



**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**DOKUMA İŞLETMELERİNDEKİ ÇÖZGÜ MAKİNELERİ İÇİN ÜRETİM
SÜRESİ TAHMİNLEME**

YÜKSEK LİSANS TEZİ

Müzeyyen GÖKSEL

Akıllı Sistemler Mühendisliği Anabilim Dalı

Tezli Yüksek Lisans Programı

TEMMUZ 2023

**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**DOKUMA İŞLETMELERİNDEKİ ÇÖZGÜ MAKİNELERİ İÇİN ÜRETİM
SÜRESİ TAHMİNLEME**

YÜKSEK LİSANS TEZİ

**Müzeyyen GÖKSEL
(20324814008)
ORCID: 0000-0002-5131-8874**

Akıllı Sistemler Mühendisliği Anabilim Dalı

Tezli Yüksek Lisans Programı

**Tez Danışmanı: Dr. Öğr. Üyesi Yunus DEMİR
ORCID: 0000-0003-3868-1860**

Temmuz 2023

BTÜ, Lisansüstü Eğitim Enstitüsü'nün 20324814008 numaralı Yüksek Lisans Öğrencisi Müzeyyen GÖKSEL, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “DOKUMA İŞLETMELERİNDEKİ ÇÖZGÜ MAKİNELERİ İÇİN ÜRETİM SÜRESİ TAHMİNLEME” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Dr. Öğr. Üyesi. Yunus DEMİR**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Aytaç YILDIZ**
Bursa Teknik Üniversitesi

Dr.Öğr.Üyesi Mete YAĞANOĞLU
Atatürk Üniversitesi

Teslim Tarihi **: Haziran 2023**
Savunma Tarihi **: Temmuz 2023**



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: Müzeyyen GÖKSEL

İmzası :

(Faint, stylized signature watermark)



Eşime ve çocuklarıma,

ÖNSÖZ

Bu çalışmada, dokuma işletmelerindeki çözgü leventi beklemeden dolayı dokuma tezgahlarının beklememelerini sağlamak amacıyla çözgü makineleri için üretim süresi tahminleme çalışması yapıldı. 12 farklı denetimli makine öğrenmesi algoritmalarının performansları incelenip değerlendirilmiştir.

Çalışmalarım süresince bana sürekli destek olan, yol gösteren, emeğini esirgemeyen danışman hocam Sayın Dr. Öğretim Üyesi Yunus DEMİR’e saygılarımı ve teşekkürlerimi sunarım.

Ayrıca bana çalışma imkânı veren Küçükçalık Tekstil A.Ş. ailesine de teşekkürlerimi sunarım.

Haziran 2023

Müzeyyen GÖKSEL
Dış Tedarik Kalite Müdürü

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xii
SUMMARY	xiv
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	3
3. MALZEME VE YÖNTEM	6
3.1 Çözümleri Hazırlama Süreci.....	6
3.1.1 Çözümleri nedir?	6
3.1.2 Atkı nedir?.....	7
3.1.3 Çözümleri hazırlama nedir?	7
3.1.4 Çözümleri hazırlamada dikkat edilecek noktalar	8
3.1.5 Çözümleri makineleri	8
3.1.6 Numune çözümü makinesi	9
3.1.7 Düz/Seri çözümü makinesi	9
3.1.8 Konik çözümü makinesi.....	9
3.1.9 İşbağ işlemi	10
3.1.10 Takım tahar nedir?	10
3.1.11 Çözümleri makinesi iş akışı.....	11
3.2 Makine Öğrenmesi Temel Kavramlar	12
3.2.1 Makine öğrenmesi teknikleri	13
3.2.1.1 Denetimli öğrenme.....	13
3.2.1.2 Denetimsiz öğrenme.....	13
3.2.1.3 Pekiştirmeli öğrenme	14
3.2.2 Makine öğrenmesi regresyon algoritmaları	15
3.2.2.1 Doğrusal regresyon algoritmaları.....	15
3.2.2.2 Doğrusal olmayan regresyon algoritmaları.....	16
3.3 Uygulama Veri Seti	17
3.4 Google CoLab Hakkında Genel Bilgiler	18
3.5 Uygulama Adımları.....	19
3.5.1 Gerekli kütüphanelerin dağıtımını projeye dahil edilmesi.....	19
3.5.1.1 Numpy.....	19
3.5.1.2 Pandas	19
3.5.1.3 Scikit-learn (sklearn).....	20
3.5.1.4 Seaborn.....	20
3.5.1.5 Statsmodels	20
3.5.1.6 Matplotlib.....	21
3.5.2 Verinin çekilmesi	21

3.5.3 Özellik seçimi	22
3.5.4 Veri ön işleme	23
3.5.5 Model seçimi ve değ erlendirmesi	25
3.5.6 Hiperparametre ayarı.....	26
4. BULGULAR VE TARTIŞMA.....	29
5. SONUÇ VE ÖNERİLER.....	31
KAYNAKLAR	32
ÖZGEÇMİŞ	40



KISALTMALAR

ÇDR	: Çoklu Doğrusal Regresyon
ÇKA	: Çok katmanlı algılayıcılar
KEKR	: Kısmi En Küçük Kareler Regresyonu
KNN	: k Nearest Neighbors
LR	: Lasso Regresyonu
MLP	: Multilayer perceptron
MLR	: Multiple Linear Regression
PCR	: Principal Component Regression
PLCR	: Partial Least Squares Regression
RA	: Regresyon Ağaçları
RF	: Random Forest
RO	: Rastsal Orman
RT	: Regression Trees
TBR	: Temel Bileşenler Regresyonu

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Tahmin parametreleri.....	22
Çizelge 3.2 : Performans metrikleri	26
Çizelge 3.3 : Seçilen algoritmalara ait hiper-parametre değerleri.....	28
Çizelge 4.1: Farklı makine öğrenimi modelleri arasında performans karşılaştırma ..	29



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1: Çözümlü ipliği.	6
Şekil 3.2: Çözümlü çekme işlemi.	6
Şekil 3.3: Kumaştaki atkı ve çözümü görseli.	7
Şekil 3.4: Çözümlü leventi.	8
Şekil 3.5: Seri çözümü makinesi.	9
Şekil 3.6: Konik çözümü makinesi.	10
Şekil 3.7: İşbağ işlemi.	10
Şekil 3.8: Tahar işlemi.	11
Şekil 3.9: Çözümlü işlemi iş akış şeması.	12
Şekil 3.10: Denetimli öğrenme (Singh, 2016).	13
Şekil 3.11: Denetimsiz öğrenme (Singh, 2016).	14
Şekil 3.12: Kategorik değişkene ait frekans bilgisi.	18
Şekil 3.13: Veri setinin sayısal değişkenlerinin tanımlayıcı istatistikleri.	18
Şekil 3.14: Kütüphane yükleme kodları.	19
Şekil 3.15: Veri çekmek için kullanılan kod.	21
Şekil 3.16: Değişken türleri.	24
Şekil 3.17: One hot encoder'ın genel işleyişi.	25
Şekil 3.18: Label encoder genel işleyişi.	25

DOKUMA İŞLETMELERİNDE ÇÖZGÜ MAKİNELERİ İÇİN ÜRETİM SÜRESİ TAHMİNLEME

ÖZET

İşlem süresinin; toplam gecikme, toplam akış süresi, tamamlama süresi gibi planlama sürecinin temel performans göstergeleri üzerinde önemli bir etkisi olduğundan, doğru belirlenmesi sürecin etkin yönetimi bağlamında anahtar role sahiptir. Geleneksel olarak işlem süresi, zaman etüdü veya basit hesaplamalar yoluyla belirlenir ve planlama öncesi kesin bir şekilde önceden bilindiği ve de genellikle sabit bir değer olduğu varsayılır.

Bazı durumlarda, işlem süresinin kendisi önceden bilinmese de hesaplanmasında kullanılan fonksiyon önceden bilinmektedir. Bu tür durumlar genellikle kimyasal ve metalürjik işlemlerde ortaya çıkar. Fiili üretim sürecinde bazı beklenmedik durumlar nedeniyle işlem süreleri tam olarak kaydedilememektedir. İşlem sürelerinde belirsizlik olarak adlandırılan bu durumun literatürde farklı şekillerde ele alındığı görülmektedir. Bunlardan ilki işlem süresinin bulanıklaştırılmasıdır. Belirsiz işlem süresini modellemek için kullanılan ikinci bir yaklaşım, aralık sayılarıdır (*interval numbers*). Rastgele veya stokastik işlem süresi, işlem süresindeki belirsizliği ifade etmenin diğer bir yoludur. Bu yaklaşımda, her işin işlem süresi, bilinen bir olasılık dağılımına sahip rastgele bir değerdir.

