamada Doğrusal Regresyonu da kısaca tanımlayacağız.

Doğrusal bir model, yalnızca girdi özelliklerinin ağırlıklı toplamını hesaplayarak ve bunun üzerine *Denklem 3.1.1.2.1*'de gösterildiği gibi eğilim terimi (kesme terimi olarak da adlandırılır) olarak adlandırılan bir sabit değer ekleyerek tahmin yapmaktadır.

Denklem 3.1.1.2.1 Doğrusal Regresyon model tahmini

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \qquad (3.1.1.2.1)$$

Denklem 3.1.1.2.1 vektörleştirilmiş bir form kullanılarak çok daha kısa bir şekilde yazılabilir.

Denklem 3.1.1.2.2 Doğrusal Regresyon model tahmini (vektörleştirilmiş yapı)

$$\hat{y} = h_\theta(x) = \boldsymbol{\theta} \cdot \boldsymbol{x} \qquad (3.1.1.2.2)$$

- $\boldsymbol{\theta}$ modelin parametre vektörüdür, sapma terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içerir.
- $\boldsymbol{x}$, x_0'ın her zaman 1'e eşit olduğu x_0 dan x_n'e kadar tanımlanan özellik vektörüdür.
- $\boldsymbol{\theta} \cdot \boldsymbol{x}$, $\boldsymbol{\theta}$ ve $\boldsymbol{x}$ vektörlerinin iç çarpımıdır ($\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$)
- $\boldsymbol{h_\theta}$, model parametreleri $\boldsymbol{\theta}$'yı kullanan hipotez fonksiyonudur.

Denklem 3.1.1.2.2 de kullanılan özellik vektörü $\boldsymbol{x}$ üzerine kurgulanan Doğrusal Regresyon hipotezi $\boldsymbol{h_\theta}$'nin hatalarının karelerinin ortalaması (MSE), *Denklem* 3.1.1.2.3'te verilmiştir:

Denklem 3.1.1.2.3 Doğrusal Regresyon modeli için hataların karelerinin ortalaması (MSE) maliyet fonksiyonu

$$MSE(\boldsymbol{\theta}) = MSE(\boldsymbol{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} - y^{(i)}\right)^2 \quad (3.1.1.2.3)$$

- **θ** modelin parametre vektörüdür, sapma (variance) terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içerir.
- $\boldsymbol{\theta}^T$, **θ** matrisinin transpose edilmiş halidir
- $\boldsymbol{X}$, tüm özellik değerlerini içeren bir matristir
- $\boldsymbol{x}^{(i)}$, veri kümesi içindeki i'inci örneğinin tüm özellik değerlerinin (etiket hariç) bir vektörüdür.
- $y^{(i)}$, veri kümesi içindeki i'inci örneğinin tüm özellik değerlerinin etiketidir (bu örnek için istenen çıktı değeri)

Ridge Regresyonu (Tikhonov regresyonu olarak da adlandırılır), Doğrusal Regresyonun düzenlenmiş bir versiyonudur: maliyet fonksiyonuna $\alpha\sum_{i=1}^{n}\theta_i^2$'ye eşit bir regresyon terimi eklenir (Géron, 2019). Hiperparametre α, modelin düzenli hale getirilmesinin derecesini kontrol eder. Eğer α = 0 ise Ridge Regresyonu sadece Doğrusal Regresyondur. Eğer α çok büyükse, tüm ağırlıklar sıfıra çok yakın olur ve sonuç, verilerin ortalamasından geçen düz bir çizgidir. *Denklem* 3.1.1.2.4, Ridge Regresyon maliyet fonksiyonunu temsil eder. (Géron, 2019)

Denklem 3.1.1.2.4 Ridge Regresyonu maliyet fonksiyonu

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + \alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2 \quad (3.1.1.2.4)$$

- **θ** modelin parametre vektörüdür, sapma terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içerir.
- α modelin düzenli hale getirilmesinin derecesini kontrol eder (Hiperparametre).

Ridge Regresyonu maliyet fonksiyonunda hatalarının karelerinin ortalamasına (MSE) eklenen ceza niteliğindeki regresyon terimi (düzenli hale getirme terimi), kullanım alanlarından birisi olan öğrenme algoritmasını yalnızca verilere uymaya değil, aynı zamanda model ağırlıklarını mümkün olduğunca küçük tutmaya zorlamaktadır.

Ridge regresyonu maliyet fonksiyonu (*Denklem* 3.1.1.2.4), katsayıların düzenli hale getirilmesi için kullanılan terimi en aza indirerek tahmin etmesi dışında en küçük karelere çok benzemektedir. Özellikle, Ridge regresyon katsayısı tahminleri θ_i^2, ayrı olarak belirlenecek olan α ≥ 0'ın bir ayar parametresi olduğu durumda minimize eden değerlerdir. *Denklem* 3.1.1.2.4 iki farklı ölçüte göre işlem yapar. En küçük karelerde olduğu gibi, Ridge regresyonu, $MSE(\boldsymbol{\theta})$'yi küçülterek verilere iyi uyan katsayı tahminleri arar. Bununla birlikte, büzülme cezası olarak adlandırılan ikinci terim $\alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2$, $[\theta_1, \dots, \theta_n]$ sıfıra yakın olduğunda çok küçüktür ve bu nedenle θ_i tahminlerini sıfıra doğru küçültme etkisine sahiptir. Ayarlama parametresi α, Ridge regresyonu maliyet fonksiyonundaki iki terimin regresyon katsayısı tahminleri üzerindeki nispi etkisini kontrol etmeye yarar. α = 0 olduğunda, ceza teriminin etkisi yoktur ve Ridge regresyonu en küçük kareler tahminlerini üretecektir. Bununla birlikte, $\alpha \to \infty$ olduğunda, büzülme cezasının etkisi büyür ve Ridge regresyon katsayısı tahminleri sıfıra yaklaşır. Yalnızca bir katsayı tahminleri kümesi oluşturan en küçük karelerin aksine, Ridge

regresyonu, her α değeri için farklı bir katsayı tahminleri kümesi, θ_i^2 üretecektir. α için iyi bir değer seçmek kritik öneme sahiptir. (James, Witten, Hastie, & Tibshirani, 2017)

