T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
FEN FAKÜLTESİ
İSTATİSTİK BÖLÜMÜ

SINIFLANDIRMA İÇİN MAKİNE ÖĞRENMESİ

Bitirme Projesi Raporu

Hüseyin Cem ARAS
İbrahim Berkay ÖZ

Haziran 2020

Rapor Değerlendirme

"SINIFLANDIRMA İÇİN MAKİNE ÖĞRENMESİ" başlıklı bitirme projesi raporu tarafımdan okunmuş, kapsamı ve niteliği açısından bir Bitirme Projesi raporu olarak kabul edilmiştir.

Dr. Özgül VUPA ÇİLENGİROĞLU

Teşekkür

Tüm çalışma süresince yönlendiriciliği, katkıları ve yardımları ile yanımızda olan danışmanımız Dr. Özgül VUPA ÇİLENGİROĞLU'na, böyle bir çalışmayı yapmamız için bize fırsat tanıyan Dokuz Eylül Üniversitesi Fen Fakültesi İstatistik Bölümüne, sanayi-üniversite işbirliği kapsamında bize veri ve bilgi desteği sağlayan ve sanayi kolumuz olan SANEM Plastik AŞ. Yönetim Kuruluna ve Tasarım Merkezi Müdürü Alim Fatih KILINÇ'a teşekkür ederiz.

Hüseyin Cem ARAS
İbrahim Berkay ÖZ

Özet

Projemizde kullanilan veriler 2018-2019 yillari arasini kapsamaktadir. Izmir ili Torbali ilçesinde plastik masa örtüsü üreten SANEM PLASTIK firmasina ait bir fabrikadan alinmistir. Alinan veriler üzerinde makine ögrenimi algoritmalarini kullanarak modeller olusturulmustur. Amaç bu modellerden elde edilen bilgilere dogrultusunda fabrikanin verimliligini arttirmaktir.
Anahtar Kelimeler: : Makine Ögrenmesi, Siniflandirma, Lojistik Regresyon, CART, Karar Agaçlari

İçindekiler

Tablo Listesi

Şekil Listesi

Bölüm 1

Büyük Veri

Gelişen bilgi ve iletişim teknolojilerinin kapsamında kabul edilen internet teknolojileri; web sayfaları, bloglar, sosyal medya uygulamaları, sensörler ve daha pek çok veri toplayan cihaz ve uygulamalar sayesinde her an bilimsel olan veya olmayan veriler toplanır hale gelmiştir. Toplanan bu veriler, pazarlama, halkla ilişkiler, bankacılık, güvenlik vb. pek çok alanın yanında araştırmacıların yaptıkları araştırmalarda da kullanılabilir nitelik taşıyabilmektedir.

Veri yığınlarının değerlerinin anlaşılması sonucunda, bu verileri toplama, işleme, kullanıcılara hazır hale getirme, erişime sunma, saklama, analiz etme gibi aşamalarda pek çok farklı yöntemler de kullanılabilmektedir. Bu verilerin büyük artış göstermesi ve bu artışa teknolojinin de destek vererek, yeni çözümler üretmesi ile birlikte "Büyük Veri" kavramı ortaya çıkmıştır.

Büyük veri, geleneksel veri tabanı yöntemlerinin kullanılması suretiyle işlenmesi mümkün olmayan, farklı hacimlerdeki heterojen veriyi tanımlayan yeni bir kavramdır ve çeşitli dijital içeriklerden oluşmaktadır (Gahi, Guennoun ve Mouftah, 2016, s. 953).

1.1 Veri Bileşenleri

Büyük veri'nin oluşumunda 5 bileşen vardır. Bu bileşenler sırasıyla; variety, velocity, volume, verification ve value 'dir. Genel olarak 5V şeklinde adlandırılmaktadır.

1. **Variety (Çeşitlilik)**: Üretilen veriler genel olarak yapısal olmadığı ve birçok farklı ortamdan elde edilen veri formatlarından oluştukları için bütünleşik ve birbirlerine dönüştürülebiliyor olmaları gerekmektedir.

2. **Velocity (Hız)**: Büyük veri üretimi her geçen gün hızına hız katmakta ve bu veriler saniyede inanılmaz boyutlara ulaşmaktadır. Hızlı büyüyen veri, o veriye muhtaç olan işlem sayısının ve çeşitliliğinin de aynı hızda artması sonucunu ortaya çıkartmaktadır. Hem yazılımsal hem de donanımsal olarak bu yoğunluğun kaldırabilmesi gerekmektedir.

3. **Volume (Hacim):** Büyük veri olarak isimlendirilen veriler her geçen gün hızına hız katarak artmaktadır. Gelecekteki durumlar göz önüne alınarak bu veri

yığınları ile nasıl başa çıkılacağı iyi düşünülmelidir ve planların bu doğrultuda yapılması gerekmektedir.

4. **Verification (Doğrulama):** Hızlı büyüyen verilerin güvenli olup olmadığının kontrol edilmesi gerekmektedir. Bu veri doğru kişiler tarafından görülebilir veya gizli kalması gerekiyor olabilir. Doğrulama verilerin akışının doğru katmanlardan, gerekli güvenlik ve gizlilik seviyesinde olması gerektiğini ifade eder.

5. **Value (Değer):** Verilerin diğer veri bileşenlerinde filtrelendikten sonra büyük verinin üretimi ve işlenmesi katmanlarında elde edilen verilerin, artı bir değer katması, anlamlı bir bilgi sunması gerekir. (Big Data Nedir?, 2018, s. 55)

1.2 Büyük Veri Kaynakları

Bugün düne göre daha fazla veri kaynağının varlığı söz konusudur. Akıllı telefonlar, tablet bilgisayarlar, sensörler, tıbbi ekipmanlar, web trafiği kayıtları, sosyal ağlardaki etkileşimler ve eczacılık, meteoroloji, simülasyon gibi alanlarda çözümler sunan bilimsel araştırmalar gibi birçok kaynak, büyük veriyi beslemektedir (Schneider, 2012, s. 6). Bununla birlikte web ortamının artan heterojenliği, web sayfaları üzerinde farklı medyalarda (metin, resim ve video), türlerde (ansiklopedi, haber, bloglar) ve konularda (eğlence, spor, teknoloji) büyük veri içeriğinin sağlanmasına neden olmaktadır (Achsas ve Nfaoui, 2017, s. 1).

Büyük veri çeşitliliğinin artmasında çok sayıda veri kaynağı etkili olmaktadır. Bu kaynaklardan bir kısmı tamamen yeni veri kaynağı olabilmekteyken, bazı veri kaynakları da mevcut verinin ayrışması, diğer bir ifadeyle mevcut kaynakların sayısal ortama aktarılması sonucu ortaya çıkmaktadır. Birçok endüstriyel alan, yeni veri üretimi ve mevcut verinin sayısallaştırılması şemsiyesi altına girmekte ve her biri ayrı bir büyük veri kaynağını oluşturmaktadır. Büyük veriyi büyüten endüstriler aşağıdaki gibi sıralanabilir (Ohlhorst, 2013, s. 41).

- **Taşımacılık, lojistik, perakendecilik, kamu hizmeti ve telekomünikasyon:** Taşımacılık, lojistik, perakendecilik, kamu hizmeti ve telekomünikasyon endüstriyel alanlarında kullanılan GPS alıcıvericileri, RFID etiket okuyucuları, akıllı sayaçlar ve telefonlarda yer alan sensörler vasıtasıyla gittikçe artan bir hızda veri toplanmaktadır. Toplanan bu veri, operasyonları optimize etmek, anlık olarak ortaya çıkan iş fırsatlarının farkına varmak ve örgütsel iş zekâsını (business intelligence) çalıştırmak amaçlı kullanılabilmektedir.

- **Sağlık hizmetleri:** Sağlık hizmetleri endüstrisi, hızlı bir şekilde elektronik tıbbi görüntüleme ve raporlamadan yararlanmaya doğru hareket etmektedir. Elektronik tıbbi görüntüleme ve raporlama verisine, kısa dönemli halk sağlığının gözlemlenmesinde ve uzun dönemli salgın hastalıkların araştırılmasında kullanılmak üzere ihtiyaç duyulmaktadır.

- **Kamu:** Birçok devlet kuruluşu, nüfus sayımı, enerji kullanımı, bütçe raporları, kanunsal yaptırım sonuçları, seçim sonuçları gibi halka ait raporları sayısal

ortama aktarmakta ve halkın erişimine sunmaktadır. Bu tarz veri, kamu kuruluşları ve bölgesel topluluklar tarafından tutulan ve geniş yelpazede faaliyet gösteren iş ve yönetim uygulamalarında kullanılabilen veridir. Bu verinin büyük çoğunluğu web ortamında serbestçe erişilebilecek durumdayken bazıları da belirli bir ücret karşılığı elde edilebilmektedir.

- **Eğlence medyası:** Kitap, gazete, magazin, televizyon, radyo, film, sinema, müzik ve oyun gibi birçok alanda hizmet veren eğlence endüstrisi, son 5 yılda artan bir hızda sayısal kayıt, üretim ve dağıtıma doğru bir geçiş sergilemiştir. Bugün eğlence medyasında kişi ve toplumların davranışlarını gözlemleyen geniş içerikte veri toplanmaktadır.

- **Yaşam bilimleri:** Yaşam bilimleri endüstrisindeki veri üretimine örnek olarak düşük maliyetli gen sayımı verilebilir. 1.000 Amerikan dolarından daha düşük maliyette gerçekleştirilebilen gen sayımı, genetik çeşitliliği araştırmada ve potansiyel tedavi etkinliğini belirlemede analiz edilebilecek onlarca terabaytlık veriyi oluşturmaktadır.

- **Video görüntüleme:** Video görüntüleme endüstrisinde, alt yazılı televizyon teknolojisinden IP temelli televizyon kameralarına ve kayıt sistemlerine doğru ilerleme kaydedilmiştir. IP temelli yeni teknolojik kamera verisi, güvenlik ve servis hizmetlerinin geliştirilmesi amacıyla analiz edilmek üzere toplanmaktadır.

Bölüm 2

Makine Öğrenmesi

Makine öğrenmesi, insanların ve hayvanların doğal olarak sahip olduğu, geçmiş deneyimlerden öğrenme yeteneğini, makinelere veriden öğrenme yoluyla uygulayan, temelde algoritmalara, matematiğe ve istatistiğe dayanan bir veri analitiği yöntemidir.

Makine öğrenmesi, insanlar tarafından kolaylıkla anlaşılabilecek, basit sınıflandırıcı ifadeler üretmeyi amaçlar. Bunu yaparken de arka planda istatistiksel yöntemleri kullanır (Michie, Spiegelhalter ve Taylor, 1994, s.3).