Bu çalışmada, işlem süresindeki belirsizlik, makine arızası, elektrik kesintisi gibi olasılıksal değişen faktörlerden değil, önceden bilinen sabit değerlere sahip faktörlerin bir kombinasyonundan kaynaklanmaktadır. Tekstil üretim süreçlerinin önemli bileşenlerinden olan çözgü hazırlama operasyonundan esinlenilen bu çalışmada, çözgü hazırlama operasyonunun işlem süresini tahmin etmek için denetimli makine öğrenmesi yaklaşımları kullanılmıştır. Dokuma makinelerinde kullanılmak üzere çözgü ipliklerinin leventlere sarıldığı ilgili süreçte bir leventin hazırlanması için gereken işlem süresi birçok parametreye bağlı olarak değişmektedir. Öncelikle konunun uzmanları ile görüşmeler yapılmış ve neticesinde işlem süresini etkileyebilecek 11 öznel belirlenmiştir. Daha sonra çözgü hazırlama işlem süresinin tahmin edilmesi için 6 doğrusal ve 6 doğrusal olmayan toplam 12 farklı makine öğrenmesi algoritması test edilmiştir. Deneysel çalışma işletmenin ERP sisteminde çekilen gerçek hayat verileri ile yapılmıştır. 12 farklı denetimli makine öğrenmesi algoritma hem eğitim hem de test veri seti ile koşulmuş ve sonuçlar karşılaştırmalı bir şekilde sunulmuştur. Hem eğitim/ayar süresi bakımında hem de tahmin doğruluğu bakımından boosting algoritmalarının üstünlük sağladığı gözlemlenmiştir.

Anahtar kelimeler: Üretim Süresi Tahmini, Üretim Planlama, Makine Öğrenmesi, Regresyon Yöntemleri

PRODUCT TIME ESTIMATION FOR WARP MACHINE IN WEAVING ENTERPRISES

SUMMARY

Since the processing time has a significant impact on the key performance indicators of the planning process such as total tardiness, total flow time, and completion time, its correct determination has a key role in the effective management of the process. Traditionally, processing time is determined by time study or simple calculations and is assumed to be precisely known in advance of planning and is usually a fixed value.

In some cases, the function used to calculate it is known in advance, although the processing time itself is not known in advance. Such situations often occur in chemical and metallurgical processes. Due to some unexpected situations in the actual production process, the processing times cannot be recorded exactly. It is seen that this situation, which is called uncertainty in processing times, is handled in different ways in the literature. The first of these is the fuzzification of the processing time. A second approach to modeling indefinite processing time is interval numbers. Random or stochastic processing time is another way of expressing uncertainty in processing time. In this approach, the processing time of each job is a random variable with a known probability distribution.

In this study, the uncertainty in the processing time is not caused by probabilistic changing factors such as machine failure, power outage, but by a combination of factors with previously known fixed values. In this study, inspired by the warp preparation operation, which is one of the important components of textile production processes, supervised machine learning approaches are used to estimate the processing time of the warp preparation operation. The processing time involved in the preparation of beams for the warp yarns, which are used in weaving machines, is subject to variations based on multiple parameters. To address this, interviews were conducted with subject matter experts, resulting in the identification of 11 features that could potentially impact the processing time. Subsequently, a total of 12 machine learning algorithms, comprising 6 linear and 6 nonlinear models, were employed to estimate the warp preparation process time. The experimental study utilized real-life data extracted from the enterprise's ERP system, ensuring its practical relevance. 12 different supervised machine learning algorithms were run with both training and test datasets and the results were presented in a comparative way. It has been observed that boosting algorithms are superior in terms of both training/adjustment time and estimation accuracy.

Keywords: Processing Time Estimation, Production Planning, Machine Learning, Regression Methods

1. GİRİŞ

Üretim planlaması, belirli hedeflere göre üretim süreçlerinin doğru ve verimli akışını sağlamak için bir dizi karar sürecidir (Khaled ve diğ., 2022). Üretim planlaması, bütünsel üretim planlaması, parti büyüklüğü belirleme ve çizelgeleme gibi çeşitli üretim aşamalarıyla ilgili birçok karar verme problemini içeren karmaşık bir görevdir (Zarte ve diğ., 2021). Çizelgeleme ise üretim planlamasından daha ayrıntılıdır. Her bir ürünü, yarı ürünü, hammaddeyi üretken kaynaklarla eşleştirir. Daha küçük zaman ve sürekli zaman birimleri kullanır (Thomas ve McClainm, 1993).

İşlem süresi, toplam gecikme, toplam akış süresi, yayılım (*makespan*) gibi çizelgeleme sürecinin temel performans göstergeleri üzerinde önemli bir etkiye sahiptir. Çizelgeleme faaliyetlerinde genel olarak işlem süresinin önceden bilindiği ve çoğu zaman sabit bir değer olduğu varsayılır. Bazı durumlarda işlem süresinin hesaplanması gereken fonksiyon önceden bilinmektedir. Bu tür durumlar çoğu zaman kimyasal ve metalürjik süreçlerde ortaya çıkmaktadır.

Örneğin çelik haddehanelerinde külçeler haddelemeden önce gerekli sıcaklığa ısıtılır. Isıtma (işlem) süresi, külçenin o anki sıcaklığına ve bu da külçenin süreç içerisinde beklediği süreye bağlıdır. Bekleme süresi boyunca külçe soğur ve bu nedenle fırında daha fazla ısıtma süresi gerekir (Gupta ve Gupta, 1988). Belirli bir fonksiyona göre değişen işlem süresine çeşitli ortamlardan örnekler verilebilir: Yangınla mücadelede, yangın söndürme faaliyetinin başlamasındaki gecikme, yangını kontrol etmek için gereken zamanı arttırabilir (Kunnathur ve Gupta, 1990). Karanlıkta veya kötüleşen hava şartlarında arama/kurtarma faaliyetinin işlem süresi uzayabilir (Mosheiov, 1996). Tarımda, mahsul kurudukça hasat için gerekli zaman kısalabilir (Alidaee ve Womer, 1999). Bu durumlarda işlem süresi, zaman geçtikçe artıp azalabilmektedir. Literatürde bu şekilde belirlenen işlem süresine bağımlı işlem süresi denmektedir. Fonksiyonun türüne göre doğrusal, parçalı doğrusal veya doğrusal olmayan (Gupta ve Gupta, 1988) türleri bulunmaktadır.

Gerçek üretim sürecinde, makine arızası, enerji sıkıntısı, acil sipariş ekleme, takım yükleme ve boşaltma gibi bazı beklenmedik durumlar sebebi ile işlem süreleri tam olarak kaydedilememekte veya toplanamamaktadır (Joo ve diğ, 2018). İşlem sürelerindeki belirsizliği olarak isimlendirilen bu durum literatürde farklı şekillerde ifade edildiği gözlemlenmektedir.

Yaygın olarak belirsiz işlem süresi, olasılık teorisine dayanan bulanık sayılar ile temsil edilir (Joo ve diğ, 2018). Bulanık işlem süreli çalışmalara özellikle çizelgeleme alanında birçok örnek bulunabilir (Gao ve diğ, 2016). İşlem süresini bulanıklaştırmanın en belirgin zorluğu üyelik fonksiyonunu ve doğru bir olasılık dağılımını belirlemesidir (Lei, 2011).

Belirsiz işleme süresini modellemek için başvuru alan bir diğer yaklaşım aralık sayılarıdır. Bir parametrenin üst ve alt sınırlarını veren aralık numarası gerçel sayı kavramının bir uzantısı olarak veya gerçel bir doğrunun bir alt kümesi olarak düşünülebilir (Moore, 1979). Bulanık sayının bahsedilen dezavantajlarını ortadan kaldıran bu yaklaşım karar matrisindeki belirsizliği temsil etmenin en basit biçimlerinden biridir ve özniteliklerin değerleri hakkında minimum miktarda bilgi gerektirir (Li ve diğ, 2019).

Rastsal (Nagasawa ve diğ, 2015; Horng ve diğ, 2012) veya olasılıklı (Hee ve diğ, 2021; Alzahrani, 2020) işlem süresi, işlem süresindeki belirsizliği ifade etmenin diğer bir yoludur. Bu yaklaşımda her bir işin işlem süresi, bilinen olasılık dağılımına sahip rastgele bir değişkendir.