Doğrusal Regresyonda olduğu gibi, kapalı formlu bir denklem hesaplayarak veya Gradyan İnişi gerçekleştirerek de Ridge Regresyonu gerçekleştirilebilir. Her ikisinin de artıları ve eksileri aynıdır. *Denklem* 3.1.1.2.*5*, kapalı form çözümünü gösterir (burada A, (n + 1) × (n + 1) birim matrisidir, bunun istinası ise sol üst hücredeki önyargı terimine karşılık gelen 0'dır).

Denklem 3.1.1.2.5 Ridge Regresyonu kapalı form çözümü

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{A})^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} \qquad (3.1.1.2.5)$$

- $\boldsymbol{\theta}$ modelin parametre vektörüdür, sapma terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içerir.
- $\boldsymbol{X}$, tüm özellik değerlerini içeren bir matristir
- $\boldsymbol{X}^T$, $\boldsymbol{X}$ matrisinin transpose edilmiş (ters çevirip yerini değiştirmek) halidir
- **A**, (n + 1) × (n + 1) birim matrisidir
- $\boldsymbol{y}$, hedef değerlerin vektörüdür

3.1.1.3. Kullanım Alanları

- Ortogonal Olmayan Problemler için Yanlı Tahmin. (Hoerl & Kennard, 1970)
- Çok boyutlu verilerle karşılaşıldığında ampirik tanımlanamazlığı ortadan kaldırmak için kullanılmasıdır.

3.1.1.4. Avantajları

- Parametreleri küçültmekte, bu nedenle çoğunlukla çoklu bağlantıları önlemek için kullanılmaktadır.
- Katsayı küçülme ile model karmaşıklığını azaltmaktadır.
- Hem kayıp fonksiyonları hem de düzenlileştirme seçenekleri vardır.
- Çok boyutlu verilerle karşılaşıldığında ampirik tanımlanamazlığı ortadan kaldırmaktadır.
- Bir ceza terimi eklemek aşırı uyumu azaltmaktadır.
- Ceza terimi, bir çözüm bulabileceğimizi garanti etmektedir.
- Bir modele fazla uymaktan kaçınmaktadır.
- Tarafsız tahmin ediciler gerektirmemektedir.
- Makul ölçüde güvenilir tahminler yapmak için veri topluluğu değerlerine yeterli önyargı değeri eklenmektedir.
- Tahmincilerin sayısı (p), gözlem sayısından (n) daha büyük olan çok değişkenli veri durumunda iyi performans göstermektedir.
- Ridge tahmincisi, çoklu bağlantı olduğunda en küçük kareler tahminini geliştirmede en iyi tercihtir.
- Çoklu bağlantı olduğunda varyansı düşürmek için önyargılı sonuçlar kullanılmaktadır.
- Eğitim setinden elde edilen tahmin uyumundan biraz daha kötü bir uyumla başlayarak daha iyi uzun vadeli tahminler sağlamaktadır.
- Ridge Regresyonu, çoğu değişken yararlı olduğunda en iyi sonucu vermektedir.
- Çok sayıda orta/büyük boyutlu etkiye sahip değişken varsa en etkili sonuç elde edilmektedir.

3.1.1.5. Dezavantajları

- Hesaplama açısından pahalıdır.
- Nihai modeldeki tüm öngörücüleri içermektedir.
- Özellik seçimi yapamamaktadır.
- Katsayıları sıfıra doğru küçültmektedir.
- Varyansı önyargı ile değiştirmektedir.
- Önyargıyı arttırmaktadır.
- En iyi α (hiper parametre) değerini seçme gerekliliği vardır.
- Model yorumlama yeteneği düşüktür.

3.1.2. En Az Mutlak Büzülme ve Seçim Operatörü - LASSO (Least Absolute Shrinkage and Selection Operator) Regresyonu

3.1.2.1. Genel Sınıflandırma Özellikleri

Lasso regresyonu, L1 düzenliliği kullanır. Amaç, maliyet fonksiyonunu (*Denklem 3.1.2.2.1*) düzenlilik terimi ($\sum_{i=1}^{n}|\theta_i|$)'ne göre en aza indirgemek olduğundan, ikinci dereceden (quadratic) optimizasyon problemidir (Tibshirani, 1996). Bu nedenle, her zaman benzersiz bir çözümü vardır.

3.1.2.2. Tanım

En Az Mutlak Büzülme ve Seçim Operatörü Regresyonu (kısaca Lasso Regresyonu olarak adlandırılır), Doğrusal Regresyonun başka bir düzenlenmiş versiyonudur. Tıpkı Ridge Regresyonu gibi, maliyet fonksiyonuna bir düzenlileştirme terimi ekler, ancak ağırlık vektörünün L2 normunun karesinin yarısı yerine L1 normunu kullanır (*Denklem 3.1.2*.2.1).