Makine öğrenmesi sayesinde, önceki tecrübelere veya örnek veri setlerine dayanan bir işlemi optimize etmek için bilgisayarlar programlanabilmektedir. İstenen sınıflandırmalar bilgisayarda kısa sürede ve etkili bir şekilde yapılabilir, bu süreçler sonunda bir model oluşturulur ve bu model geleceğe yönelik öngörülerde bulunabilir veya denetim amacıyla kullanılabilmektedir.

Modelde hangi öğrenme yönteminin seçileceği veri setine ve hipoteze bağlı olarak değişebilmektedir. Problem çözme ve algoritma tasarımında kullanılan, "Problemi küçük parçalara ayırarak problemle baş etme", bu noktada kullanabilecek önemli bir çözüm yaklaşımıdır.

Brownlee (2014), makine öğrenmesi problemlerinin adım adım çözümünde kullanılabilecek uygulamalı makine öğrenmesi süreci sunmuştur:

1. Problem Tanımı
2. Veri Analizi
3. Veri Hazırlama
4. Algoritma Seçimi
5. Sonuçların İyileştirilmesi

2.1 Problem Tanımı

Makine öğrenmesi çalışmalarında bir algoritma tasarımı yapmaya başlamadan önce problemin daha iyi anlaşılması için bazı sorulara cevap verilmesi gerekir. Bunlar:

1. Problem nedir?
2. Çözülmesi durumunda getirileri nelerdir?
3. Nasıl çözülebilir?

2.1.1 Problem Nedir?

Problemin tam olarak ne olduğu (tanımı), hangi parametrelerin kullanılması gerektiği, hangi veriler ile çalışılması gerektiği, sonuçların nasıl test edileceği mutlaka en başta belirlenmelidir.

2.1.2 Çözülmesi Durumunda Getirileri Nelerdir?

Problem istenilen ölçüde çözüldüğü durumda, getirilerinin (faydaların) neler olacağı ortaya konulmalıdır. Çözümün sağlayacağı faydalara ek olarak, çözümün nasıl kullanılacağı da ayrıca listelenmelidir. Daha sonra, problemin karmaşıklığı ile elde edilecek getirilerin (kazanımların) oranı hesaplanmalıdır. Buradaki amaç, çok ciddi getirileri olmayan, karmaşık ve çözülmesi çok zor olan veya getirisinin maliyetinden büyük olan bir problem olup olmadığının tespit edilmesidir.

2.1.3 Nasıl Çözülebilir?

Problemi çözmek için öncelikle aşağıdaki sorulara cevap verilmelidir:

- Gerekli veriler neler?
- Bu veriler nasıl toplanmalı?
- Veriler nerede saklanmalı?
- Verilerin değişim hızı nedir?

Daha sonra, verilerin ön işleme adımlarının neler olacağı ve verinin nasıl hazırlanacağı düşünülmelidir. Son olarak ise nasıl bir veri bilimi yaklaşımının ya da algoritmanın kullanılması gerektiği üzerinde araştırma yapılmalıdır.

2.2 Veriyi Anlama

Makinenin deneyim olarak yararlanacağı veri, ele alınan probleme uygun bir biçimde temin edilir. Veri toplama aşamasında farklı kaynaklardan yararlanılabilmektedir. Bunlardan biri ele alınan probleme özgü orijinal veri setlerinin araştırmacılar tarafından oluşturulmasıdır. Bir diğeri ise internette yer alan hazır veri setleridir. Bu veri setleri, erişim ve kullanım kolaylığı bakımından makine öğrenmesi çalışmalarına avantaj sağlamaktadır. University of California, Irvine (UCI) Machine Learning Repository (Lichman, 2013 s, 45), Machine Learning Dataset Repository (Braun ve diğ., 2015 s, 102) web sayfalarından bu gibi paylaşıma açılan veri setlerine erişmek mümkündür.

Verinin analizler için hazırlaması belki de makine öğrenmesi sürecinin en zaman alıcı aşaması olarak kabul edilebilir. Modelin kurulmasından önce mevcut verinin iyi anlaşılması ve iyi analiz edilmesi gerekmektedir. Veri seti temin edildikten sonra veri hakkında ön fikir edinilmesi için bazı basit istatistiksel hesaplamalar yapılabilir ve grafikler çizilebilir. Niteliklerin kategorik ya da nümerik olmasına göre maksimum, minimum, mod, medyan, ortalama ya da kartil hesaplamaları yapılabilir, nitelikler kutu grafiği (box and whisker plot), histogram, sütun ve pasta grafikleri ile görselleştirilebilir.

Tüm bu işlemler, veri ön-işleme sürecinde hangi analizlerin gerçekleştirilmesi gerektiği hakkında da bilgi sunmaktadır.

2.3 Veri Hazırlama

Veri hazırlama, makine öğrenmesi projelerinin en önemli ve zaman alan aşamalarından birisidir. Hazırlanan veri, eğitilecek algoritmanın temel yapı taşı olacaktır. İlk olarak, tahmin yapmak istenilen durumu en iyi anlatan parametrelerden oluşan bir veri seti seçilmelidir. Daha sonra, veri setinde yapılacak ön işlemeler ye dönüşümler, algoritmaya uygun bir şekilde gerçekleştirilmelidir. Veri hazırlama aşamasını 3 adımda incelemek mümkündür:

- Veri Seçimi
- Veri Ön İşleme (Pre-processing)
- Veri Dönüştürme

2.3.1 Veri Seçimi

Veri seçimi, ham veri içerisinde projede tanımlanmış problemle ilgili olan değişkenlerden yeni bir veri seti oluşturulmalıdır. Bu veri seti, istatistiksel hesaplamada veya model oluşturmada bir anlam ifade edebilecek kadar büyük olmalıdır. Daha büyük veri daha anlamlıdır denemez. Gerçek hayat problemlerinde verinin yanlış yönde büyümesi, algoritmaların hesaplamasındaki karmaşıklığın artmasına neden olmakla birlikte performansı da olumsuz yönde etkilemektedir.

Bu nedenle, hedef problemi, direkt veya dolaylı yoldan etkileyen değişkenlerin belirlenmesi birinci önceliktedir. Elde bulunan ham bir veri setinde hedeflenen problemle ilişkisi bulunmayan gereksiz değişkenler de yer alabilir. Bu veriler ilk aşamada yeni veri setinde yer almamalıdır. Daha sonra, elde edilen algoritma sonuçlarına göre dolaylı etkisi olan değişkenler tekrar belirlenip, veri setine dahil edilebilirler.

2.3.2 Veri Ön İşleme

Veri Temizleme

Veri ön işlemenin ilk adımıdır. Bu adımda temel odak, eksik verilerin giderilmesi (doldurma/çıkarma), gürültülü verilerin ayıklanması ve aykırı değerlerin temizlenmesidir.

Kullanılan veri seti içerisindeki değişkenlerde eksik değerler bulunabilmektedir. Bu değerleri doldurmadan ya da eksik değerlere sahip değişkenleri veri setinden çıkarmadan makine öğrenmesi algoritmaları üzerinde çalışmak hataya neden olabilmektedir. Eksik veri içeren bir veri seti üzerinde uygulanabilecek yöntemlerden bazıları şunlardır:

1. **Eksik Verileri Göz Ardı Etme:** Eksik verinin sayısının, eksik olmayan veri sayısına göre çok düşük olduğu durumlarda, veri setinden eksik verileri tamamen çıkarmak çok fazla anlam kaybettirmeyecektir ve oldukça kolay bir çözüm yönetimidir.

2. **Eksik Verilerin Elle Doldurulması:** Eksik olan verilerin, veri kaynağına ulaşarak bulunması ve elle veri setine eklenmesi yöntemidir. Küçük miktardaki eksik veriler için uygulaması mümkün bir yöntem olmakla birlikte eksik veri sayısı ve önemi arttıkça zaman alıcı veya mümkün olmayan bir yönteme dönüşebilmektedir.
3. **Eksik Veriyi Hesaplama Yöntemleri İle Doldurma:** Eksik veriler, eksik olmayan verilerin ortalama, mod veya medyan değerleri ile doldurulabilmektedir. Verinin yapısı uygun ise bir önceki veya bir sonraki değer ile doldurulabildiği gibi herhangi bir makine öğrenmesi algoritması kullanılarak da tahmini değerler hesaplanarak doldurulabilmektedir.

2.3.3 Veri Dönüştürme

Veri hazırlama aşamalarının sonuncusu olan veri dönüştürmede, kullanılacak algoritmaya ve iş alanındaki amaçlara göre farklı yöntemler kullanılabilmektedir. Bu yöntemlerden en yaygın kullanılanları şu şekildedir:

Normalleştirme: Veri setinde yer alan nümerik değerlerin tümü aynı aralıklarda bulunmayabilir. Normalizasyon, verileri aynı ölçeğe, örneğin, 0-1 aralığına indirger (Shalabi, 2006). Bu sayede farklı ölçekteki verilerin birlikte ele alınabilmesi sağlanır. Verilerin dağılımı bilmediği durumlarda kullanmak için iyi bir yöntemdir. Veriler farklı yöntemlerle normalize edilebilmektedir. Bu yöntemlerden bazıları şu şekildedir (Han ve Kamber, 2006, s, 120):

- **Min-Max:**

$$yeni\_deger = \frac{deger - min}{max - min}(yeni\_max - yeni\_min) + yeni\_min$$

- **Z-Score:**

$$AA - yeni\_deger = \frac{deger - ortalama}{standart\ sapma}$$

Birleştirme-Toplama (Aggregation): Birlikte olduklarında daha anlamlı olduklarını, iş bilgisi ya da veri analizi sayesinde bilinen (keşfedilen) verilerin birlikte ele alınmasını sağlar (Dean ve Ghemawat, 2010). 2 kategorik değişkenin yeni bir grup olarak birleştirilip veri setine eklenmesi, bu dönüşüm yöntemine örnektir.

Kategorik Değişkenler: Makine öğrenmesi algoritmalarının birçoğu kategorik değişkenleri, bir dönüşüm yapmadan kullanamamaktadır. Kategorik bir değişkeni; "Sıralayıcı Değişkenlere Kodlama (Encoding to ordinal variables)" yöntemi ile içinde bulunan kategorileri sayılara dönüştürüp, bu sayıların kategorileri temsil etmesi sağlanabilmektedir. Başka bir yöntem olan "Tek Yüklü Kodlama (One hot encoding)" ile veri içerisinde bulunan her kategorinin bir sütunu temsil ettiği bir ikili matrise dönüşüm gerçekleştirilebilmektedir.

2.4 Algoritma Seçimi

Makine öğrenmesi projesinde öncelikle algoritma seçiminin hangi kategoriden yapılacağına karar verilmeli, daha sonra seçim yapılan kategori içerisinde bulunan algoritmalardan veri setine ve hipoteze uygun olanı seçilmelidir.