Görüldüğü gibi literatürde işlem süresi belirsizliğinin sebebi genel olarak eksik/güvenilmez bilgi veya kaçınılmaz stokastik değişkenlik şeklinde tanımlanmıştır (Jamrus ve diğ, 2017). Bu çalışmada işlem süresindeki belirsizlik, makine arızası, enerji kesintisi gibi olasılıklı bir şekilde değişen etkenlerden değil de önceden bilinen sabit değerlere sahip etkenlerin kombinasyonuna göre değişmektedir. Ele alınan problem dokuma kumaş üretim sürecinin önemli bileşenlerinden olan çözgü hazırlama operasyonundan esinlenilmiştir. İşlem süresi tahmini için çeşitli makine öğrenmesi algoritmaları kullanılmıştır.

2. LİTERATÜR TARAMASI

Faydalı bilgini çıkarımında istatistiksel öğrenme 90'ların başından beri araç olarak kullanılmaktadır (Cheng ve diğ, 2017). İstatistiksel ya da makine öğrenmesi dediğimiz bilim, insanların öğrenme şeklini taklit edebilmek için veri ve algoritmaları kullanır ve doğruluğunu kademeli olarak arttırarak öğrenme sürecini gerçekleştirir (Binaman, A. (2022). İlk zamanlar, tıp, biyo-teknoloji, finans, pazarlama gibi alanlarda yoğun bir şekilde uygulanmıştır (Rainer, 2013). Son yıllarda üretim yönetiminde farklı veri analitiği yöntem ve tekniklerinin kullanımını tartışan makalelerin sayısında önemli bir artış gözlemlenmektedir. Cheng ve diğ. (2017) tarafından yapılan literatür çalışmasında, üretim yönetiminde bu tarz çalışmalar daha çok ileri planlama ve çizelgeleme, kalite iyileştirme (*quality improvement*), arıza teşhisi (*fault diagnosis*) ve arıza analizi (*defect analysis*) olmak üzere dört başlık altında toplandığı tespit edilmiştir. Aynı çalışmada akış süresi/çevrim süresi tahmini şeklinde yeni bir kategori tanımlanmıştır. 2009 yılında Choudhary ve diğ. tarafından yayınlanan çalışmada özellikle üretim planlama alanında istatistiksel öğrenme tekniklerinin uygulanması konusunda büyük bir boşluk olduğu belirlenmiştir. Cheng ve diğ. (2017) tarafından yapılan literatür çalışmasında aradan geçen 9 yıla rağmen ilgili konuda birkaç çalışmanın yapıldığı ortaya çıkmıştır. Bununla birlikte, son birkaç yılda, üretim yönetiminde farklı veri analitiği yöntem ve tekniklerinin kullanımını tartışan makalelerin sayısında önemli bir artış olmuştur. Bu bölümde üretim yönetiminde, zamanla ilgili istatistiksel öğrenme uygulamalarına odaklanan çalışmalar incelenmiştir. Çalışmalar genel olarak iki grup altında toplanmaktadır.

Bunlardan ilki müşterinin siparişinin alınması ile mal veya hizmetin müşteriye teslimi arasında geçen süre olarak tanımlanan akış süresidir (*lead time*) (Gunasekaran ve diğ, 2001). Akış süresi; tasarım, sipariş hazırlama, hammaddeler / bitmiş parçalar için satın alma, kuyruk, kurulum, çalışma, taşıma, inceleme ve teslim süresi dahil olmak üzere bir ürünü üretmek için gereken tüm süreleri içerir (Business Dictionary; Schuh ve Stich, 2012). Literatürde akış süresi ürün tamamlama süresi, yayılım süresi

(*makespan*), veya akış (*flow*) süresi olarak da isimlendirilebilmektedir (Huang ve diğ., 2020). Akış süresi; makine arızaları, eksik malzeme, personel eksikliği veya yetersiz çalışan kalifikasyonu gibi sebeplerden ötürü değişkenlik gösterebilmektedir (Burggräf ve diğ., 2017). Bu değişkenlik müşteri güveninin kaybolması ve geç teslimatlar nedeni ile ortaya çıkan yaptırımlardan kaynaklı maliyetlere sebep olabilir. Buna rağmen teslimat tarihleri erken bir aşamada belirlenir ve programdan sapmalar tanımlanabilir ise ortaya çıkan bu maliyetlerden kaçınılabilir (Raaymakers ve Weijters, 2003; Lödding, 2005). Bu yüzden akış süresinin tahmini özellikle son yıllarda araştırmacıların yoğun ilgisini görmektedir.

Lingitz ve diğ. (2018), yarı iletken endüstrisinde lead time tahmini için 11 farklı makine öğrenmesi algoritması kullanmıştır. Gyulai ve diğ. (2018a), optik endüstrisindeki bir akış atölyesi üretim ortamı analiz etmişlerdir. İlgili çalışmada yazarlar, süreç parametrelerinin çeşitliliği ve özellikle müşteri sipariş akışının devam eden işler üzerindeki yüksek etkisi nedeni ile belirsizlik barındıran lead time tahmin etmişlerdir. Gyulai ve diğ. (2018b) çalışmalarında, kapalı döngü bir üretim kontrolü uygulayarak kararlar almak için üretim sağlama sürelerini (lead) proaktif olarak tahmin etmek için makine öğrenimi tekniklerini uygulamıştır. Huang ve diğ. (2020), çok ürünlü seri üretim hattı için yeni bir matematiksel model ve derin öğrenme tekniğini entegre ederek lead time tahmini için hibrit bir yaklaşım önermiştir. Lead time prediction ile ilgili kapsamlı literatüre Burggräf ve diğ. (2017)'nin çalışmasından ulaşılabilir.

Üretim yönetiminde zamanla ilgili istatistiksel öğrenme uygulamalarının yoğunlaştığı diğer bir başlık çevrim zamanıdır. Araştırmacıların bu konuya odaklanmasının en önemli sebebi, teslim tarihine riayet ile çevrim süresinin doğru belirlenmesi arasındaki kuvvetli bağlantıdır (Wang ve diğ., 2018). Bazı çalışmalarda (Chen, 2006; Chen, 2007) output time olarak da isimlendirilen çevrim süresi basitçe bir işin serbest kalma süresi ile tamamlanma süresi arasındaki fark olarak tanımlanabilir. Makinenin bir çevrimi tamamlaması gereken süre olarak da tanımlanan çevrim süresi; bekleme, kontrol, yeniden işlem ve arıza gibi süreleri kapsamaktadır. Çevrim süresinin tahmini konusunda çalışmalar yarı iletken üretimi alanında yoğunlaşmaktadır (Chen ve diğ., 2001; Chien ve diğ., 2005; Backus ve diğ., 2006; Chen ve diğ., 2009; Tirkel, 2011; Meidan ve diğ., 2011). Bu bölümde nispeten güncel çalışmalara yönelik kısa açıklamalar paylaşılmıştır. Wang ve diğ. (2018b), yarı iletken üretiminin en karmaşık

aşaması olan gofret üretimine odaklanmıştır. İlgili çalışmada önce gofret partilerinin çevrim sürelerinde etkisi olabilecek faktörler seçimi için üç bölümden oluşan regresyona dayalı bir model önerilmiştir. Ardından, gofret partilerinin çevrim sürelerini tahmin etmek için paralel bir hesaplama modeli uygulanmıştır. Wang ve diğ. (2018a) Yarı İletken Gofret Üretim Sistemi (*Semiconductor Wafer Fabrication System*) içinde büyük veri kümelerine sahip gofret partilerinin çevrim süresini hızlı bir şekilde tahmin etmek için paralel hesaplama yeni bir veri yoğun döngü süresi tahmin sistemi önermiştir. Chen ve Wang (2020) tarafından yapılan çalışmada DRAM (Dynamic Random Access Memory - Dinamik Rastgele Erişimli Bellek) üretim fabrikasındaki işlerin çevrim sürelerini doğru bir şekilde tahmin etmek için derin sinir ağları (DNN'ler) uygulanmıştır. Çalışmada açıklanabilir bir derin öğrenme yaklaşımı önerilmiştir. Önerilen metodolojide, iş çevrim süresi tahmini için bir DNN'nin tahmin mekanizmasını açıklamak için bir sınıflandırma ve regresyon ağacı (CART) oluşturulmuştur. Sun ve diğ. (2022) NC makinelerinde işlem gören işlerin, besleme hızı ve çevrim süresini tahmin etmek için veriye dayalı bir yöntem (sinir ağları) önermişlerdir. Yazarlar önerilen yöntemin, diğer analitik yöntemlerle karşılaştırıldığında çevrim süresinin %90'dan fazla doğrulukla tahmin ettiğini göstermiştir.