Denklem 3.1.2.2.1 Lasso Regresyonu maliyet fonksiyonu

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{n} |\theta_i|$$

- **θ** modelin parametre vektörüdür, sapma (variance) terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içerir.
- α modelin düzenli hale getirilmesinin derecesini kontrol eder (Hiperparametre).

Ridge regresyonunun bariz bir dezavantajı vardır. Nihai modeldeki tüm tahminleri içermektedir (James, Witten, Hastie, & Tibshirani, 2017). Buna karşın, Lasso Regresyonunun önemli bir özelliği ise, en az önemli özelliklerin ağırlıklarını tamamen ortadan kaldırma (yani, onları sıfıra ayarlama) eğiliminde olmasıdır. Diğer bir deyişle, Lasso Regresyonu özellik seçimini otomatik olarak gerçekleştirmekte ve sıfır olmayan birkaç özellik ağırlığına sahip olan seyrek (aralıklı) bir model ortaya çıkarmaktadır (Géron, 2019).

Ridge Regresyonunda olduğu gibi, *Denklem 3.1.2*.2.1 de kullanılan α, 0'dan pozitif sonsuza kadar herhangi bir değer olabilir ve Çapraz Doğrulama kullanılarak belirlenmektedir. Ridge ve Lasso Regresyonu arasındaki en büyük fark, Ridge Regresyonunun eğimi ($\sum_{i=1}^{n}\theta_i^2$) sadece 0'a yakın bir değere asimptotik olarak küçültebilmesi ancak Lasso Regresyonunun eğimi ($\sum_{i=1}^{n}|\theta_i|$) tamamen 0'a kadar küçültebilmesidir. Lasso Regresyonu, işe yaramaz değişkenleri denklemlerden hariç tutabildiğinden, çok fazla yararsız değişken içeren modellerde varyansı

azaltmada Ridge Regresyonundan biraz daha iyidir. Buna karşılık, Ridge Regresyonu, çoğu değişken yararlı olduğunda Lasso Regresyonundan daha iyi sonuç vermektedir.

3.1.2.3. Kullanım Alanları

- Lasso ve çeşitleri, sıkıştırılmış algılama alanı için temeldir. (Scikit-Learn, 2020)
- Otomatik olarak özellik seçimi yapmasından dolayı, özellik sayısının çok olduğu durumlarda kullanılmaktadır.

3.1.2.4. Avantajları

- İşe yaramayan değişkenleri denklemlerden hariç tutabildiğinden, çok fazla yararsız değişken içeren modellerde varyansı azaltmada Ridge Regresyonundan biraz daha iyidir.
- Lasso Regresyonu kullanılarak üretilen modelleri yorumlamak genellikle Ridge regresyonuyla üretilenlere göre çok daha kolaydır. (James, Witten, Hastie, & Tibshirani, 2017)
- Orta/büyük etkiye sahip yalnızca birkaç değişken varsa en etkili sonuç elde edilmektedir.
- Katsayıyı sıfıra doğru daraltarak işe yarayan veya istenilen özellikleri seçme imkanı vardır.
- Aşırı uyum (overfitting) durumunun oluşmasını önlemektedir.

3.1.2.5. Dezavantajları

- Seçilen özellikler oldukça etki altında kalmış olacaktır.
- n << p için (n: veri noktası sayısı, p: özellik sayısı), LASSO en fazla n özelliği seçmektedir.
- Birbiriyle ilişkili bir özellik grubundan yalnızca bir özellik seçecektir ve seçim doğası gereği gelişigüzel olacaktır.
- Farklı test veriler için, seçilen özellik çok farklı olabilmektedir.
- Tahmin performansı Ridge regresyonundan daha kötüdür.
- En genel dezavantajı, otomatik olmasıdır. Bu nedenden dolayı:
    - Mantıklı olmayan modeller ortaya çıkabilmektedir.
    - İlginç veya önemli olabilecek önemsiz değişkenleri otomatik olarak göz ardı edebilmektedir.
    - Hiyerarşiyi dikkate almamaktadır.

3.1.3. Elastic Net Regresyonu

3.1.3.1. Genel Sınıflandırma Özellikleri

Katsayıların hem L1 hem de L2 norm düzenlileştirmesi ile eğitildiği doğrusal bir regresyon modelidir. (Scikit-Learn, 2020)

3.1.3.2. Tanım

Elastic Net, Ridge Regresyonu ve Lasso Regresyonu arasında bir orta yoldur. Düzenlileştirme terimi hem Ridge hem de Lasso'nun düzenlileştirme terimlerinin basit bir karışımıdır ve karışım oranı r katsayısı ile kontrol edilebilmektedir. $r = 0$ olduğunda, Elastic Net, Ridge

Regresyonuna eşdeğerdir ve $r = 1$ olduğunda ise Lasso Regresyonuna eşdeğerdir (bakınız Denklem *3.1.3.2.*1) (Géron, 2019).

Denklem 3.1.3.2.1 Elastic Net maliyet fonksiyonu

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^{n} \theta_i^2$$

- **θ** modelin parametre vektörüdür, sapma terimi θ_0 ve θ_1 den θ_n 'e kadar olan özellik ağırlıklarını içermektedir.
- α modelin düzenli hale getirilmesinin derecesini kontrol etmektedir (Hiperparametre).
- r düzenlileştirme terimlerinin karışım oranını kontrol etmektedir.