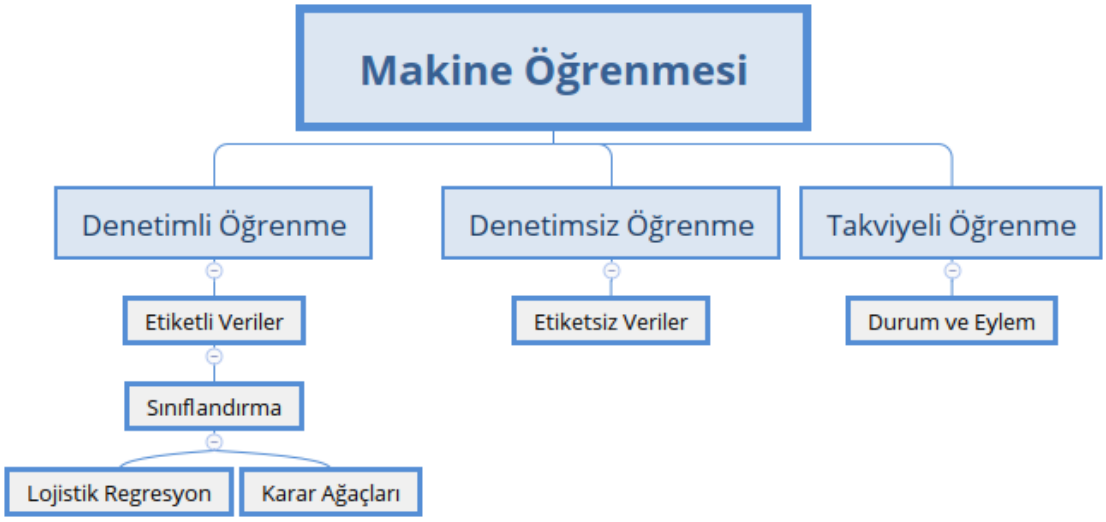
Bölüm 3

Algoritmalar

Makine öğrenmesi alanında yer alan yöntemler ve algoritmalar, öğrenme yöntemine göre üç kategoride incelenirler (Alpaydin, 2009).

- Denetimli Öğrenme (Supervised Learning)
- Denetimsiz Öğrenme (Unsupervised Learning)
- Takviyeli/Pekiştirmeli Öğrenme (Reinforcement Learning)

Bu çalışmada denetimli öğrenme üzerinden gidilmiştir.



Şekil 3.1: Makine Öğrenmesi

3.1 Denetimli Öğrenme

Eğitim verisinde bulunan girdileri ve bunlara ait etiketlenmiş çıktı değerlerini kullanarak model üreten (fonksiyon) ve test kümesi üzerinden bu modeli sınayan öğrenme yöntemidir (Onan, 2015). Örneğin; bir görüntü tanıma çalışması yapıldığını varsayılsın. Hangi görüntünün hangi cisme ait olduğu etiketlenmiş bir veri seti olsun. Bu veri

setini ve uygun algoritmayı kullanarak gerçekleştirilen öğrenme türüne **Denetimli Öğrenme** denilmektedir. Denetimli öğrenme yöntemlerini **Regresyon** ve **Sınıflandırma** olarak 2 grupta incelemek mümkündür.

3.1.1 Regresyon

İstatistik biliminin temel ilgi alanlarından olan regresyon, bağımlı bir değişkenin davranışının, bağımsız bir veya daha fazla değişken üzerinden modellenmesidir (Özift, 2014 s, 215). Bağımsız değişkenlere, bağımlı değişkeni etkiledikleri derecede bir katsayı atanır. Hipotez, nicel bağımlı bir değişken tahmin etmeye yöneliktir. Regresyon yöntemleri doğrusal ve doğrusal olmayan regresyon modelleri üzerinden incelenmektedir.

Doğrusal Regresyon

Regresyon analizinde temel amaç bağımlı değişkeni tahmin edecek en iyi modelin tahmin edilmesidir. Bir diğer ifadeyle bağımlı değişkendeki varyasyonu en iyi açıklayan denklemin oluşturulmasıdır. Regresyon modelindeki bağımsız değişken birinci dereceden ise bu model doğrusal model olarak ifade edilir.

Basit doğrusal regresyon modelinde bir bağımlı (Y) ve bir bağımsız değişken (X) vardır. Çoklu doğrusal regresyon modelinde ise bir bağımlı (Y) ve birden fazla bağımsız değişken (X_i) vardır.

(X_i Y_i) gözlemlerine ait basit doğrusal regresyon modeli aşağıdaki gibidir:

$$Y_i = \alpha + \beta X_i + \varepsilon_i \quad i = 1, 2, \ldots, n$$

Burada; X : Bağimsız değişken
Y : Bağımlı değişken
α : Regresyon doğrusunun Y eksenini kestiği değer
β : Regresyon doğrusunun eğimidir.
ε : hata değerleridir.

ε'nun ortalamasının sıfır, varyansının olduğu ve normal dağılış gösterdiği varsayılır. Bu bir hata değerinin başka bir hata değerinden etkilenmediği anlamına gelir. Yani hata terimleri arasında otokorelasyon yoktur. ε değerleri kesin olarak bilinmeyen, pozitif veya negatif değerler alabilen rassal bir değişkendir (Anderson et al. 1981; Gujarati 2005 s,132). Hata terimi Y bağımlı değişkenini etkileyen diğer değişkenlerin modele dahil edilmemesi, modelin yanlış seçilmesi, bilgi kaynağının homojen olmaması ve ölçme yanlışlıklarından dolayı ortaya çıkmaktadır (Kılıçbay 1980; Johnson and Wichern 1998; Gujarati 2005 s,141).

Lojistik Regresyon

Günümüzde biyoloji, tıp, tarım ve ekonomi, gibi alanlarda kolay kullanımı ve yorumlanması nedeniyle lojistik regresyon yaygın olarak kullanılan ve tercih edilen bir yöntem haline gelmiştir.

Doğrusal regresyon modelinden farkı ise, lojistik regresyon analizinde bağımlı değişkenin iki ya da çok sınıflı olmasıdır. Lojistik regresyon ve doğrusal regresyon

analizi arasındaki bu farklılık hem parametrik model seçimine hem de varsayımlara yansımaktadır [Hosmer, D. W., Lemeshow, S., 1989, 5-50 s,102]. Lojistik regresyon, normallik varsayımının bozulması nedeniyle doğrusal regresyon analizine alternatif olmaktadır. Doğrusal regresyon analizinde bağımlı değişkenin değeri, lojistik regresyon analizinde ise bağımlı değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilmektedir. Temel olarak lojistik regresyonda bağımsız değişkenler ile iki ya da çok sınıflı kategorik bağımlı değişken arasındaki ilişkinin tanımlanması için matematiksel modelleme yapmak amaçlanmaktadır [Kleinbaum,G., D., 1994 s,76].

Lojistik regresyonun modelleme aşamasında kullanılacak olan lojistik modeli elde etmek için yapılan adımlara aşağıda değinilmiştir.

$$y_i = \sum_{k=0}^{p} \beta_k X_{ik} + \varepsilon_i$$

şeklinde ifade edilen modelde bağimsiz değiskenler üzerinde bir kısit yoktur. Ayni zamanda y_i bağimli değisken değeri de $-\infty$ ile $+\infty$ arasinda tüm değerleri alabilmektedir. Bağımlı değiskenin 0 ve 1 gibi değerler aldığı durumda bu kural bozulmakta ve $P\left(y_i = 1\right)$ i. gözlemin 1 değerini alma olasılığı olmak üzere, beklenen değer:

$$E\left(y_i\right) = 1 \times P\left(y_i = 1\right) + 0 \times P\left(y_i = 0\right) = P\left(y_i = 1\right) \text{ olmaktadır.}$$

Bu sonuç regresyon denklemi olarak yazılacak olursa:

$$E\left(y_i\right) = P\left(y_i = 1\right) = \sum_{k=0}^{p} \beta_{k^x ik} \quad , i = 1, \ldots, n$$

ifadesi elde edilmektedir. Sol tarafı. $0 - 1$ arasinda olasılık değerleri alan bu denkleme "Doğrusal olasılık modeli" adı verilmektedir [Tatlıdil, H., 1996 s, 120] Doğrusal olasılık modelinde bağımlı değisken değeri olarak ifade edilen olasılık değerinin çesitli dönüşümlerle - $-\infty, +\infty$ arasinda tanimlı hale getirilmesi amaciyla yapılacak dönüşümlerden birisi lojit dönüşüm olup. lojit dönüşümde, ilk olarak;

$$E\left(y_i\right) = P\left(y_i = 1\right) = \sum_{k=0}^{p} \beta_{k^x ik} \quad , i = 1, \ldots, n$$

modelinde olasilik değerleri üzerinde $\frac{P}{1-P}$ dönüsümü yapılarak bağımlı değiskenin sinirları 0,1 yapilmakta, daha sonra ise bu oran değerinin logaritması alınarak bağımlı değişkenin sinirları -∞, + ∞ yapılmaktadir. Bu dönüşümlerden sonra elde edilen yeni fonksiyon:

$$g(x) = \ln\left[\frac{\pi(x)}{1 - \pi(x)}\right] = \beta_0 + \beta_1 X$$

olarak yazılabilir. Bu modele de "Lojistik model" ya da kısaca "Lojit" denmektedir. Transformasyon değiskeni $g(x)$, modeldeki parametreler ile doğrusaldir, süreklidir ve - ∞ ile $+\infty$ aralığında değisen değerler alir. $\pi(x)$ arttikça $g(x)$ 'te artar ve eğer $\pi(x) < 0,5$ ise $g(x)$ negatif, $\pi(x) > 0,5$ ise $g(x)$ pozitif değerler alır (Hosmer ve Lemeshov, 1989: 5,307).

$$P_i = \frac{\exp\left(\sum_{k=0}^{p} \beta_{k^x ik}\right)}{1 + \exp\left(\sum_{k=0}^{p} \beta_{k^x ik}\right)}$$

biçiminde tanimlanmaktadır [Collet, D., 2003].