Araştırmalarımıza göre bu çalışma tekstil endüstrisinde genel olarak süre tahmini konusunda bir ilktir. Çalışmada, dokuma aşamasına gönderilecek çözgü leventinin hazırlanması için gerekli süre tahmin edilmektedir. Üretim sürecine, tahminde kullanılan özniteliklere ilişkin detaylı bilgi takip eden bölümde sunulmuştur.

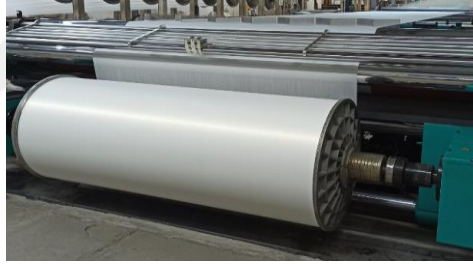
3. MATERYAL VE YÖNTEM

3.1 Çözü Hazırlama Süreci

Bu alt bölümde çözgü hazırlama sürecine ilişkin temel bilgiler paylaşılmaktadır.

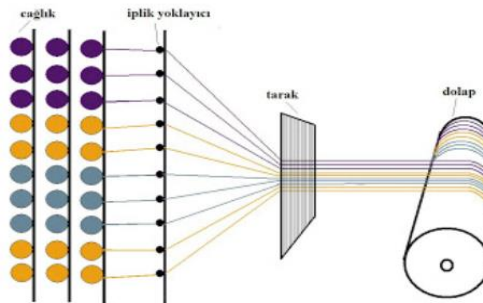
3.1.1 Çözgü

Çözgü örne veya dokuma kumaşlarda, kumaşın kenarına paralel olan veya kumaş metrajını belirleyen boyuna yönündeki ipliklere Şekil 3.1 de görüldüğü gibi çözgü iplikleri ve/veya çözgü (arış) denir. Kumaşın oluşabilmesi için, bu çözgü ipliklerinin içinden tahar planına göre atkı iplikleri geçirilir. Çözgü iplikleri, atkı ipliklerine oranla daha sağlam, kaliteli ipliklerdir.



Şekil 3.1: Çözgü ipliği.

Şekil 3.2 de görüldüğü gibi ipliğin dokumaya hazırlanması için çekilen çözgü işleminin şematik olarak anlatımı aşağıdaki gibidir.



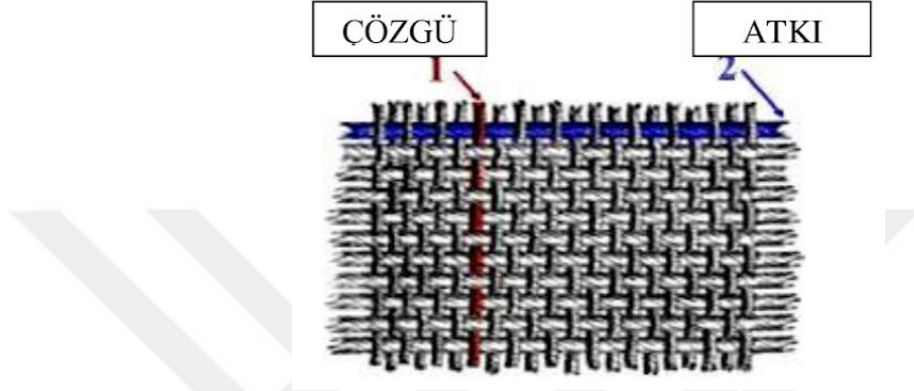
Şekil 3.2: Çözgü çekme işlemi¹.

¹ <http://dokumatasarim.blogspot.com/2008/12/ipliin-dokumaya-hazrlanmas.html>

3.1.2 Atkı

Dokuma kumaşlarda kumaşın üretim yönüne yani çözgü ipliklerine dik doğrultuda, kumaşın eni boyunca uzanan ipliklere atkı iplikleri ve/veya atkı (argaç) denir.

Şekil 3.3'te görüldüğü gibi atkı ve çözgü iplikleri, birbirine dik açıdır ve birleşerek kumaşın dokusunu oluşturur. Dokuma işleminin gerçekleşmesi atkı ipliklerinin atılmasıyla başlar.



Şekil 3.3: Kumaştaki atkı ve çözgü görseli².

3.1.3 Çözgü hazırlama

Bobin halinde bulunan iplikler çözgü hazırlama işleminden sonra dokumada kullanılacak hale gelir (MEGEP 2007). Kumaş dokunurken öncelikle çözgü ipliklerinin hazırlanması gerekir. Dokuma makinesine çözgü için, bobinden direk olarak iplik verilerek kumaş dokunamaz. Bobinlerden ipliklerin, kumaşın dokunacağı makineye uygun levent adı verilen Şekil 3.4'te görüldüğü gibi büyük makaralara aktarılması gerekir. Aktarırken de çözgü tel sayısı, çözgü uzunluğu ya da kumaş boyu, çözgü renk raporu gibi dokunacak kumaşın özelliklerine göre aynı gerginlikte, yan yana ve birbirine paralel olacak şekilde dokuma levendi üzerine sarılarak konumlandırılması işlemine çözgü hazırlama işlemi denir.

² <https://tekstilsayfasi.blogspot.com/2017/12/cozgu-ve-atki-nedir.html>



Şekil 3.4: Çözü leventi.

3.1.4 Çözü hazırlamada dikkat edilecek noktalar

Çözü hazırlama sırasında (İhtiyar, 2019);

- Çözü ipliklerinin çaprazının doğru alınmasına,
- Renk raporunun doğru olmasına,
- İpliklerin birbirine paralel olmasına, üst üste gelmemesine,
- Hazırlanan çözü sayısının tam olmasına,
- İpliklerin çok gevşek ya da çok gergin olmamasına,
- Bütün çözü ipliklerinin aynı gerginlikte olmasına,
- Çözü iplerinin çubuklara çok sıkı bağlanmamasına,
- Çözü ipleri tezgâha bağlanırken çapraz tarafından bağlanmamasına dikkat edilmelidir.

3.1.5 Çözü makineleri

Levent denilen büyük makaralara çözü ipliklerinin istenilen kumaşın özelliklerine göre aktarılırken kullanılan dokuma hazırlık makinelerine çözü makineleri denir.

Konik ve seri çözü makinesi olmak üzere iki tip çözü makinesi vardır. Aralarındaki en belirgin fark ise; seri çözü makinesinde çözü ipliklerinin önce beam denilen büyük makaralara sarılır, daha sonra birleştirme makinesinde birleştirilerek dokuma leventine sarılır. Konik çözü makinesinde ise çözü iplikleri önce kalbalar halinde tambura sarılır, sonra dokuma leventi üzerine aktarılır.

İşletmelerin üretim planlama bölümlerinden gelen iş emrine göre çözü ile ilgili bilgilere göre bobinler ilk olarak çağlığa dizilir. Hazırlanacak olan çözü ipliklerinin hatasız olarak levent üzerine aktarılması, aktarılırken de aynı iplik tansiyonu ile kopuşlara anında müdahâle edilmesi gerekmektedir.

Temel çözgü hazırlık işlemleri 3 şekilde gerçekleştirilmektedir;

- a. Numune çözgü,
- b. Düz/seri çözgü,
- c. Konik çözgü.

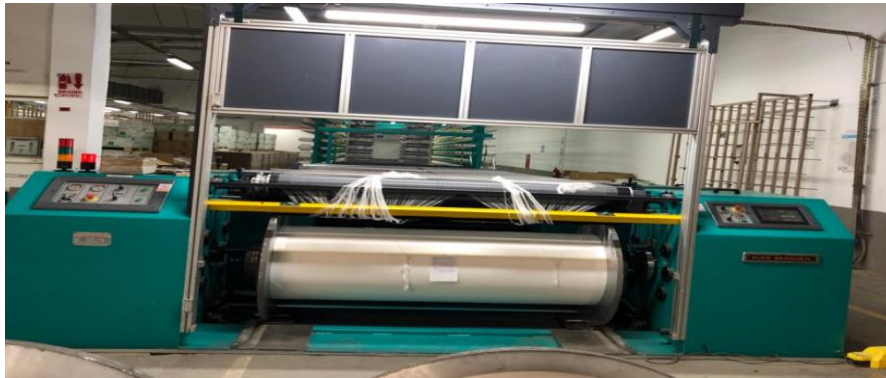
Bu tezde konik çözgü hazırlık işlemlerine değinilecektir. Çünkü çalışma bu çözgü hazırlama yöntemlerinde iplik bobinden doğrudan çekilmektedir (İhtiyar, 2019).