Elastik Net Regresyonu, Lasso ve Ridge Regresyonlarının güçlü yönlerini birleştirerek, düzenlileştirilmiş değişkenlerle ilişkili parametreleri gruplandırıp küçülterek onları denklemde bırakmakta veya hepsini bir kerede kaldırmaktadır. Bundan dolayı, birbiriyle ilişkili birden fazla özellik olduğunda kullanışlıdır. Lasso bunlardan sadece birini rastgele seçmektedir. Elastik-net ise ikisini birden seçmektedir.

Elastik Net Regresyonunda katsayılar hem L1 hem de L2 norm düzenlileştirmesi ile eğitildiği için, Ridge Regresyonunun düzenlileştirme özelliklerini korurken, ağırlıkların birkaçının Lasso gibi sıfır olmadığı seyrek bir modelin öğrenilmesine olanak tanımaktadır ve L1 ve L2'nin dışbükey kombinasyonunu r (l1_ratio) parametresini kullanarak kontrol etmektedir. (Friedman, Hastie, & Tibshirani, 2010)

Öyleyse, Doğrusal Regresyon, Ridge, Lasso veya Elastic Net hangi durumlarda kullanılmalıdır? Doğrusal Regresyon sadece herhangi bir düzenleme gerekliliği olmadığında kullanılabilir. Ancak neredeyse her zaman en azından biraz düzenleme ihtiyacı ortaya çıkmaktadır. Bu nedenle genellikle düz Doğrusal Regresyondan kaçınılmalıdır. Eğer yalnızca birkaç özelliğin gerçekten yararlı olduğu bir durumla karşılaşılırsa Ridge Regresyonu iyi bir varsayılandır. Eğer yararsız özelliklerin ağırlıklarını sıfıra düşürme gerekliliği varsa Lasso veya Elastic Net tercih edilmelidir. Lasso veya Elastic Net arasındaki tercih ise, özelliklerin sayısının eğitim durumlarının sayısından fazla ve birkaç özellik arasında güçlü bir ilişki varsa, Elastik Net olmalıdır, çünkü böyle bir durumda Lasso düzensiz davranış gösterebilir, aksi halde Lasso ile devam edilmelidir.

3.1.3.3. Kullanım Alanları

Doğrusal Regresyon	Düzenlileştirmenin hiç gerekmediği durumlarda
Ridge Regresyonu	Düzenlileştirme gerekli ise (ve) Özelliklerin tamamı gerekli ise.
Lasso Regresyonu	Düzenlileştirme gerekli ise ve, Yararsız özelliklerin ağırlıklarının sıfıra düşürülmesi gerektiğinde (ve) Özelliklerin sayısı eğitim durumlarının sayısından az olduğunda (veya) Güçlü bir şekilde ilişkilendirilmiş birkaç seçilmiş özellik olmadığında

Elastic Net Regresyonu	Düzenlileştirme gerekli ise ve, Yararsız özelliklerin ağırlıklarının sıfıra düşürülmesi gerektiğinde (ve) Özelliklerin sayısı eğitim durumlarının sayısından fazla olduğunda (veya) Birkaç özellik güçlü bir şekilde ilişkilendirildiğinde

3.1.3.4. Avantajları

- {n << p} olduğunda n'den fazla tahminci seçme problemi yoktur, oysa LASSO {n << p} olduğunda en fazla n özelliği seçmektedir. (p: boyut sayısı, n: nokta sayısı)
- Hem Lasso hem de Ridge Regresyonundan elde edilen faydaların her ikisini de sağlamaktadır.
- Ridge tipi ceza yoluyla düzenlileştirme ve Lasso benzeri ceza yoluyla özellik seçimi sağlamaktadır (Zou & Hastie, 2005).
- Özellik seçimini gerçekleştirirken daha iyi tahmin gücüne sahip olan Lasso Regresyonundan faydalanılmaktadır.
- Ridge Regresyonunun özellik grubu seçimiyle Lasso Regresyonunun özellik seçiminin her ikisini de bünyesinde bulundurarak, her iki modelinde en iyi yönlerini tek bir modelde buluşturmaktadır.
- Yüksek korelasyonlu tahmin gruplarıyla başa çıkmak için iyidir.

3.1.3.5. Dezavantajları

- Hesaplama açısından LASSO veya Ridge Regresyonundan daha fazla işlem gerektirmektedir ve daha fazla zaman almaktadır.

3.1.4. LARS Regresyonu (Least Angle Regression)

3.1.4.1. Genel Sınıflandırma Özellikleri

LARS regresyonu, L1 düzenliliği kullanılmaktadır (Wikipedia, Least-angle regression, 2020).

3.1.4.2. Tanım

Bradley Efron, Trevor Hastie, Iain Johnstone ve Robert Tibshirani (Efron, Hastie, Johnstone, & Tibshirani, 2004) tarafından geliştirilen en küçük açı regresyonu (LARS), istatistikte, doğrusal regresyon modellerini çok boyutlu verilere uydurmak için kullanılan bir algoritmadır (Wikipedia, 2020).

Bu algoritmanın ilk defa açıklandığı makalede (Efron, Hastie, Johnstone, & Tibshirani, 2004) de üzerinde durulduğu gibi, kısaltılmış kod adı LARS olarak belirtilen en küçük açı regresyonunun (LAR) sonuna eklenen ‘S’ ise "Lasso" ve "Stagewise" anlamına gelmektedir (Efron, Hastie, Johnstone, & Tibshirani, 2004) ve ilgili makalede de üzerinde durulduğu gibi hem Lasso hem de Stagewise, En Küçük Açı Regresyonu (LARS) adı verilen temel bir prosedürün varyantlarıdır.