Lojistik analizinde yapılması gereken önemli noktalardan biri kurulan model katsayısinin yorumlanmasidir. Bağımsız bir k_x değişkeninin katsayısı β_x, k_x 'da meydana gelen bir birim değişikliğin y bağımlı değiskeni üzerinde yarattığı değisimin. miktarını ve yönünü vermektedir. Bunun isin öncelikle bağimlı ve bağimsiz değiskenler arasindaki fonksiyonel iliskinin bulunmasi gereklidir. Bir modeldeki bağimsiz değiskenler ile bağımlı değişken arasındaki lineer ilişkiyi veren fonksiyona "link fonksiyonu" adı verilmektedir. Bağımlı değiskenin tanimı gereği parametrelerinde doğrusal olan doğrusal regresyon modelinde link fonksiyonu birim fonksiyon (matris) iken; lojistik regresyonda söz konusu fonksiyon logit dönüçümdür. Bu dönüşüm de,

$$g(x) = \ln\left\{\frac{P(x)}{[1 - P(x)]}\right\} = \beta_0 + \beta_1 x$$

Bu denkleme göre lojistik regresyon modelinde β_1 katsayısı, x bağimsiz değişkeninin bir birim değişiminin lojitte sağlayacağı değişim olup $\beta_1 = g(x+1) - g(x)$ olarak ifade edimektedir. Yani lojistik regresyon modelinde katsayının yorumu, iki lojit arasındaki farka anlam kazandırılması esasına dayanmaktadır.Buda odds ratio ile ifade edilir.

Odds Oranı Odds, başarı ya da görülme olasılığının "p", başarısızlık ya da görülmeme olasılığına "1-p" oranıdır. Odds ratio iki odds'un birbirine oranıdır. İki değişken arasındaki ilişkinin özet bir ölçüsüdür.

$$\frac{p/(1-p)}{q/(1-q)} = \frac{p(1-q)}{q(1-p)}$$

Regresyon Performansını Ölçme Yöntemleri

Kök Ortalama Kare Hatası (Root Mean Square Error - RMSE), modelin çıktısının rakam olduğu durumlarda, modelin tahmin kabiliyetini ölçmek için kullanılan yaygın bir yöntemdir (Chai ve Draxler, 2014). Ortalama Kare Hatası (Mean Square Error - MSE), artıkların karelerinin toplamlarının örnek sayısına bölünmesi ile elde edilir. Burada artıklar, gözlem değerleri ile tahmin değerlerinin farkından oluşur. MSE ölçütüne ait formül şu şekildedir: (e_t); değişkeni artıkları temsil etmektedir. RMSE ise MSE değerinin karekökünün alınması ile elde edilir) (Yücalar vd., 2016).

$$MSE = \frac{1}{n}\sum_{t=1}^{n} {e_t}^2$$

Yaygın kullanılan diğer bir ölçüt ise R^2'dir. R^2, bağımlı değişkenin/değişkenlerin, bağımsız değişkeni, kullanılan model ile ne derece açıkladığını göstermektedir. R^2 değeri 0 ile 1 arasında değerler almaktadır. Bu ölçüt 1'e ne kadar yakın ise modelin değişiminin o derece iyi açıklandığı söylenmektedir.

Yanlılık-Varyans ikilemi: MSE ölçütünün farklı bir şekilde ele alınması olarak düşünülebilmektedir. MSE ölçütü hesaplandığında veri setindeki verilerin birbirinden bağımsız olduğunu, artıkların ortalamasının sıfır, varyansının sabit ve σ^2 olduğunu varsayılsın. Bu durum şu şekilde bir denklem ile ifade edilir.

$$\mathrm{E[MSE]} = \sigma^2 + (\text{ Model yanlılığı})^2 + \text{ model varyansı}$$

Burada; **E** beklenen değeri temsil ederken, σ^2 azaltılamayan gürültü değerini temsil etmektedir. İkinci terim model yanlılık değerinin karesi olup, tahmin edilen değerler ile olması gereken değerler arasındaki yakınlığı temsil etmektedir. Varyansı düşük olan algoritmalar, karmaşıklığı daha az ve daha sabit alt yapıya sahip algoritmalar iken, yanlılığı düşük olan algoritmalar daha karmaşık ve daha esnek yapıya sahip algoritmalardır (Zhang vd., 2006). Karmaşıklığı çok düşük olan algoritmalar veriye tam uymaz ve veriden yeterince öğrenemezler. Oldukça karmaşık olan algoritmalar ise veriye aşırı uygunluk/uyum (overfitting) gösterirler, yani veriden öğrenmek yerine ezberleme yoluna giderler (Cao ve Tay, 2003 s,210).

İyi bir tahmin modelinde, toplam hatayı en küçükleyecek şekilde yanlılık-varyans dengesi kurulmalıdır. Bunu başarabilmek için izlenebilecek bazı yöntemler şu şekildedir:

- Veriyi eğitim ve test kümesi olarak ayırmak
- Uygun olan algoritmaları denemek
- Model parametrelerini uydurmak
- Etkili parametre ayarlamaları yapmak
- Uygun performans ölçütleri kullanmak
- Sistematik çapraz doğrulama kullanmak

3.1.2 Sınıflandırma

Sınıflandırma, verileri benzerliklerine ve farklılıklarına uygun şekilde birbirlerinden ayırmak/gruplamak için kullanılan bir yöntemdir (Sayad, 2008). Sınıflandırma yöntemleri, veri setinin iki veya çok boyutlu düzlem üzerindeki dağılımına uygun şekilde, birbirine benzer özellikler taşıyan örneklerin her birinin kendi içinde gruplanmasını sağlamak amacıyla, en uygun fonksiyonu belirlemek için kullanılır.

Problemdeki her girdi vektörünü, sonlu sayıdaki bir ayrık kategoriye atamayı amaçlayan durumlar, **sınıflandırma** (classification) problemi olarak ele alınmaktadır (Bishop, 2007 s,165).

Sınıflandırma problemlerinde çıktı uzayındaki her bir eleman birer **sınıf** (class), sınıflandırma problemini çözen algoritmaya da **sınıflandırıcı** (classifier) adı verilmektedir (Camastra ve Vinciarelli, 2008 s,147).

Sınıflandırma, her verinin hangi sınıfa ait olduğu bilindiği bir veri setinin eğitilerek, yeni gelecek bir test verisinin hangi sınıfa ait olduğunun tahmin edilmesi üzerine çalışır.

Makine öğrenmesinde sınıflandırma için kullanılan Naive Bayes Sınıflandırıcı, k-En Yakın Komşu Algoritması, Karar Ağaçları, Yapay Sinir Ağları, Destek Vektör Makineleri gibi çok sayıda algoritma mevcuttur. Bir sınıflandırıcı, bir başka ifade ile bir sınıflandırma modeli (classification model), örneklerden tahmin edilen sınıflara doğru bir haritalamadır (Fawcett, 2006 s,176).

Bu çalışmada karar ağacı algoritmalarından CART algoritması uygulanmıştır.

CART Algoritması

CART algoritması, ağaç yapısına dayalı olarak sınıflandırma ve regresyon modellerinin türetilmesi için yaygın olarak kullanılan bir istatistiksel prosedürdür. CART ağaç modeli, tek değişkenli ikili kararların bir hiyerarşisini içerir. CART verileri iki alt kümeye ayırdığı için her bir alt küme içindeki durumlar, bir önceki alt kümeden daha homojen olacaktır. Bu ardışık süreç, homojenlik kriterine ulaşılıncaya veya diğer bazı durma kriterleri sağlanıncaya kadar kendini tekrar eder. Aynı kestirim değişkeni ağaçta farklı düzeylerde pek çok kez kullanılabilir. Ağacın yapısı önceden belirlenmemekte, verilerden türetilmektedir (Answer Tree 3.0 User's Guide, 2001 s,189).

CART, kök düğümünde, verilerin iki gruba bölünmesi için en iyi değişkenin seçilmesini sağlar ve farklı bölümlendirme (splitting) kriterleri kullanır. Bu bölümlendirme kriterlerinin tümü, her bir alt kümedeki sınıf etiketlerini mümkün olduğunca homojen olacak biçimde bölümlendirir (Classification and Regression Trees: An Introduction, 2003: s,12). Bölümlendirme prosedürü çocuk düğümlere (child node) veya alt düğümlerin her birine ardışık olarak uygulanır (Hand, Manila ve Smyth, 2001 s,147).

CART ağaçları, kesin bir heterojenlik (impurity) ölçüsüne bağlı olarak düğümlere ayrılmış iki değerli (binary) ağaçlardır ve bu nedenle de sonuçta homojen dallar oluşmaktadır (Ahola ve Rinta-Runsala, 2001 s,17). Ağacın hedefi benzer veya aynı çıktı değerlerine sahip olma eğiliminde olan alt gruplar yaratmaktır. CART modelleri için bölünmelerin bulunmasında kullanılan dört farklı heterojenlik ölçüsü mevcuttur. Kategorik hedef degişkenler için Gini. Twoing veya (sıralayıcı hedef değişkenleri için) sıralı Twoing. sürekli hedef değişkenler için ise en küçük kareli sapma (LSD) kullanılabilir.

Gini indeksi aşağıdaki şekilde yazılabilir:

$$g(t) = 1 - \sum_j p^2(j/t)$$

Her hangi bir düğümde durumlar kategoriler arasında eşit biçimde dağıldığında, Gini indeksi $1 - \frac{1}{k}$ maksimum değerini alır. Bir düğümdeki durumlar aynı Bir düğümdeki durumlar aynı kategoriye ait olduğunda ise Gini indeksi 0'a eşit olacaktır (Apte ve Weiss,1997 s,41).

Twoing indeksi, hedef değişken kategorilerinin iki süper sınıfa bölümlendirilmesine dayalıdır ve ardından bu iki süper sınıfa dayalı olarak kestirim değişkenindeki en iyi bölünmeyi bulur. t düğümünde s bölünmesi için Twoing kriter fonksiyonu şu şekilde tanımlanabilir (Answer Tree 3.0 User's Guide, 2001 s,194):

$$\Phi_{(s,t)} = p_L p_R \left[\left| \sum_j p\left(\frac{j}{t_L} - p\left(\frac{j}{t_R}\right)\right) \right| \right]$$

Fonksiyonda yer alan t_L ve t_R, s bölünmesi tarafından yaratılan düğümleri göstermektedir. s bölünmesi, bu kriteri maksimize eden bölünme olarak belirlenir. İki süper sınıf olan C1 ve C2 aşağıdaki biçimde tanımlanabilir:

$$C_1 = \left\{ J : p\left(\frac{j}{t_L}\right) \geq p\left(\frac{j}{t_R}\right) \right\}$$
$$C_2 = C - C_1$$

Burada C, hedef değişkenin kategori kümesidir.

Sıralı Twoing indeksi, sıralayıcı hedef değişkenleri için Twoing indeksinin değiştirilmiş şeklidir. Sıralı Twoing kriterindeki farklılık yalnızcabitişik kategorilerin süper sınıflar ile birleştirilmesidir. Örneğin bir değişkenin 4 kategorisi olsun. Twoing kriteri i ve 4'ü bir süper sınıf ve 2 ve 3' ü de diğer bir süper sınıf olarak belirlemiş olsun. Bununla beraber kategoriler sıralı olduğundan 1 ve 4 kategorileri birleştirilemez çünkü bunlar bitişik kategoriler değillerdir. Sıralı Twoing indeksi bu durumu göz önüne aldığından i ve 4 gibi kategoriler bitişik olmadığından birleştirilemez.