3.1.6 Numune çözgü makinesi

Atkı besleyiciler kullanılarak hazırlanan çözgülerdir.

3.1.7 Düz/Seri çözgü makinesi

Çözgülerin uzun metrajlarda tek çeşit iplikten veya sınırlı renk raporu içeren ipliklerden hazırlanan çözgülerdir. Seri çözgüde, çözgüler önce kısmi leventlere (bim olarak da bilinir) düşük sıklıkta ve uzun metrajlarda Şekil 3.5'te görüldüğü gibi sarılır. Daha sonra istenen/dokunacak kumaşın çözgü sıklığına göre bim'ler çözgü birleştirme makinesinde birleştirilme esnasında dokuma levendine sarım gerçekleştirilir (İhtiyar, 2019).



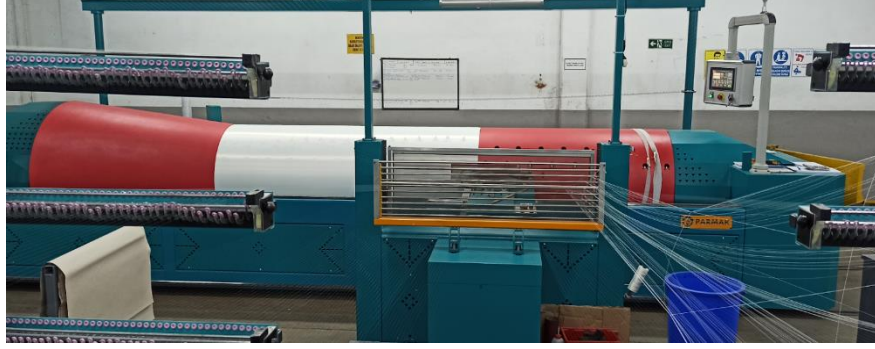
Şekil 3.5: Seri çözgü makinesi.

3.1.8 Konik çözgü makinesi

Konik çözgü makinesine konik denmesinin nedeni, tambur kenarında konik şeklinde (işletme içinde “yastık” olarak bilinen) bir bölümün bulunması ve bu koniklik, çözgünün levente aktarımında çözgü ipliklerinin kaymasını engellemektir.

Şekil 3.6'daki gibi konik çözgü makinesi üzerinde çapraza alma tarağı, toplama tarağı, support, kızak, toplama tarağının konikliğine göre ayarlanabilen aparat gibi aksamlar

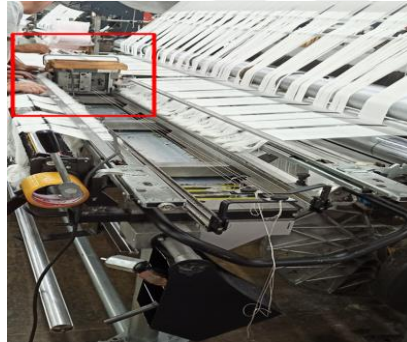
bulunur. Tamburun üzerinde ise kalbaların sarılmasına yardımcı kancalar ve çapraz iplerin bağlanacağı parça vardır³.



Şekil 3.6: Konik çözgü makinesi.

3.1.9 İşbağ işlemi

Çözgü leventlerinin, dokuma makinelerine yerleştirilmesi için Şekil 3.7'deki gibi işbağ yapılır. İşbağ, Dokuma tezgahında eğer aynı çözgüye yani aynı dokumaya aynı kumaşa devam edilecekse makinedeki biten çözgü ile haşıllanmış yeni çözgü uç uca bağlanır.

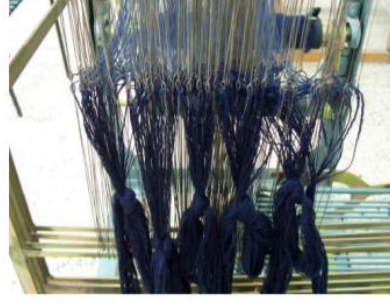


Şekil 3.7: İşbağ işlemi.

3.1.10 Takım tahar

Aynı hareketi yapan çözgü ipliklerinin aynı çerçevelerde toplanmasıdır. Dokumada çözgü ipliklerinin belirli kurallara göre çerçevelerdeki gücü gözlerinden ve tarak dişleri arasından geçirilmesi işlemine Şekil 3.8'deki gibi Takım-Tahar denir.

³ <https://www.derstekstil.name.tr/konik-%C3%A7%C3%B6zg%C3%BC-makinesi.html>



Şekil 3.8: Tahar işlemi.

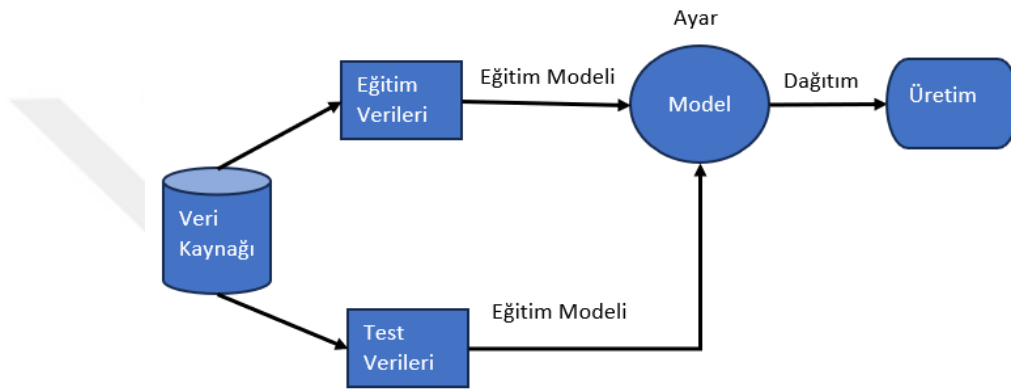
3.1.11 Çözü makinesi iş akışı

Şekil 3.9’da iş akış şeması verilen çözgü makinesinde çözgü hazırlamak için yapılacak işlemleri şu şekilde sıralayabiliriz. Çözgü çekmek için kullanılacak iplik bobinlerinin çağlığa dizilmesinden sonra iplik, çağlık üzerindeki geçiş yollarından / kılavuzlardan geçirilir. Çözgü ipliklerinin çağlık bitimindeki çapraz ve toplama tarağından geçirilmesi, çağlık tansiyon ve direk ayarının yapılması gerekir. Çekilecek çözgünün kaç metre olacağının bilinmesi ve buna göre çözgüsü çekilecek kumaşın tel sayısına, stoktaki ilgili bobinin adedine ve çağlık kapasitesine göre kalba sayısının tespit edilmesi önemlidir. Kalbalar halinde çözgünün tambura alınması, her kalba başında ve sonunda iplerin çapraz alınması işlemi, toplam tel sayısı tamamlanana kadar devam eder. Tamburdaki çözgünün alınacağı, dokuma leventinin makineye takılması, ayarlanması ve kalbalar halinde sarılan çözgü ipliklerinin levente sarılması gerekir.

3.2.1 Makine öğrenmesi teknikleri

3.2.1.1 Denetimli öğrenme

Girdi ve çıktıları olan bir eğitim setinin bir denetmen gözetiminde gözlemlendiği ve girdilerin çıktılar üzerindeki etkileşiminin gözlemlendiği makine öğrenmesi metoduna şekil 3.10’da görüldüğü gibi denetimli öğrenme denir. Gözlemlerden oluşan veri setinin sisteme öğretilip tanıtılmasıyla daha önce hiç bilinmeyen bir örnek için tahminde bulunmamızı sağlar. Girdileri belirlenen çıktılara göre haritalayarak, girdi ile çıktı arasındaki ilişkiler öğrenilmiş olur (Öztemel, 2012).



Şekil 3.10: Denetimli öğrenme (Singh, 2016).