LARS, ileri yönlü aşamalı regresyona benzemektedir. Her adımda, hedefle en ilişkili özellik bulunmaktadır. Eşit korelasyona sahip birden fazla özellik olduğunda, aynı özellik boyunca devam etmek yerine, özellikler arasında eşit açılı bir yönde ilerlemektedir. (Scikit-Learn, 2020)

Aşırı uyum problemi ile karşılaşıldığında veya modelin kolayca yorumlanabilir olması istendiğinde En Küçük Açı Regresyonu (LARS), doğrusal regresyon için bir model seçim yöntemidir.

İleri Seçim ve Geriye Doğru Eliminasyon gibi model seçim algoritmalarının amacı, modelin uygulanacağı aynı veri kümesi temelinde doğrusal bir model seçmektir. En Küçük Açı Regresyonu (LARS), geleneksel ileri seçim yöntemlerinin kullanışlı ve daha az açgözlü bir versiyonudur. Bu amaçla üç ana özellik türetilmiştir:

1. LARS algoritmasında yapılan basit bir değişiklik ile Lasso Regresyonu (mutlak regresyon katsayılarının toplamını sınırlayan sıradan en küçük karelerin kullanışlı bir versiyonu) elde edilmektedir. LARS'a uygulanan değişiklik, belirli bir problem için olası tüm Lasso Regresyon tahminlerini önceki yöntemlerden daha az bilgisayar zamanı kullanarak hesaplamaktadır.
2. Farklı bir LARS modifikasyonu ise başka bir model seçim yöntemi olan Forward Stagewise doğrusal regresyonunu verimli bir şekilde uygulamaktadır. Uygulanan bu modifikasyon, daha önce Lasso ve Stagewise için gözlemlenen benzer sayısal sonuçları açıklamakta ve daha basit LARS algoritmasının kısıtlı sürümleri olarak görülen her iki yöntemin özelliklerini anlamaya yardımcı olmaktadır.
3. LARS tahmininin serbestlik dereceleri için kullanılacak bir tahmin hatası tahmini türetilmektedir. Bu, olası LARS tahminleri aralığı arasında ilkeli bir seçimin yapılabilmesini sağlamaktadır.

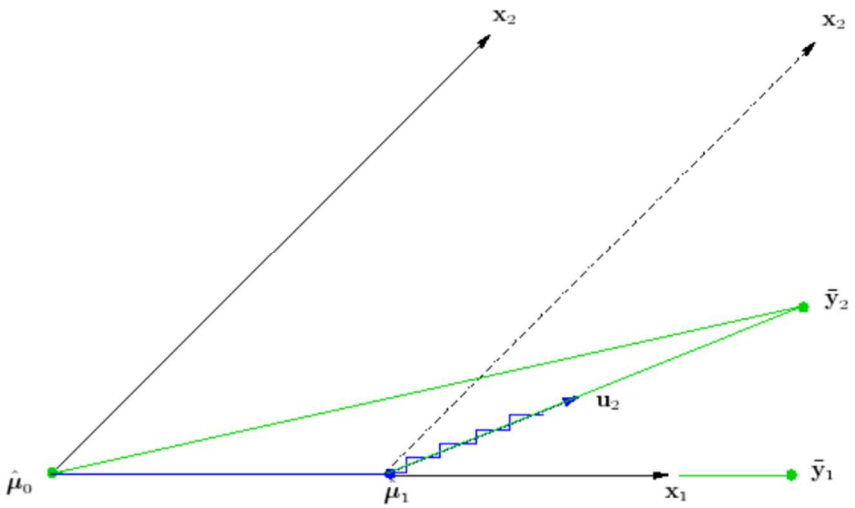
LARS Algoritması:

En Küçük Açı Regresyonu, hesaplamaları hızlandırmak için basit bir matematiksel formül kullanan Stagewise prosedürünün stilize edilmiş bir versiyonudur. Tam çözüm seti için yalnızca m adım gereklidir, burada m ortak değişkenlerin sayısıdır.

LARS prosedürü kabaca aşağıdaki gibi çalışır:

1. Klasik İleri Seçimde olduğu gibi, başlangıçta tüm katsayılar sıfıra eşitlenmektedir
2. Yanıtla en çok ilişkili tahmin ediciyi bulmaktadır, x_{j_1}.
3. Başka bir tahminci, diyelim ki x_{j_2}, mevcut geriye kalanlarla aynı oranda korelasyona sahip olana kadar eldeki tahminci yönünde mümkün olan en büyük adım atılmaktadır. Bu noktada, LARS Regresyonu parçaları İleri Seçim Regresyonu ile elde edilebilecek parçalarla aynıdır.
4. x_{j_1} boyunca devam etmek yerine, LARS, üçüncü bir x_{j_3} değişkeni "en korelasyonlu" kümeye girene kadar iki tahminci arasında eşit açılı bir yönde ilerlemektedir.
5. Daha sonra LARS, dördüncü bir değişken girene kadar x_{j_1}, x_{j_2}ve x_{j_3}arasında eşit açılı olarak, yani "en az açı yönü" boyunca ilerler ve bu böyle devam etmektedir.