En küçük kareli sapma (LSD) heterojenlik ölçüsü sürekli hedef değişkenleri için kullanılmaktadır. LSD ölçüsü R(t), t düğümü için basit (ağırlıklandırılmış) düğüm içi varyansıdır ve düğüm için risk tahminine eşittir. R(t)' nin formülü aşağıdaki şekildedir (Answer Tree 3.0 User's Guide,2001 s,195):

$$R(t) = \frac{1}{N_w(t)} \sum w_n f_n \left(y_1 - y(t)\right)^2$$

$N_w(t)$, t düğümündeki ağırlıklandırılımış durum sayısı, w_n'nin durumu için mevcut ise ağırlıklandırılımış değişken değeri, f_n mevcut ise frekans değişkeninin değerini, y_1 hedef değişkenin değerini ve $y(t)$ ise t düğümü için ağırlıklı ortalamayı göstermektedir.

Sonuçta elde edilen ağacın büyüklüğü, karmaşık budama sürecinin bir sonucudur. Çok büyük bir ağaç, uyumun üzerinde (overfitting) ve çok küçük ağaç, yetersiz tahmin gücüne sahip olacaktır. Ağaç yapısının hiyerarşik formu, CART gibi algoritmaları ağaç yapısına dayanmayan diğer sınıflandırma algoritmalarından açık bir şekilde ayırır.

Sınıflandırma Performansını Ölçme Yöntemleri

Regresyon modellemesinde sıkça kullanılan RMSE veya R2 ölçütleri sınıflandırma yöntemlerinin performansını ölçmek için uygun değildir. Bu tarz problemler için kullanılabilecek ölçütlerden bazıları şu şekildedir; Log-Kaybı (Log-Loss), Karmaşıklık Matrisi (Doğruluk), F1 skoru ve Eğri Altında Kalan Alan (Area Under Curve - AUC).

Log-Kaybı Logaritmik kayıp oldukça önemli bir performans ölçütüdür. Tahmin değerinin 0 ile 1 arasında bir olasılık değeri olduğu durumlarda sınıflandırma modelinin performansını ölçer. Mükemmel modelin Log-Kaybı değeri sıfırdır. Makine öğrenmesi modelimizin hedefi bu değeri 0'a yaklaştırmak olmalıdır. Örneğin, doğru etiket değeri 1 olan bir örnek için yaptığımız tahminin 0.023 gibi bir değer çıkması, yüksek Log-Kaybı olduğu ve kötü bir model kurulduğu anlamına gelmektedir.

Karmaşıklık (Hata) Matrisi İki veya çok sınıflı sınıflandırma probleminde, modelin doğruluğunu ölçmek için yaygın şekilde kullanılan ve anlaşılması basit bir matristir. Bu matrisi bir örnek üzerinden açıklamak adına hedef değişkenimiz için aşağıdaki iki etiketi kullandığımızı varsayalım:

- "0": Kişide test edilen hastalık bulunmamaktadır.
- "1": Kişide test edilen hastalık bulunmaktadır.

Bu etiketlerle sınıflandırma yapıldığında oluşacak karmaşıklık matrisi Şekil 1.6'da gösterilmiştir. Matriste yer alan bilgiler kısaca aşağıdaki gibidir:

*__Doğru Pozitif (DP):__ Verinin gerçek değerinin pozitif (1) ve tahmin edilen değerin de pozitif (1) olduğu durum.

*__Doğru Negatif (DN):__ Verinin gerçek değerinin negatif (0) ve tahmin edilen değerin de negatif (0) olduğu durum.

*__Yanlış Pozitif (YP):__ Verinin gerçek değerinin negatif (0) fakat tahmin edilen değerin pozitif (1) olduğu durum.

*__Yanlış Negatif (YN):__ Verinin gerçek değerinin pozitif (1) fakat tahmin edilen değerin negatif (0) olduğu durum.

Tablo 3.1: Karmaşıklık Matrisi

		Tahmin Edilen	
		Pozitif	Negatif
Gerçek	Pozitif	DP	YN
Durum	Negatif	YP	DN

Doğruluk (Accuracy) Sınıflandırma problemlerinde doğru tahminlerin bütün tahminlere oranıdır ve hesaplama formülü aşağıdaki gibidir.

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN}$$

Hedef değişken sınıflarının veri kümesinde dengeli dağıldığı durumlarda Doğruluk ölçütünü kullanmak mantıklıdır. Fakat birçok gerçek hayat probleminde, bu dengeyi yakalamak zordur. Bu dengenin olmadığı ve bir sınıf değerinin çoğunlukta olduğu durumlarda ACC ölçütünün kullanılması önerilmez.

Kesinlik (Precision) Kesinlik ölçütü, pozitif tahminde bulunduğumuz verilerin gerçekte hangi oranda pozitif olduğu sorusuna cevap verir ve hesaplama formülü aşağıdaki gibidir.

$$\text{Kesinlik} = \frac{DP}{DP + YP}$$

Hassaslık (Sensitivity) Aynı zamanda 'Doğru Pozitif Oranı (DPO)' olarak da adlandırılan bu ölçüt, gerçekte pozitif olanların ne kadarının doğru tahmin edildiğini ölçer ve hesaplama formülü aşağıdaki gibidir.

$$\text{Hassaslik} = \frac{DP}{DP + YN}$$

Kesinlik ölçütü, sınıflandırıcı performansını yanlış pozitifler ile açıklarken, yakalama ölçütü bu performansı yanlış negatifler ile açıklar. Problem tipimize ve hipotezimize bağlı olarak, hangi ölçütü iyileştirmeye çalışacağımızı belirlemeliyiz.

Belirlilik (Specifity) Aynı zamanda 'Doğru Negatif Oranı' olarak da adlandırılan bu ölçüt gerçekte negatif olanların ne kadarının doğru tahmin edildiğini ölçer ve hesaplama formülü aşağıdaki gibidir.

$$\text{Belirlilik} = \frac{DN}{DN + YP}$$

F1 Skoru Her seferinde kesinlik ve yakalama ölçütleri ile ayrı ayrı uğraşmak yerine ikisini birlikte temsil eden bir sınıflandırma performans ölçütü kullanmak mümkündür. Kesinlik ve yakalama ölçütlerinin ağırlıklı ortalamaları ile hesaplanan F1 Skorunun, doğruluk ölçütünden (ACC) daha kullanışlı olduğunu söylemek mümkündür. Hesaplama formülü aşağıdaki gibidir.

$$\text{F1 Skoru} = \frac{2 * \text{ Kesinlik} * \text{ Hassaslik}}{\text{Kesinlik } + \text{ Hassaslik}}$$

3.1.3 Sınama ve Geçerleme

Denetimli öğrenme uygulamalarında, kullanılan algoritmanın başarısının sınanması için elimizdeki veri kümesinin; eğitim ve test kümesi olarak ayrılması gerekmektedir. Ayırma işlemi çeşitli şekillerde yapılabilir.

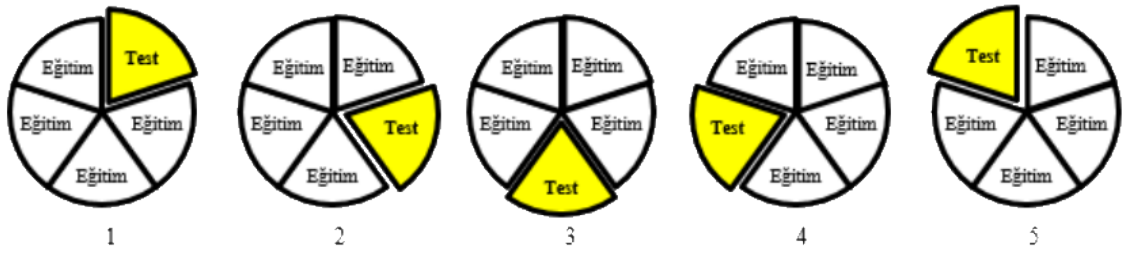
Sabit Ayırma

Klasik veri ayırma yöntemi olup, veri kümesi %80 eğitim ve %20 test olacak şekilde ayrılır (%66 – %34 gibi değerler de kullanılabilir). En büyük dezavantajı, verinin dağılımına göre bazı yanlılık durumlarının ve hataların oluşabilmesidir.

K-Katlamalı Çapraz Doğrulama

Bu yöntemde veri kümesi k sayısı kadar eşit parçaya bölünür ve her parçanın hem eğitim hem de test için kullanılması sağlanır. Bu sayede test kümesi seçimi yaparken oluşan yanlılık ortadan kaldırılmış olur. Aşağıdaki örnekte, k sayısının 5 seçildiğini düşünelim. Veri kümesi 5 eşit parçaya bölünür ve seçilen algoritma, eğitim ve test aşamalarını bu 5 parçayı da kullanarak gerçekleştirir.

- Model 1: (Parça 1 + Parça 2 + Parça 3 + Parça 4) eğitim kümesi, Parça 5 test kümesi
- Model 2: (Parça 1 + Parça 2 + Parça 3 + Parça 5) eğitim kümesi, Parça 4 test kümesi
- Model 3: (Parça 1 + Parça 2 + Parça 4 + Parça 5) eğitim kümesi, Parça 3 test kümesi
- Model 4: (Parça 1 + Parça 3 + Parça 4 + Parça 5) eğitim kümesi, Parça 2 test kümesi
- Model 5: (Parça 2 + Parça 3 + Parça 4 + Parça 5) eğitim kümesi, Parça 1 test kümesi



Şekil 3.2: K-Katlamalı Çapraz Doğrulama

3.2 Denetimsiz Öğrenme

Veri kümesindeki örneklerin herhangi bir etiketle (sınıfla) ayrıştırılmadığı ve etiket sayısının bilinmediği durumda kullanılan öğrenme yöntemidir (Yıldırım ve Birant, 2018). Literatürde en çok kullanılan denetimsiz öğrenme yöntemi, birbirine benzeyen örneklerin bir araya getirilmesini (gruplanmasını/kümelenmesini) amaçlayan Kümelemedir. Örneğin, bir alışveriş sitesini ele alalım. Siteyi kullanan müşterilerin farklı özelliklerini ve geçmişte yaptıkları, site içi gezinme, sepete atma, alışveriş yapma vb. bilgileri barındıran bir veri kümesinde, benzer davranışları gerçekleştiren müşterileri gruplamak mümkündür.

Kümeleme yöntemlerinin çoğu, veri örnekleri arasındaki uzaklık/yakınlık bilgisini kullanarak benzerlik bulma ve gruplama işlemini gerçekleştirmektedir. Verilerin benzerliğinin bulunmasında, basit bir uzaklık ölçütü (Öklid, Manhattan, Minkowski) kullanılabilir. Bunun dışında, yoğunluk ve komşuluk gibi özel hesaplamalar da benzerlik bulma için kullanılabilir.