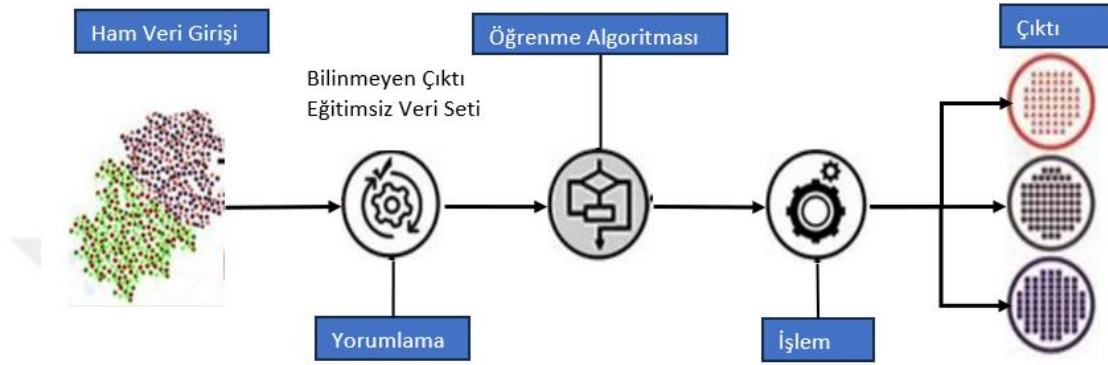
Öğrenme yapılırken bir fonksiyon oluşturulur ve bu fonksiyon ile her bir girdi için elde edilen çıktı ile gerçekleşen çıktı arasındaki fark, en küçüklenmeye çalışılır. Fark, ile etapta belirlenmiş hata aralığına ulaşana kadar sistem eğitime devam eder. Eğitim bittiğinde model istenilen seviyeye ulaşmış olur. Nihai olarak ulaşılan model kullanılarak, eğitim setinde daha önce olmayan veri işlenip, gerçeğe yakın tahmin yapılmaya çalışılır. Regresyona dayalı tahminleme ve sınıflandırma problemlerinde genellikle denetimli öğrenme kullanılmaktadır (Şanlıtürk, 2018). Toplam veri setinin, eğitim veri seti olarak %70’i, değerlendirme veri seti %20’si, test verisi olarak da %10’u kullanılır (Sümer, 2022).

Denetimli öğrenme modeli, kredi derecelendirme modellerinden bilimsel alanlara kadar pek çok alanda önemli başarı elde etmiştir (Melnikov ve diğ., 2023).

3.2.1.2 Denetimsiz öğrenme

Herhangi bir denetiçinin olmadığı, yalnızca girdi değişkenlerine sahip gözlemlerden kendi kendine öğrenerek oluşan, girdi değişkenleri arasındaki ilişkiyi bulmaya çalışan

Şekil 3.11’de de gösterildiği gibi öğrenme yöntemidir. Kümeleme problemleri ile boyut azaltmada kullanılan öğrenme yöntemidir (Şanlıtürk, 2018). Bu öğrenme şeklinde amaç aslında, kümeleme yöntemiyle bilinmeyen nesne kategorilerini bulabilmektir. (Sümer, 2022) Denetimsiz öğrenme, farklı endüstrilerdeki müşteri segmentasyonundan suç faaliyetlerinin tespitine kadar birçok gerçek dünya sorununa uygulanır (Melnikov ve diğ., 2023).



Şekil 3.11: Denetimsiz öğrenme (Singh, 2016).

3.2.1.3 Pekiştirmeli öğrenme

Denetimli öğrenmeden farklı olarak, pekiştirmeli öğrenme yöntemi, deneme yanılma yöntemiyle istenen hedefe ne şekilde ulaşılacağını öğrenen, bulan bir öğrenme türüdür. Sistem her aksiyon/eylem sonrasında bir ceza veya ödül alır ve bu geri bildirimleri düşünerek bir sonraki aksiyon/eylemde ödülü/başarıyı maksimize etmeyi hedefler (Şanlıtürk, 2018). Mevcut durumu verilen belirli eylemlerin sonuçlarını araştırarak ve buna göre model değişkenlerini optimize ederek bu ortamı keşfetmesi sağlanır (Melnikov ve diğ., 2023).

Pekiştirmeli öğrenmede her girdi ve aksiyon için denetçi doğrudan hedeflenen çıktıyı göstermez. Ancak denetçi, geri bildirimler verir. Bu geri bildirimlerde sistemin yarattığı sonucun doğru olup olmadığı hakkında bilgilendirmeler bulunur. Böylece sistem hedefe en yakın sonuçları elde etmeyi öğrenmeye çalışır (Şanlıtürk, 2018).

Pekiştirmeli öğrenme birçok alanda uygulanabilir, örneğin robot navigasyonlarında, beceri edinme süreçlerinde, gerçek zamanlı karar verme problemlerinde ve yapay zeka tabanlı bilgisayar oyunlarında kullanılır. Örneğin, satranç oyununun bir bilgisayar tarafından öğrenilmesi bu öğrenme türüne bir örnek olarak verilebilir. Bilgisayar, beraber oynadığı gerçek kişinin hamlelerinin akabinde rakibini yenmek için en doğru

hamleyi belirlemeye çalışıp ve karşısındaki oyuncunun bir sonraki hamlelerini tahmin ederek üstünlük sağlamayı hedefler (Şanlıtürk, 2018). Genellikle oyun endüstrisinden büyük ilgi görmektedir, aynı zamanda portföy yönetimi gibi gerçek hayat senaryolarına da katkıda bulunmuştur (Melnikov ve diğ., 2023).

3.2.2 Makine öğrenmesi regresyon algoritmaları

3.2.2.1 Doğrusal regresyon algoritmaları

Çoklu doğrusal regresyon (Multiple Linear Regression -MLR): Parametreler arasında kolaylıkla yorumlanabilen, takribi olarak doğrusal bir ilişki olduğunu kabul eden en temel doğrusal regresyon modelleridir (Lingitz ve diğ., 2018).

Temel bileşenler regresyonu (Principal component regression -PCR): Çok sık kullanılan bir boyut azaltma tekniği olan TBR, çoklu bağlantı sorununu çözer ve bağımsız değişkenlerin sayısını azaltmak için bağımsız değişkenlerin doğrusal kombinasyonlarını kullanır. TBR, yanlış bir regresyon tekniğidir ve temel fikri, bağımlı değişkeni doğrudan bağımsız değişkenler üzerine regresyonlamak yerine, bağımsız değişkenlerin temel bileşenlerine indirgeyerek regresyon modeli oluşturmaktır. (Kurnaz, 2020).

Kısmi En Küçük Kareler Regresyonu (Partial least squares regression -PLCR): Wold (1966) tarafından 1960'ların başında geliştirilen Kısmi En Küçük Kareler Regresyonu (KEKR), eksik değerlerle tahmine dayalı etkili modeller oluşturmak için küçük veri kümeleriyle ve çok sayıda (gözlem sayısından bile fazla) yüksek oranda çoklu bağlantıya sahip olduğunda kullanılan bir yöntemdir. (Phatak ve Jong, 1997). Amaç, tahmin için yararlı olan bağımsız değişkenlerdeki bilgilerin çoğunluğunu yakalayan bileşenler yaratarak regresyon probleminin boyutunu azaltmaktır. (Garthwaite 1994).

Ridge regresyonu (RR): Çoklu bağlantı problemlerini ele almak amacıyla kullanılan Ridge regresyon yöntemi, en yaygın yöntemlerden birisi olmakla beraber örneklem boyutu küçük olduğunda diğer yaklaşımlardan özellikle daha iyi performans gösterir (El-Dereny ve Rashwan, 2011; Fitrianto ve Yik, 2014). Önceki iki yöntemde, çoklu bağlantı sorununu çözmek için mevcut bağımsız değişkenlerin kombinasyonlarından oluşan daha az değişken seti ile tahmin modeli oluşturulması esas alınmaktaydı. Ancak, bu ve sonraki başlıklarda açıklanan doğrusal yöntemler, cezalandırma

prensibine dayanmaktadır. Bu modeller aynı zamanda küçültme/daraltma modelleri (Shrinkage models) olarak da adlandırılır. (Fidanoğlu and Akdeniz, 2010).

Lasso regresyonu (LR): Yüksek boyutlu verilerin regresyonu için uygun bir yöntemdir. Ridge regresyonu (RR), nihai modelde tüm bağımsız değişkenleri içerirken, Lasso regresyonu en az önemli özelliklerin ağırlıklarını tamamen ortadan kaldırma eğilimindedir (yani, onları sıfıra ayarlama). Başka bir deyişle, Lasso regresyonu aynı zamanda özellik seçimi yapar. (Aytekin, 2021). Dolayısıyla LR, aşırı öğrenmeyi önleyen yorumlanabilir modeller verir (Tibshirani, 1996).