LARS algoritmasının işleyişi Şekil 1'deki geometri ile de gösterilebilir. Şekildeki $\bar{y}_2$, LARS algoritmasının m = 2 ortak değişken durumunda, y'nin $L(x_1, x_2)$'ye olan izdüşümüdür. $\hat{\mu}_0$ = 0'dan başlayarak, artık vektör $\bar{y}_2 - \hat{\mu}_0$, x_1 ile x_2'den daha fazla korelasyona sahiptir; sonraki LARS tahmini $\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 x_1$ dir, burada $\hat{\gamma}_1$, $\bar{y}_2 - \hat{\mu}_1$, x_1 ve x_2 arasındaki açıyı ikiye bölecek şekilde seçilmektedir; o zaman u_2 birim açıortay olmak üzere $\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 u_2$ dir; m = 2 durumunda $\hat{\mu}_2 = \bar{y}_2$ değerine eşittir, ancak m> 2 durumunda bu eşitlik geçerli değildir.



Şekil 1. LARS Regresyonu Algoritması Geometrisi (Efron, Hastie, Johnstone, & Tibshirani, 2004)

3.1.4.3. Kullanım Alanları

- Aşırı uyum problemi ile karşılaşıldığında veya modelin kolayca yorumlanabilir olması istendiğinde En Küçük Açı Regresyonu (diğer adıyla LARS), doğrusal regresyon için en uygun yöntemdir.
- Sıradan en küçük kareler (OLS) ile aynı karmaşıklıkta tüm düzenlileştirme yolu çözümlerini hesaplayan bir yol bulma algoritmasına ihtiyaç olduğu durumlarda kullanılmaktadır.
- L1 düzenlileştirilmiş doğrusal regresyonu veya lojistik regresyonu çözmek için kullanılmaktadır.

3.1.4.4. Avantajları

- {p >> n} olduğu bağlamlarda sayısal olarak etkilidir (yani, boyutların sayısı nokta sayısından önemli ölçüde daha büyük olduğunda) (p: boyut sayısı, n: nokta sayısı)
- Özelliklerin sayısı, veri örneklerinin sayısından çok daha fazla olduğunda sayısal olarak çok etkilidir.
- Hesaplama açısından ileri seçim kadar hızlıdır ve sıradan en küçük kareler ile aynı karmaşıklık düzeyine sahiptir.
- Çapraz doğrulamada veya modeli ayarlamada yararlı olan tam bir parçalı doğrusal çözüm yolu üretir.
- Eğer değişkenlerden ikisi yanıtla hemen hemen eşit derecede ilişkiliyse, katsayıları yaklaşık olarak aynı oranda artmalıdır. Bu nedenle algoritma, tamda beklediği gibi davranır ve ayrıca daha kararlıdır.
- Lasso ve İleri Stagewise Regresyonlarında olduğu gibi, diğer tahmin ediciler için çözümler üretmek amacıyla kolayca değiştirilebilir.

3.1.4.5. Dezavantajları

- LARS, artıkların yinelemeli bir şekilde yeniden yerleştirilmesine dayandığından, özellikle gürültünün etkilerine karşı hassastır. Bu sorun Efron ve ark. (2004) tarafından Annals of Statistics dergisinde yayınlanan makalesinin devamındaki tartışma

bölümünde detaylı bir şekilde ele alınmaktadır (Efron, Hastie, Johnstone, & Tibshirani, 2004).

- Gerçek dünyadaki neredeyse tüm çok boyutlu veriler, sadece şans eseri, en azından bazı değişkenler arasında bir miktar doğrusallık sergileyeceğinden, LARS'ın ilişkili değişkenlerle ilgili sorunu, uygulanmasını çok boyutlu verilerle sınırlayabilmektedir.

3.2. Uygulama

Örnek araştırma için Python programlama dili kullanılarak örnek bir program oluşturulmuştur ve Anaconda bulut ortamından Jupyter Notebook olarak paylaşılmıştır (Aytekin, 2021). Ayrıca, programın çalıştırılacağı Python ortamı ve gerekli kütüphaneleri önceden yüklenmiş olmalıdır. Bu araştırmada kullanılan veriler Dünya Bankası Açık Veri ortamından alınan çevrimiçi veri setleri (World Bank Open Data, 2021) ve bütün veri setlerinin yer aldığı veritabanının tamamını içeren CSV formatındaki Dünya Gelişim Göstergesi zaman serisi veri setleridir (World Bank, World Development Indicators, 2021).

Geliştirilen yazılım dört ana sınıftan oluşmaktadır. İlgili Python sınıfları ve sunduğu faydalar aşağıda belirtilmiştir:

- **LoadWorldBankData**: Bu sınıfta dünya bankasının çevrimiçi veri altyapısına nasıl erişileceği ve istenilen veri setlerinin nasıl indirileceği gösterilmektedir.
- **LoadWorldBankWDIData**: Bu sınıf, indirilmiş olan CSV formatındaki tüm veri tabanı veri setlerini içeren verilerin Python ortamına nasıl yükleneceğini göstermektedir.
- **PrepareCleanData**: LoadWorldBankWDIData sınıfı ile ortama yüklenen ham verilerin nasıl gözden geçirileceği ve analiz işleminde kullanılacak verilerin ham verilerden nasıl elde edilebileceği birçok değişik yöntem ile bu sınıfta ele alınmıştır. Bu sınıfta gerçekleştirilen temizleme ve ayıklama işlevlerinden de görüleceği gibi, analizin en zor kısmı verileri anlamak ve sonrasında da işlem yapacağımız veri setini oluşturmaktır.
- **AnalyzeData**: PrepareCleanData sınıfındaki metotlar ile hazırlanmış işlenmeye ve analize hazır veri seti kullanılarak yapılacak analizlerin sistematiği ve gerekli ayarlamaları bu sınıfta ele alınmıştır. Hazır hale gelen veriler birçok şekilde modellenerek her bir modelde elde edilen başarımı ölçmek ve sıralayabilmek için çalıştırılarak kök ortalama kare hata (RMSE) sonuçları listelenmiştir.