Kümeleme işleminde kullanılan 5 farklı veri gruplama yaklaşımı bulunmaktadır.

1. Bölümleme Tabanlı (Partitioning) Yöntemler (K-Ortalamalar, K-Medoid, PAM, CLARA)
2. Hiyerarşik Yöntemler (AGNES, DIANA)
3. Yoğunluk Tabanlı Yöntemler (DBSCAN, OPTICS)
4. Model Tabanlı Yöntemler (EM)
5. Izgara Tabanlı Yöntemler (CLIQUE)

3.2.1 Sonuçları İyileştirme

Bir veri bilimi çalışması yaparken verimli ve yüksek performanslı bir model oluşturmak kolay değildir. Genellikle en büyük çaba, veri keşfi ve doğru makine öğrenmesi algoritmasını seçmek üzerine olsa da sonuçları iyileştirmek için yapacağımız çalışmalara projenin en başından itibaren başlamalıyız. Aynı zamanda şunu da unutmamak gerekir; bir makine öğrenmesi uygulaması yaparken, kullandığımız algoritma veya algoritmaların performansları bize makul derece iyi görünebilir, fakat elde ettiğimizin en iyi sonuç olup olmadığını tek seferde bilemeyiz. Dolayısıyla, birkaç yöntem kullanarak, elde ettiğimiz sonuçların iyileştirilmesi üzerine çalışmalar yapmamız sağlıklı olacaktır.

Algoritma Ayarı (Algorithm Tuning): Birçok gerçek hayat probleminde kullanılan algoritma, var olan (varsayılan) parametreleri ile en iyi sonucu vermeyebilir. En iyi sonucu elde etmek için belli bir kalıbı kullanmak yerine, çalışılan veri kümesine ve algoritmaya özel algoritma parametre ayarı yapılması önemlidir. Bu aşamada ilk yapılması gereken, algoritmanın parametreleri için birden fazla kombinasyonun denenerek en iyi alınan sonuçları saklamaktır. Daha sonra saklanan parametreler üzerinde küçük değişiklikler yapılarak daha iyi bir sonuç elde etmeye çalışılabilir.

Topluluk Yöntemleri (Ensemble Methods): Topluluk yöntemleri adından da anlaşılabileceği üzere, birden çok sonucu birlikte değerlendirmeyi ifade eder. Problemin yapısına göre tamamını çözmek için bir algoritma kullanmaktansa birçok aşamasında birden fazla algoritma kullanılabilir. Birden çok algoritmanın sonuçlarını birleştirerek alacağımız sonuç, yalnızca bir yöntem izleyerek alacağımız çözümden daha performanslı olacaktır.

Veri Miktarını Büyütme: Birçok deneme yapmamıza rağmen tahminlerimiz yüksek varyansa sahip olabilir. Her senaryoda veri miktarını büyütmek mümkün olmayabilir, bu tarz durumlarda diğer bir seçenek de eğitim veri kümesini büyütmek olabilir. Örnek sayısını yeni verilerle büyüterek algoritmanın daha iyi öğrenmesini sağlayabiliriz.

Bölüm 4

Uygulama

İzmir ili Torbalı ilçesinde plastik masa örtüsü üretimi gerçekleştiren SANEM PLASTİK adlı fabrikadan alınan 2018-2019 yılları arasındaki veriler ile sınıflandırma makine öğrenimi algoritmalarını kullanılarak oluşturulan modellere göre fabrikanın verimliliğini arttırmaktır.

Fabrikadan alınan veri "csv" formatında olduğu için RStudio üzerinden aşağıdaki kod ile data frame veri yapısında "veri" adı ile kaydedilmiştir. Daha sonra veri setindeki değişkenlerin yapıları Tablo 4.1'de verilmiştir.

```
library(readr)
veri <- read_delim("veri.csv", ",", escape_double = FALSE,
    col_types = cols(tecrube = col_factor(levels = c("0",
        "1")),gun = col_factor(levels = c("1",
        "2", "3", "4", "5", "6")), hafta = col_factor(levels =
        c("Hafta 01","Hafta 02", "Hafta 03", "Hafta 04", "Hafta 05",
        "Hafta 06", "Hafta 07", "Hafta 08", "Hafta 09", "Hafta 10",
        "Hafta 11", "Hafta 12", "Hafta 13", "Hafta 14", "Hafta 15",
        "Hafta 16", "Hafta 17", "Hafta 18", "Hafta 19","Hafta 20",
        "Hafta 21", "Hafta 22", "Hafta 23", "Hafta 24", "Hafta 25",
        "Hafta 26", "Hafta 27", "Hafta 28", "Hafta 29","Hafta 30",
        "Hafta 31", "Hafta 32", "Hafta 33", "Hafta 34", "Hafta 35",
        "Hafta 36", "Hafta 37", "Hafta 38", "Hafta 39","Hafta 40",
        "Hafta 41", "Hafta 42", "Hafta 43", "Hafta 44", "Hafta 45",
        "Hafta 46")), tarih = col_date(format = "%Y-%m-%d")),
    locale = locale(decimal_mark = "."),
    trim_ws = TRUE)
```

Tablo 4.1: Veri Seti Değişkenleri

Değişkenler	Tür	Kategorik
OEE *(Hedef Değişken)*	Sürekli	
Hafta	Kategorik	1'den 46'ya kadar
Üretim	Sürekli	
Tarih	Tarih	Ay- Gün- Yıl
Saat	Sürekli	1 – 24
Kayıp metre	Sürekli	
Gün	Kategorik	1: Pazartesi 2: Salı 3: Çarşamba 4: Perşembe 5: Cuma 6: Cumartesi 7: Pazar
On Yıl Tecrübe (Simülasyon)	Kategorik	0: On yıl tecrübe altı 1: On yıl tecrübe ve üstü

Tablo 4.1'de görüldüğü üzere OEE değişkeni bağımlı değişken iken diğer değişkenler bağımsız olarak ele alınmıştır. OEE, üretim, saat, kayıp metre, değişkenleri sürekli, hafta, gün ve tecrübe değişkenleri kategorik değişkenlerdir. Tüm bu değişkenlerin ilk 6 satırının görüntüsü ve özet istatistikleri aşağıda verilmiştir.

```
head(veri)
```

```
# A tibble: 6 x 8
  tarih      hafta    uretim   OEE  saat kayip_metre gun   tecrube
  <date>     <fct>     <dbl> <dbl> <dbl>       <dbl> <fct> <fct>
1 2019-01-10 Hafta 02  72810    75   5.5       21600 4     0
2 2019-01-11 Hafta 02  88195    92   2          7802 5     1
3 2019-01-12 Hafta 02  84400    85   3.6       13846 6     1
4 2019-01-14 Hafta 03  11800    36   4.1       16118 1     0
5 2019-01-15 Hafta 03  85780    89   2.7       10473 2     1
6 2019-01-16 Hafta 03  89530    94   1.5        5795 3     1
```

```
summary(veri)
```

```
   tarih                hafta         uretim          OEE        
Min.   :2019-01-10   Hafta 16:  6   Min.   : 4750   Min.   : 29.00  
1st Qu.:2019-03-28   Hafta 22:  6   1st Qu.:49889   1st Qu.: 82.00  
Median :2019-06-15   Hafta 27:  6   Median :80145   Median : 89.00  
Mean   :2019-06-16   Hafta 34:  6   Mean   :68018   Mean   : 85.15  
3rd Qu.:2019-09-10   Hafta 35:  6   3rd Qu.:86760   3rd Qu.: 93.75  
Max.   :2019-11-13   Hafta 37:  6   Max.   :94680   Max.   :100.00  
                     (Other) :162                                   
     saat          kayip_metre    gun    tecrube
Min.   : 0.200   Min.   : 1021   1:27   0: 83  
1st Qu.: 1.000   1st Qu.: 3806   2:36   1:115  
Median : 1.950   Median : 6943   3:38          
Mean   : 2.576   Mean   : 8728   4:38          
3rd Qu.: 3.000   3rd Qu.:11651   5:34          
Max.   :23.400   Max.   :36471   6:25          
```

Modellerin amacı bağımsız değişkenlerin, OEE puanına nasıl etkileyeceğini araştırmaktır. Model için sürekli değişkenlerimiz, ortalamaya göre bölünüp 0-1 olarak kategorik değişken haline getirilmiştir.

```
veri$uretim[veri$uretim<mean(veri$uretim)]=0
veri$uretim[!(veri$uretim<mean(veri$uretim))]=1
veri$uretim=as.factor(veri$uretim)
```

```
veri$kayip_metre[veri$kayip_metre<mean(veri$kayip_metre)]=0
veri$kayip_metre[!(veri$kayip_metre<mean(veri$kayip_metre))]=1
veri$kayip_metre=as.factor(veri$kayip_metre)
```

```
veri$saat[veri$saat<mean(veri$saat)]=0
veri$saat[!(veri$saat<mean(veri$saat))]=1
veri$saat=as.factor(veri$saat)
```

Daha önce belirtildiği üzere hedef değişkenimiz OEE değişkenidir. OEE, "Overall Equipment Effectiveness" kısaltmasıdır. Türkçe çevirisi Toplam Ekipman Etkinliği'dir. OEE bütün ekipmanların ne ölçüde kullanıldığına işaret eden bir TPM(Total Productive Maintenance) hesabıdır. Arızalar, ekipman ayarları, duruşlar, çalışma hızındaki azalmalar, ıskartalar ve yeniden işlem gibi kayıplar üzerine düşer. Amacı; şirketlerin eldeki makine ve ekipmanların performanslarının arttırılmasına odaklanmaktır. Gerekli görüldüğü ölçüde başka önemli ölçütlerin eklenebilmesi esnekliğine sahip olmasıyla beraber genel olarak 3 önemli değişkeni bir arada hesap eder:

OEE [%]= Kullanılabilirlik Oranı x Performans Oranı x Kalite Oranı

- **Kullanılabilirlik Oranı** [%]**:** Ekipmana ait sebeplerden (arıza, ayarlamadan kaynaklanan duruş süresi vs.) kaynaklanan kullanılabilirlik miktarını gösterir.
- **Performans Oranı** [%]**:** Çalışma hızlarında tasarımla belirlenmiş hızlara göre düşüşleri ve birkaç saniyelik duruşları hesap eder.
- **Kalite Oranı** [%]**:** Toplam işlenen parçaların ıskarta ve yeniden işlem kayıplarının yüzdesidir. [`https://lean.org.tr/oee-nedir/`, Can YÜKSELEN]