ElasticNet: Ridge regresyon ve LR arasında bir köprü sağlayan, katsayıların ve her iki algoritmanın düzenlileştirme normu ile eğitilen doğrusal regresyon yaklaşımıdır (Scikit-Learn, 2020; Hans, 2011). Bu ismi almasının nedeni, büyük balıkları kaçırmayan bir ağ gibi davrandığı içindir (Zou and Hastie 2003).

3.2.2.2 Doğrusal olmayan regresyon algoritmaları

K en en yakın komşular (k nearest neighbors -KNN): Eğitim setinden en yakın k örneğini kullanarak sonuçları tahmin eder (Kuhn, 2013). Parametrik olmayan bir yöntem olan, K en yakın komşular algoritması, sınıflandırma ve regresyon içindir. (Altman, 1992). KNN algoritmasının tek hiper parametresi k değeridir. Bu değer çok düşükse aşırı öğrenme (*overfitting*) durumu oluşurken, çok yüksekse eksik uyum durumu (*underfitting*) oluşur. KNN algoritmasının dezavantajı, mesafe hesaplama işleminden ötürü veri miktarı arttıkça işlem yükünü artmasıdır (Seçkin ve diğ, 2019).

Çok katmanlı algılayıcılar (Multilayer perceptron -MLP): ÇKA'ler, evrensel tahminçiler olarak çalışan ve sinir ağı modelleridir, yani herhangi bir sürekli fonksiyona yaklaşabilirler (Hornik, 1991). İleri beslemeli algoritmalar kategorisine giren çok katmanlı algılayıcılar, giriş ve çıkış katmanlarına ve birçok nöronun birlikte istiflendiği bir veya daha fazla gizli katmana sahiptir (Bento, 2021).

Regresyon ağaçları (Regression trees -RT): RA, basit if-then-else karar kurallarıyla çalışan ve denetimli öğrenmede sıklıkla kullanılan bir algoritmadır (Quinlan, 1986). Regresyon ağaçları, adından da anlaşılacağı üzere ağaç şekli biçiminde sınıflandırma yada regresyon modelleri oluşturur. Veri setindeki karmaşık düzensiz yapıları, basit anlaşılır karar yapılarına dönüştürmeyi amaçlar. Bir hedef değişkene göre belirlenen veri setleri, homojen şeklinde alt gruplara ayrılır (Breiman ve diğ, 2017). Algoritmanın

sonucu, karar düğümleri olan bir ağaçtır. RA'ler hem kategorik hem de sayısal verileri işleyebilir (Quinlan, 1987).

Rastsal orman (Random forest -RF): Rastsal orman (RO), tahmin ediciler arasındaki korelasyonu azaltmak için ağaç oluşturma sürecinde sadece rastgele seçilen tahmin edicilerin kullanıldığı torbalı (*bagged*) regresyon ağacının özel bir durumudur (Lingitz ve diğ, 2018). RF, birçok RT'den oluşan bir orman şeklinde tasarlanmıştır. Topluluk yöntemleri grubundan olan RO, (*ensemble methods*) yorumlanması çoğu durumda kolaydır ve değişkenler arasında doğrusal olmayan korelasyonu yakalama yeteneğine sahiptir (James ve diğ, 2013). Ancak aşırı öğrenme eğilimdedir (Bishop, 2006).

LightGBM: Gradyan arttırma algoritması olarak da bilinen LightGBM, Ağaç tabanlı öğrenme algoritmalarını kullanır (LightGBM documantation 2022). Microsoft tarafından geliştirilen bu algoritma bellek tüketimini azaltmak eğitim sürecini hızlandırmak için histogram tabanlı bir yaklaşım kullanır (Machado ve diğ, 2019). Sürekli değere sahip olan değişkenleri kesikli (*discrete bin*) hale getirerek hesaplama maliyetini azaltır (Muratlar, 2020). Ke ve diğ. (2017) tarafından yapılan çalışmada diğer modellere göre 20 kat daha hızlı olduğu sonucuna ulaşılmıştır.

CatBoost: Yandex mühendis ve araştırmacıları tarafından geliştirilen yeni bir gradyan artırma algoritmasıdır. Açık kaynaklı olarak hizmete sunulan bu algoritma, kategorik özellikleri düşük bilgi kaybıyla işleyebilmektedir (Jabeur ve diğ, 2021). CatBoost, eğitim için tüm veri setinin kullanılmasına izin veren daha verimli bir strateji kullanır (Dorogush ve diğ, 2017). Ayrıca CatBoost, geleneksel gradyan artırma algoritmalarının neden olduğu aşırı öğrenmenin üstesinden gelmek için ağaç yapısını seçerken yaprak değerlerini tahmin etmek için rastgele permütasyonlar gerçekleştirir (Dorogush ve diğ, 2018). CatBoost'un bir diğer avantajı, varsayılan parametrelerle iyi sonuçlar sağladığından parametre ayarlama için harcanan süreyi azaltmasıdır (CatBoost documantation).

3.3 Uygulama Veri Seti

Bu çalışma için kullanılan veri seti, işletmenin 2018-2022 yılları arası tutulan üretim kayıtlarından alınmıştır. Toplam tarihi kayıt sayısı, 8000 satırdır. Veriler, işletmenin ERP sisteminden çıkarılmış ve xlsx formatındaki dosyada saklanmıştır. Veri setindeki değişkenlerin açıklamaları Çizelge 3.1'de, iki kategorik değişken bulunmaktadır.

Örnek olarak iplik cinsi (iplik ürün no) kategorik değişkeni frekansı Şekil 3.12’de ve nümerik değişkenlerin tanımlayıcı istatistikleri Şekil 3.13’te verilmiştir.

```

56857    522
10044    456
24801    415
51225    294
10402    270
...
69342     1
27579     1
69649     1
27507     1
68142     1
Name: IplikUrunNo, Length: 281, dtype: int64

56857    522
10044    456
24801    415
51225    294
10402    270
...
52390     34
48981     33
29252     32
28129     31
35751     31
Name: IplikUrunNo, Length: 67, dtype: int64

```

Şekil 3.12: Kategorik değişkene ait frekans bilgisi.

İplik ürün no kategorik değişkenine ait toplam 281 sınıf bulunmaktadır. Veri ön işleme aşamasında 30 in altında kayıt olan sınıflar veri setinden çıkarılmıştır. Düzenleme sonrası kategorik değişkenlere ait sınıf değerleri Şekil 3.12’nin sağ tarafındaki hale gelmiştir.

	count	mean	std	min	25%	50%	75%	max
Çzg_Tel_Sayisi	7558.0	10063.310797	8540.930505	286.00	4380.0000	6800.00	12900.00	44880.00
Iplik_Denye	7558.0	354.592088	550.989547	20.00	75.0000	100.00	400.00	2500.00
Cozgu_Eni	7558.0	277.680868	69.057499	95.00	214.0000	315.00	330.00	360.00
Levente_Alma_Devri	7558.0	131.414792	47.497375	50.00	100.0000	100.00	200.00	250.00
Cozme_Devri	7558.0	368.582562	161.633340	80.00	200.0000	400.00	500.00	700.00
Levent_Metre	7558.0	2558.223736	2143.181288	70.00	1200.0000	2000.00	3200.00	19250.00
Toplam_Kalba_Sys	7558.0	20.224663	14.880346	1.00	9.0000	15.00	28.00	109.00
Caglik_Dizildi_mi	7558.0	0.518391	0.499695	0.00	0.0000	1.00	1.00	1.00
Tarak_Atildi_mi	7558.0	0.619873	0.485450	0.00	0.0000	1.00	1.00	1.00
Islem_Sure	7558.0	4.921917	2.070487	1.16	3.4425	4.49	5.95	21.14

Şekil 3.13: Veri setinin sayısal değişkenlerinin tanımlayıcı istatistikleri.