Verinin hazırlanması, birleştirilmesi ve temizlenmesi için LoadWorldBankData**,** LoadWorldBankWDIData sınıfları kullanılarak elde edilen dünya bankası verileri PrepareCleanData sınıfı ile hazırlanma, birleştirme ve temizleme işleminden geçirilerek işlenmeye hazır hale getirilmiştir.

Verileri analiz etmek için ise AnalyzeData sınıfı kullanılmaktadır. Bu sınıf analiz modellerini n hepsini içermektedir. AnalyzeData sınıfı içindeki yazılımın çok küçük bir parçası Ridge Reg resyonu tahmin modeli için aşağıda verilen örnek Python kodunda, modelleme (`model_ridge = RidgeCV(scoring=scorer, cv=5)`), eğitim (`model_ridge.fit(data_train_regressors_std.loc[:, self.target], data_train_targets_subset)`), tahmin oluşturma (`predictions = model_ridge.predict(data_test_regressors_std.loc[:, self.target])`) ve tahmin hatalarının istatistiklerini elde etme (`mse = mean_squared_error(data_test_targets_subset, predictions)`) adımları görülmektedir.

AnalyzeData sınıfı Ridge Regressin modelleme örnek kodu:

```
# Ridge Regression
scorer = make_scorer(mean_squared_error)
```

```
model_ridge = RidgeCV(scoring=scorer, cv=5)
model_ridge.fit(data_train_regressors_std.loc[:, self.target], data_train_targets_subset)

# Tahminleri oluşturalım
predictions = model_ridge.predict(data_test_regressors_std.loc[:, self.target])

mse = mean_squared_error(data_test_targets_subset, predictions)
print("%-30s %8.4f %8.2f" % ("Ridge Regression", np.sqrt(mse), self.model_timer.reset()))
```

Verinin hazırlanması, birleştirilmesi, temizlenmesi, keşfe yönelik veri analizi, test edilecek modellerin oluşturulması, eğitimi, test edilmesi ve değerlendirmesi ile ilgili yazılımın tamamına https://anaconda.org/hta_65/ufuk-kongre-ybs-sunum-demo/notebook edresinden (Aytekin, 2021) erişilebilir.

Programın çalıştırılması ile elde edilen sonuçlar Tablo 2'de verilmiştir.

Tablo 2. Python programı ile değerlendirilen World Bank Development Indicies verilerinin modellerinin tahmin performansları

Model	**RMSE**	**Time**
Exponential Smoothing (Holt)	10,1103	4,71
Linear Regression	2,0127	82,24
Ridge Regression	2,1532	0,52
Lasso Regression	3,3680	15,80
LARS Regression	3,2265	1,82
ElasticNet Regression	3,8111	17,08
Time-Series LassoCV Regression	3,4684	12,60
Naive Predictor	9,5067	31,19
Random Forest	3,0415	93,77
Gradient Boosting	2,4705	8,68

Tablo 1.'deki sonuçlardaki Time sütunu, ilgili modelin çalışmasının sonuçlanması için geçen süreyi göstermektedir.

4. TARTIŞMA VE SONUÇ

Bu araştırmadaki kazanımlardan birisi, Dünya Bankası Veri setlerine Python programlama dili kullanılarak çevrimiçi olarak nasıl erişilebileceğinin uygulamalı olarak gösterilmiş olmasıdır.

Bundan daha önemli olan ise çalışma ortamına yüklenen ham verilerin nasıl gözden geçirileceği ve analiz işleminde kullanılacak verilerin ham verilerden nasıl elde edilebileceği birçok değişik yöntem ile ele alınmış olmasıdır. İlgili Python kodları incelendiğinde, gerçekleştirilen temizleme ve ayıklama işlevlerinin, analizin kendisinden çok daha zor olduğu görülecektir. Doğru bir analiz yapabilmek için gereken en önemli ilk adım eldeki verileri anlamak ve sonrasında da işlem yapacağımız veri setini oluşturmaktır.

Hangi durumda hangi modelin kullanılmasının gerektiğini teknik olarak analiz ve modelleme öncesinde belirlemek mümkündür. Ancak, bu çalışmada asıl üzerinde durulan konu, bu bilgiye deneme yanılma yolu ile de ulaşılabilecek olunmasının gösterilmesidir. Böylece eldeki verilerin analizi ve tahmin modellerinin oluşturulabilmesi için gereken ön araştırma, bilgisayar ve veri bilimleri ile ilgisi olmayan dallardaki araştırmacıların da kullanımına sunulabilecektir.

Makine öğrenimi, birçok alanda olduğu gibi toplumsal gelişimi anlamada da yol gösterici bir işlev olarak kullanılabilmektedir. Aynı şekilde, farklı ekonomik göstergeler ve diğer farklı veri kaynakları arasındaki karmaşık ilişkilerin ortaya çıkarılmasına yardımcı olabilmektedir. Bu işlevi göz önüne serebilmek için Dünya Bankası tarafından sağlanan Dünya Gelişim Göstergesi zaman serisi verileri ile hazırlanan örnek araştırmada çoğunlukla doğrusal tekniklere ve yüksek boyutlu verileri işlemek için uygun olan diğer bazı yöntemlere odaklanılmıştır.