Uygulanabilen en iyi OEE değeri %85 olarak kabul edilmektedir ve bu seviyedeki şirketler **"World Class"** üretim yapan şirketler olarak anılırlar. Dünya standartlarında olan şirketler OEE performanslarını yükselterek, kapasite ihtiyaçlarını karşılamaya çalışırlar. Böylece kendilerini ek yatırım, ek alan kullanımı, ek işçilik, fazla stok, fazla enerji kullanımı, kalitesizlik ve finansman maliyetlerinden korurlar. Bu nedenle hedef değişken 85 puan ve üstü ile 85 puan altı olarak ayırdık ve 1-0 değerleri ile kategorik değişken olarak değiştirildi.

```
veri$OEE[veri$OEE<85]=0
veri$OEE[!veri$OEE<85]=1
veri$OEE=as.factor(veri$OEE)
```

Yeni durumda özet istatistikler şu şekildedir:

```
summary(veri[,-1])
```

```
     hafta      uretim   OEE      saat     kayip_metre gun     tecrube
Hafta 16:  6   0: 72    0: 68    0:132    0:124       1:27    0: 83
Hafta 22:  6   1:126    1:130    1: 66    1: 74       2:36    1:115
Hafta 27:  6                                          3:38
Hafta 34:  6                                          4:38
Hafta 35:  6                                          5:34
Hafta 37:  6                                          6:25
(Other) :162
```

Daha sonra veri %80 eğitim ve %20 test olarak iki parçaya aşağıdaki gibi ayrılmıştır.

```
set.seed(1)
r=as.logical(rbinom(length(veri$OEE),1,prob = 0.8))
train=veri[r,-c(1,2)]
test=veri[!r,-c(1,2)]
```

Lojistik Regresyon Modeli (Full Model)

Veri setindeki tüm değişkenlerle train verisi üzerinde lojistik regresyon modeli kurulmuştur. Lojistik regresyon için yapılan full modelin R çıktısı aşağıda verilmektedir.

```
full.model = glm(OEE ~. - tecrube ,family=binomial, data=train)
summary(full.model)
```

```
Call:
glm(formula = OEE ~ . - tecrube, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7868  -0.4047   0.2040   0.6068   1.4250

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.14138    0.54440  -0.260  0.79509
uretim1       3.90538    0.80857   4.830 1.37e-06 ***
saat1        -2.21958    0.86006  -2.581  0.00986 **
kayip_metre1 -1.90456    1.00137  -1.902  0.05718 .
gun2          0.09838    0.73493   0.134  0.89351
gun3         -0.20531    0.75528  -0.272  0.78575
gun4         -0.03542    0.73268  -0.048  0.96144
gun5          1.07417    0.82050   1.309  0.19048
gun6          1.56849    0.87277   1.797  0.07231 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 214.49  on 162  degrees of freedom
Residual deviance: 120.85  on 154  degrees of freedom
AIC: 138.85

Number of Fisher Scoring iterations: 6
```

Çıktıda görüldüğü üzere üretim, saat anlamlı çıkmıştır. Kayıp metre değişkeni 0.05 sınırında olduğundan '.' ile işaretlenmiştir. Yapılan Korelasyon analizine göre kayıp metre ve saat değişkeni ile ilişkili bulunmuştur (0.48). Bu yüzden kayıp metrenin modele alınmamasına karar verilmiştir.

Lojistik Regresyon Modeli (En İyi Model)

En iyi modeli oluşturmak için stepwise yöntemi kullanılmıştır. Bu yöntemde modele hem değişken ekleme hem çıkarma işlemlerini uygulaması için direction = "both" olarak tanımlanmıştır. En iyi modelin modelin R çıktısı aşağıda verilmektedir.

```
best.model <- stepAIC(glm(OEE ~. - tecrube -kayip_metre ,family=binomial, data=train), d
summary(best.model)
```

```
Call:
glm(formula = OEE ~ uretim + saat, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5993  -0.2757   0.2635   0.6622   1.2583

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.2805     0.3149   0.891    0.373
uretim1       3.0630     0.6392   4.792 1.65e-06 ***
saat1        -3.5317     0.6397  -5.521 3.38e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 214.49  on 162  degrees of freedom
Residual deviance: 131.67  on 160  degrees of freedom
AIC: 137.67

Number of Fisher Scoring iterations: 6
```

Çıktıda görüldüğü üzere en iyi modelde üretim, saat değişkenleri %95 anlamlılık düzeyinde istatistiksel olarak anlamlı çıkmıştır. Daha sonra OR, CI ve Wald testine bakılmıştır.

```
wald.test(b=coef(best.model),Sigma=vcov(best.model),Terms=1:2)
```

```
Wald test:
----------

Chi-squared test:
X2 = 31.2, df = 2, P(> X2) = 1.6e-07
```

Kategorik değişkenlerle kurulan lojistik regresyon en iyi modelindeki Wald değerlerine bakıldığında, tüm p değerleri 0.05'den küçük olduğu için katsayılar %95 güven seviyesinde istatistiksel olarak anlamlı bulunmuştur.

```
exp(confint(best.model))
```

```
Waiting for profiling to be done...
```

```
                 2.5 %      97.5 %
(Intercept) 0.715917230  2.48285748
uretim1     7.032365088 93.58095767
saat1       0.006676091  0.08888818
```

Ayrıca OR ve güven aralıklarına bakıldığında, OR'a ait güven aralıklarının 1'i içermediği görülmektedir. Bu yüzden OR'ler anlamlıdır ve yorumlanabilir.

```
exp(coef(best.model))
```

```
(Intercept)     uretim1       saat1
 1.32378898 21.39165960  0.02925425
```

Üretimi yüksek olanların, üretimi düşük olanlara göre Toplam Ekipman Etkinliğinin (OEE) yüzdesinin yüksek olma şansı yaklaşık olarak 21 katıdır. Saat değişkeninde ise saati az olanların, yüksek olanlara göre OEE yüzdesinin yüksek olma şansı ise 1/0.29 oranından yaklaşık olarak 3 katı olduğu elde edilmektedir.

Train verisinde oluşturduğumuz model, test verisinde sınıflandırma performansını ölçme yöntemleri ile şu şekilde test edilmiştir.

```
#Tahmin Değerleri
pred = predict(best.model, test, type = "response")
print(round(pred,2))
```

```
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
0.04 0.97 0.97 0.97 0.97 0.57 0.97 0.97 0.04 0.04 0.45 0.97 0.57 0.97 0.45 0.57
  17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
0.57 0.57 0.57 0.04 0.57 0.97 0.45 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
  33   34   35
0.97 0.97 0.57
```

Tahmin değerleri logistic regresyon modelinde sigmoid fonksiyon gereği 0 ile 1 arasında değerler almaktadır. Bu değerleri kategorik hale getirdiğimiz OEE değişkenindeki 0 ile 1 değerleri gibi tam hale çevirmek için şu kod kullanılmıştır.

```
pred = ifelse(pred<0.5,0,1)
print(pred)
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 0  1  1  1  1  1  1  1  0  0  0  1  1  1  0  1  1  1  1  0  1  1  0  1  1  1
27 28 29 30 31 32 33 34 35
 1  1  1  1  1  1  1  1  1
```

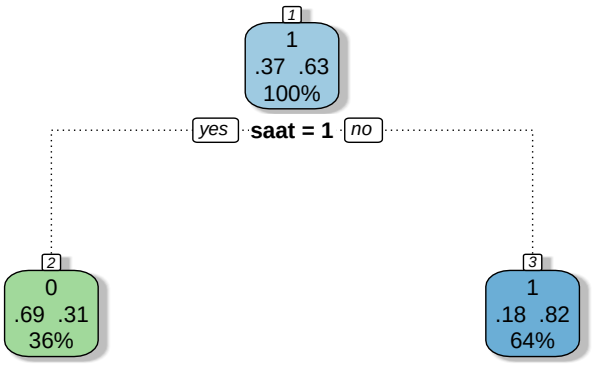
```
En İyi Model için OEE Karmaşıklık Matrisi

          Reference
Prediction  <85  >=85
       <85    5     2
      >=85    3    25
```

Doğruluk	Kesinlik	Hassaslık	Belirlilik	F1 Skoru
0.857	0.625	0.926	0.714	0.667

CART Algoritması (En İyi Model)

```
cart=rpart(formula = OEE ~ uretim+saat, data = train, method = "class")
fancyRpartPlot(cart,sub = "")
```



Şekil 4.1: CART (En İyi Model)

CART ağacımız Gini katsayısına göre hesaplanıp, ilk olarak saat kategorik değişkenine ayrılmıştır. Bu durumda OEE'nin yorumlanmasında en önemli değişken olarak saat değişkeni olduğu söylenebilir. Buna göre saati ortalama veya ortalamadan yüksek olanlar, sol tarafa ayrılmıştır. 163 birimlik "train" verisi üzerinde hesaplandığı için bu düğüme %36 oranla 59 birim düşmüştür. 59 birimin %69'u yani 41 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 18 tanesi ise OEE'si 1 olan sınıfta yer almıştır. Saati ortalamadan az olanlar ise, sağ tarafa ayrılmıştır. 163 birimin %64 oranla 104 birim bu düğüme düşmüştür. 104 birimin %18'i yani 19 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 85 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

```
En İyi Model için OEE Karmaşıklık Matrisi

          Reference
Prediction  <85  >=85
       <85    5     2
      >=85    3    25
```

Doğruluk	Kesinlik	Hassaslık	Belirlilik	F1 Skoru
0.857	0.625	0.926	0.714	0.667

Yeni Değişken

Fabrika verileri ile kurulan lojistik regresyon modelinde saat ve üretim değişkenleri modelde anlamlı çıkmıştır. CART modelinde ise en önemli değişkenin saat olduğu tespit edilmiştir. Daha sonra CART modelinde ağaç derinliği kontrol edildiğinde 2. düğümün üretim değişkenine ayrıldığı gözlemlenmiştir. Fakat üretim değişkeni tek başına yeterli ayrıştırmayı yapamadığından ağaçta kırpılmış olduğu fark edilmiştir.

Modellerde değişken sayısını artırıp modelin nasıl değiştiğini görmek için simülasyon çalışması ile makineyi çalıştıran kişinin tecrübesi (10 yıldan az&10 yıl ve daha fazla olarak belirlenmiş kategorik değişken) verilerimize eklenmiştir. Bu yeni değişkene göre yeni modeller kurulmuştur.

Lojistik Regresyon Modeli (Tecrübe, Üretim, Saat (TÜS))

```r
tus.model = glm(OEE ~tecrube +uretim+ saat ,family=binomial, data=train)
summary(tus.model)
```

```
Call:
glm(formula = OEE ~ tecrube + uretim + saat, family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1667  -0.3807   0.1177   0.5445   1.9852

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.2441     0.4895  -2.541    0.011 *
tecrube1      2.7232     0.5543   4.913 8.98e-07 ***
uretim1       3.4907     0.7087   4.926 8.41e-07 ***
saat1        -4.0670     0.7303  -5.569 2.56e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 214.49  on 162  degrees of freedom
Residual deviance: 100.22  on 159  degrees of freedom
AIC: 108.22

Number of Fisher Scoring iterations: 6
```

Burada en uygun modelde tecrübe, üretim, saat değişkenleri %95 anlamlılık düzeyinde istatistiksel olarak anlamlı çıkmıştır. Daha sonra OR, CI ve Wald testine bakılmıştır.

```
wald.test(b=coef(tus.model),Sigma=vcov(tus.model),Terms=1:2)
```

```
Wald test:
----------

Chi-squared test:
X2 = 24.5, df = 2, P(> X2) = 4.8e-06
```

Kategorik değişkenlerle kurulan lojistik regresyon en iyi modelindeki Wald değerlerine bakıldığında, tüm p değerleri 0.05'den küçük olduğu için katsayılar %95 güven seviyesinde istatistiksel olarak anlamlı bulunmuştur.

```
exp(confint(tus.model))
```

```
Waiting for profiling to be done...

                 2.5 %       97.5 %
(Intercept) 0.102099647   0.71419154
tecrube1    5.466755363  48.97582068
uretim1     9.408205782 161.53474388
saat1       0.003329586   0.06180391
```

Ayrıca OR ve güven aralıklarına bakıldığında, OR'a ait güven aralıklarının 1'i içermediği görülmektedir. Bu yüzden OR'ler anlamlıdır ve yorumlanabilir.

```
exp(coef(tus.model))
```

```
(Intercept)    tecrube1     uretim1       saat1
 0.28821447 15.22833475 32.81030772  0.01712815
```

10 Yıllık Tecrübesi olanların, olmayanlara göre OEE yüzdesinin yüksek olma şansı yaklaşık olarak 15 kattır. Üretimi yüksek olanların, üretimi düşük olanlara göre OEE yüzdesinin yüksek olma şansı yaklaşık olarak 33 kattır. Saati az olanların saati yüksek olanlara göre OEE yüzdesinin yüksek olma şansı ise 1/0.17 yaklaşık olarak 6 kattır.

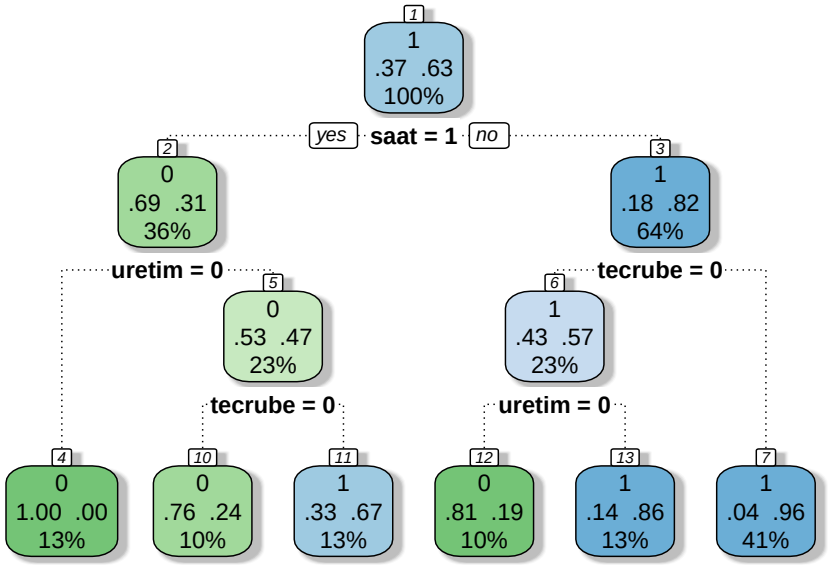
```
TÜS Model için OEE Karmaşıklık Matrisi

          Reference
Prediction  <85  >=85
       <85    5     2
      >=85    3    25
```

Doğruluk	Kesinlik	Hassaslık	Belirlilik	F1 Skoru
0.886	0.75	0.926	0.75	0.75

CART Algoritması (Tecrübe, Üretim, Saat, (TÜS))

```
cart=rpart(formula = OEE ~ tecrube +uretim+ saat, data = train, method = "class")
fancyRpartPlot(cart,sub = "")
```



Şekil 4.2: CART (TÜS)

Buna göre saati ortalama veya ortalamadan yüksek olanlar, sol tarafa ayrılıp, 2. Düğümde yer almıştır. 163 birimlik "train" verisi üzerinde hesaplandığı için bu 2. düğüme %36 oranla 59 birim düşmüştür. 59 birimin %69'u yani 41 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 18 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

2. düğüm daha sonra üretim değişkenine göre ikiye ayrılmıştır. Üretimi ortalama ya da ortalamadan yüksek olanlar sağ tarafa ayrılıp 5. düğümü oluşturmuşlardır. 5. düğüme toplam verinin %23'üne denk gelen 38 birim düşmüştür. Bu 38 birimin %53'ü

yani 20 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 18 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

2. düğümde üretimi ortalamadan az olanlar sol tarafa ayrılıp 4. düğümü oluşturmuşlardır. 4. düğüme toplam verinin %13'üne denk gelen 21 birim düşmüştür. Bu 21 birimin %100'ü yani tamamı OEE'si 0 olan sınıfta yer almıştır. OEE'si 1 olan sınıfta yer alan birim yoktur.

5. düğüm daha sonra tecrübe değişkenine göre ikiye ayrılmıştır. On yıllık tecrübesi olanlar sol tarafa ayrılıp 10. düğümü, olmayanlar sağ tarafa ayrılıp 11. düğümü oluşturmuşlardır. 10. düğüme toplam verinin %10'una denk gelen 16 birim düşmüştür. Bu 16 birimin %76'sı yani 12 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 4 tanesi ise OEE'si 1 olan sınıfta yer almıştır. 11. düğüme ise toplam verinin %13'üne denk gelen 21 birim düşmüştür. Bu 21 birimin %33'ü yani 7 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 14 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

Saati ortalamadan az olanlar ise, sağ tarafa ayrılmıştır. 163 birimin %64 oranla 104 birim bu düğüme yani 3. düğüme düşmüştür. 104 birimin %18'i yani 19 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 85 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

3. Düğüm daha sonra tecrübe değişkenine göre ayrılmıştır. On yıllık tecrübesi olmayanlar sol tarafa ayrılıp 6. düğümü oluşturmuştur. 6. düğüme toplam verinin %23'üne denk gelen 38 birim düşmüştür. Bu 38 birimin %43'ü yani 16 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 22 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

3. düğümdeki on yıllık tecrübesi olanlar sağ tarafa ayrılıp 7. düğümü oluşturmuşlardır. 7. düğüme toplam verinin %41'ine denk gelen 67 birim düşmüştür. Bu 67 birimin %4'ü yani 3 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 64 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

6. düğüm daha sonra üretim değişkenine göre ikiye ayrılmıştır. Üretimi az olanlar sol tarafa ayrılıp 12. düğümü, çok olanlar sağ tarafa ayrılıp 13. düğümü oluşturmuşlardır. 12. düğüme toplam verinin %10'una denk gelen 16 birim düşmüştür. Bu 16 birimin %81'i yani 13 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 3 tanesi ise OEE'si 1 olan sınıfta yer almıştır. 13. düğüme ise toplam verinin %13'üne denk gelen 21 birim düşmüştür. Bu 21 birimin %14'ü yani 3 tanesi OEE'si 0 olan sınıfta yer almıştır. Kalan 18 tanesi ise OEE'si 1 olan sınıfta yer almıştır.

```
TÜS Model için OEE Karmaşıklık Matrisi

          Reference
Prediction  <85  >=85
       <85    6     2
      >=85    2    25
```

Doğruluk	Kesinlik	Hassaslık	Belirlilik	F1 Skoru
0.886	0.75	0.926	0.75	0.75

Sonuç

İzmir ili Torbalı ilçesinde plastik masa örtüsü üretimi gerçekleştiren SANEM Plastik A.Ş. adlı fabrikadan 2018-2019 yılları arasında alınan 198 veri ile makine öğrenimi algoritmalarını kullanarak saat ve üretim ve değişkenlerinin OEE puanı üzerindeki etkileri, lojistik regresyon ve CART algoritmaları üzerinden modeller kurulmuştur. Kurulan lojistik regresyon modelinde saat ve üretim değişkenleri modelde anlamlı çıkmıştır. CART modelinde ise en önemli değişkenin üretim olduğu tespit edilmiştir.

Modellerimizin iki büyük kısıtlaması vardır. Biri değişken sayısı diğeri ise örneklem sayısının azlığıdır. Bu amaçla modellemede değişken sayısını artırıp modelin nasıl değiştiğini görmek için verilerimize simülasyon çalışması ile makineyi çalıştıran kişinin tecrübesi(10 yıldan az&10 yıl ve daha fazla olarak belirlenmiş kategorik değişken) eklenmiştir.

Son eklenen veriler ile lojistik regresyon modelimizde saat, üretim ve tecrübe değişkenlerinin anlamlı, CART modelinde ise en önemli değişkenin üretim olduğu, sonra tecrübe en son olarak da saat olduğu bulunmuştur.

Bu çalışma ile üretim yapan fabrikaların verisinde makine öğrenimi kapsamında yer alan sınıflandırma modellerinden lojistik regresyon ve CART modellerinin uygulanabilirliği tespit edilmiştir. Bu tip verilerde değişik sınıflandırma algoritmaları da kullanılabilir. Ayrıca bu algoritmaların performansları da karşılaştırılabilir.

Fabrikadaki uzmanlarla yapılan ortak çalışmalarla bu tespitin farklı veri türlerinde ve farklı modellerle yapılması planlanmalıdır. Böylece sanayi-üniversite işbirliği gerçekleşmesi sağlanabilir.

Kaynaklar

Angel, E. (2000). *Interactive computer graphics : A top-down approach with opengl*. Boston, MA: Addison Wesley Longman.

Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with quicktime*. Boston, MA: Wesley Addison Longman.

Angel, E. (2001b). *Test second book by angel*. Boston, MA: Wesley Addison Longman.

Wong, E. (1999). *Artistic Rendering of Portrait Photographs* (Master's thesis). Cornell University.

Atalay M., Çelik E., 2017, Artificial İntelligence And Machine Learning Applications İn Big Data Analysis

Özkaya A., 2012, Makine Öğrenmesi İle Ürün Sınıflandırma İncelemesi

KALAYCI E., 2018, Comparison of machine learning techniques for classification of phishing web sites

Abbas G., Shirali M., Moloud V., Amal S., 2018, Predicting the outcome of occupational accidents by CART and CHAID methods at a steel factory in Iran

Yan-yan S., Ying L., 2015, Decision tree methods: applications for classification and prediction

Antipov E., Pokryshevskaya E., 2009, Applying CHAID for logistic regression diagnostics and classification accuracy improvement