Bu örnek vakada, ilk olarak veriler incelenmiş ve makine öğrenimi modellerinin eğitimi için kullanılacak eğitim ve test verileri hazırlanmıştır. Verilerin anlaşılması çok önemlidir. Detaylı bir şekilde veriler üzerinde çalışılması gerekliliği açıkça hazırlık aşamasında ortaya çıkmaktadır. Verileri anlaşılır ve tutarlı bir şekilde hazırlamak, modellemek kadar önemli ve zordur. Hazırlanan veriler ayrı ayrı doğrusal regresyon, Ridge regresyonu, Lasso regresyonu, LARS regresyonu, ElasticNet regresyonu, zaman serisi LassoCV regresyonu, Naive Predictor, Random Forest ve Gradient Boosting algoritmaları ile modellenerek değerlendirilmişlerdir. Değerlendirme kriteri olarak, her bir modelin eğitiminden sonraki tahmin performansı kök ortalama kare hatası yöntemiyle değerlendirilmiştir.

Her model için Dünya Bankası tarafından sağlanan Dünya Gelişim Göstergesi zaman serisi verileri ile hazırlanan eğitim ve test verileri kullanılmıştır. Tahminler için her eğitilen modelde aynı test verileri kullanılmıştır. Hazırlanan örnek araştırma yazılımı birincisi kök ortalama kare hatası (RMSE) ve ikincisi de ilgili modelin tahmin sonuçlarını üretmesi için geçen program çalışma zamanı olmak üzere iki çıktı ile araştırmacılara değerlendirme sonuç verileri sağlamaktadır. Sonuç, tamamen araştırmacının hedefine bağlıdır. Tablo 1 incelendiğinde en düşük kök ortalama kare hatasının doğrusal regresyon modeli ile elde edildiği görülmektedir. Ancak araştırmacının kriteri hem en düşük kök ortalama kare hatası hem de işlem zamanı ise Ridge regresyonu modeli daha uygun bir seçim olabilir.

Bu çalışmada amaçlanan çıktı, veri analizinde makine öğrenimi yöntemlerinin nasıl kullanılabileceğinin gösterilmesidir. Zaman serisi verilerinin, değişik makine öğrenimi modelleme teknikleri kullanılarak işlenmesi sonucunda Tablo 1.'deki sonuçlara ulaşılmıştır. Bu sonuçlar, araştırmacının amacına uygun bir şekilde değerlendirilebilecek bir yapıdadır. Araştırmacılar, amaçlarına uygun bir değerlendirme sonucunda karar verecekleri metodun makine öğrenimi modelini daha sonra kullanılmak üzere diskte saklayabilir ve sonraki zamanlarda ise bu modeli yeni verilerle eğitmeye kaldıkları yerden devam edebilirler. Böylece modelin tahmin performansı da sürekli olarak artırılabilir.

KAYNAKÇA

Aytekin, H. T. (2021, Şubat 12). Jupyter Notebook. Ankara. https://anaconda.org/hta_65/ufuk-kongre-ybs-sunum-demo/notebook adresinden alındı

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least Angle Regression. *Annals of Statistics, 2*(32), s. 407-499. http://statweb.stanford.edu/~tibs/ftp/lars.pdf adresinden alındı

Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization Path For Generalized Linear Models by Coordinate Descent. *Journal of Statistical Software*(33). https://www.jstatsoft.org/article/view/v033i01 adresinden alındı

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media.

Gujarati, N., D., & Madsen, J. B. (1998, February). Basic econometrics. *Journal of Applied Econometrics*(13), s. 209-212.

Hoerl, A. E., & Kennard, a. R. (1970, January). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*(12), s. 55-67. https://www.math.arizona.edu/~hzhang/math574m/Read/RidgeRegressionBiasedEstimationForNonorthogonalProblems.pdf adresinden alındı

Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology, 12*(69), s. 14413-14423.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning.* New York Heidelberg Dordrecht London: Springer.

Kratsios, A. (2020). *Deep Arbitrage-Free Learning in a Generalized HJM Framework via Arbitrage-Regularization Data.* https://www.mdpi.com/2227-9091/8/2/40 adresinden alındı

Mitchell, T. (1997). *Machine Learning.* New York: McGraw Hill.

NCSS. (2020). Ridge Regression. *NCSS Statistical Software.* içinde NCSS Statistical Software. 02 05, 2021 tarihinde https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf adresinden alındı

Neumaier, A. (1998). *Solving ill-conditioned and singular linear systems: A tutorial on regularization.* https://www.mat.univie.ac.at/~neum/ms/regtutorial.pdf adresinden alındı

Samuel, A. L. (1959, July). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, s. 211-229. doi:https://doi.org/10.1147/rd.33.0210

Scikit-Learn. (2020). *Linear Models, 1.1.3. Lasso, scikit-learn 0.23.2, User Guide.* Scikit-Learn: https://scikit-learn.org/stable/modules/linear_model.html adresinden alındı

Tibshirani, R. (1996). Regularized shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*(58(1)), s. 267-288.

Wikipedia. (2020). *Least-angle regression*. Wikipedia: https://en.wikipedia.org/wiki/Least-angle_regression adresinden alındı

Wikipedia. (2021). *Regularization (mathematics)*. Wikipedia: https://en.wikipedia.org/wiki/Regularization_(mathematics) adresinden alındı

World Bank Open Data. (2021). World Bank, Open Data for Online Retrieval. https://data.worldbank.org/ adresinden alındı

World Bank, World Development Indicators. (2021). World Development Indicators. http://databank.worldbank.org/data/download/WDI_csv.zip adresinden alındı

Zou, H., & Hastie, T. (2005, March 09). Regularization and variable selection via the elastic net. *2*(67), 301-320. https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2005.00503.x adresinden alındı