3.4 Google CoLab Hakkında Genel Bilgiler

Makine öğrenmesi ile ilgili çok çeşitli bilgisayar destekli öğretim/uygulama araçları bulunmaktadır. Son zamanlarda, Google Colaboratory veya kısaca "Colab" gibi yeni araçlar, eğitim ve araştırma faaliyetleri için ücretsiz olarak kullanılabilen çevrimiçi bir Jupyter not defteri hizmeti sunmaktadır. Başka bir deyişle Google Colab, tarayıcı aracılığıyla metin açıklamaları ve Python kodları yazarak makine öğrenimini öğretmek için yaygın olarak kullanılan bir bulut Jupyter not defteridir.

Colab, tarayıcı aracılığıyla python kodu ve makine öğrenmesi araçlarını (Tensorflow, Keras, vb.) yazıp çalıştırmaya ayrıca tekrarlanabilir içeriklerin paylaşımına imkân tanır (Canesche ve diğ, 2021).

3.5 Uygulama Adımları

Bu bölümde yapılan uygulamaya yönelik adımlar alt başlıklar halinde verilmiştir.

3.5.1 Gerekli kütüphanelerin dağıtımı projeye dahil edilmesi

Bu bölümde çalışmada kullanılan kütüphanelerden bahsedilmiştir. İlgili kütüphanelerdeki fonksiyonların ve algoritmaların kullanılabilmesi için yüklenmesi gerekmektedir. Uygulamada kullanılan kütüphanelerin yükleme kodları Şekil 3.14'te verilmiştir.

```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.metrics import mean_squared_error, r2_score
```

Şekil 3.14: Kütüphane yükleme kodları.

3.5.1.1 Numpy

Numerical Python'ın kısaltması "NumPy"dır. NumPy, bilimsel hesaplamalar için temel bir kütüphanedir. Python'daki matrisler ve çok boyutlu diziler üzerinde işlemler yapmak için birçok özellik sunar. Matematiksel işlemleri kolaylaştırır ve hesaplamaların hızlı bir şekilde gerçekleştirilmesine olanak sağlar (Osmanoğlu, 2021). Makine diline yakın bir dili olan Numpy kütüphanesi, C++ dilinde yazıldığı için hızlı çalışmasını sağlamaktadır. Bu yüzden makine öğrenmesi, veri bilimi gibi yapay zekâ uygulamalarında sıklıkla kullanılmaktadır (Bozdemir, 2022).

3.5.1.2 Pandas

Etiketli ve ilişkisel verilerle basit ve sezgiye dayanarak çalışmak üzere dizayn edilmiş bir Python kütüphanesidir. Veri üzerinde ön işleme ve analizler gerçekleştirmek için kullanılan bir pakettir. Hızlı ve kolay bir şekilde veri işleme, görselleştirme ve veriyi yönetme için tasarlanmıştır (Osmanoğlu, 2021).

Python için yüksek performanslı veri yapıları ve veri analiz araçları sağlayan bir kütüphanedir. Pandas, veri manipülasyonu, temizleme, dönüşüm ve analizi için çeşitli fonksiyonlar ve veri yapıları sunar (Karaj, 2023).

3.5.1.3 Scikit-learn (sklearn)

Birçok algoritmayı bize sunan Scikit-learn kütüphanesi, ihtiyaç duyulan sınıflama, kümeleme, tahminleme ve modelleme gibi bütün alanları karşılayan, veri analistleri ve yapay zekâ araştırmacıları tarafından en çok kullanılan açık kaynak kodlu kütüphane olma özelliğiyle öne çıkmaktadır (Osmanoğlu, 2021). Scikit-learn kütüphanesi ile; destek vektör makineleri, k-en yakın komşuluk, karar ağaçları, rasgele orman, yapay sinir ağları, naive bayes, k-ortalamlar algoritmaları kullanılabilir (Torkul ve diğ., 2017).

3.5.1.4 Seaborn

Seaborn, Python'da veri görselleştirmesi için kullanılan popüler bir kütüphanedir. Matplotlib'e dayanan Seaborn, daha çekici ve bilgilendirici görselleştirmeler oluşturmayı kolaylaştıran daha yüksek seviye bir arayüz sunar. Seaborn, istatistiksel veri keşfi ve veri analitiği süreçlerinde kullanılan bir dizi grafik türü sunar. Bu grafik türleri, çizgi grafikleri, saçılım grafikleri, histogramlar, yoğunluk grafikleri, kutu grafikleri, sıcaklık haritaları, çift eksenli grafikler, etkileşimli grafikler ve daha fazlasını içerir. Bu grafik türleri, veri setinizin özelliklerini ve ilişkilerini keşfetmek için kullanılabilir. Seaborn, veri görselleştirmesini basitleştirmek için Matplotlib'in üzerine bir katman ekler. Seaborn'un ana avantajlarından biri, grafiğinize hızla stil ve renk paleti uygulamanızı sağlayan yerleşik renk paletleri ve temaları içermesidir. Böylece, daha profesyonel görünümlü ve tutarlı görseller oluşturabilirsiniz. Ayrıca, Seaborn, veri setindeki değişkenler arasındaki ilişkileri incelemek için çeşitli istatistiksel yöntemleri destekler. Örneğin, regresyon analizi, noktalardaki ilişkileri görselleştirmek için yaygın olarak kullanılan bir yöntemdir. Seaborn, regresyon çizgilerini ve güven aralıklarını göstermek için bu tür analizleri kolaylıkla uygulayabilir.

3.5.1.5 Statsmodels

Statsmodels, Python'da kullanılan bir kütüphanedir. Bu kütüphane, kullanıcıların verileri anlamalarına, istatistiksel modelleri tahmin etmelerine ve istatistiksel testler

gerçekleştirmelerine olanak sağlar. Farklı veri tipleri için çeşitli açıklayıcı istatistikler, istatistiksel testler, çizim işlevleri ve sonuç istatistiklerini içeren kapsamlı bir özellik listesi sunar (Osmanoğlu, 2021).

3.5.1.6 Matplotlib

Matplotlib, python programlama dili ve onun sayısal matematik uzantısı NumPy için bir çizim kitaplığıdır. Histogram, line plot, 3D plot, scatter plot, contour plot, image plot, ploar plot ve bunların farkı versiyonlarını çizebilen bir çizim kütüphanesidir (Osmanoğlu, 2021). Kütüphaneyi kullanabilmek için, Windows/Linux’de açıp, ‘pip install matplotlib’ komutu ile yüklenir. Oluşturulan modelin doğruluk değerlerini grafik şeklinde görmemizi sağlayabilir (Bozdemir, 2022).

3.5.2 Verinin çekilmesi

.csv formatındaki uygulama verileri pandas kütüphanesi yardımı ile okunarak Şekil 3.15’te görüldüğü gibi tanımlanan dataframe’e aktarılmıştır.

```
df=pd.read_csv("ProblemData.csv", usecols=[1,2,3,4])
```

Şekil 3.15: Veri çekmek için kullanılan kod.

Çözgüyü çekerken o işe ait fiili süre, mevcut erp programından alınmıştır. Fakat sürelerin bazılarında aşağıdaki nedenlerden dolayı sapmalar olabilmektedir:

- Elektrik kesintisi olduğunda erp sistemine kayıt düşmüyor. Sonradan operatöre sorularak süreye elle müdahale edilebiliyor.
- Bir çözgü planı içinde genelde sadece bir alt iş emri olur. Fakat bazı durumlarda birden fazla alt iş emirleri olabiliyor. Birden fazla iş emri olduğunda, erp programında plana başla dediğimizde her bir alt iş emri için iş başlamış oluyor. Dolayısıyla işlerin başlangıç zamanları, ikinci ve sonrası alt iş emirleri için gerçek başlangıç zamanları olmuyor, sadece ilk başlanan alt iş emri için gerçek başlangıç zamanı oluyor. Bu gibi durumlarda yine operatör ile görüşerek elle müdahale ediliyor.
- Plan kartı üzerindeki barkodun silik çıkması, barkod okuyucudaki tutukluk vb... nedenlerden dolayı barkodun okutulamaması.

3.5.3 Özellik seçimi

Makine öğrenimi iş akışının ilk adımı olan verilerin toplanması doğru modeller oluşturmak için kritik öneme sahiptir. Makine öğrenimi yöntemleri için girdi değişkenlerini seçerken, eksiksizlik-veri boyutu ve gürültüden kaçınma arasında bir denge yapılmalıdır. Eksik veri, bağımlı değişkenin bağımsız değişkenlerce açıklama oranının düşük olmasına ve dolayısıyla yüksek tahmin hatasına sebep olacaktır. Aksi durumda yani fazla verinin kullanılması fazla bilgiyi sağlayacak, ancak tahmin için gerekli olmayan birçok bilgiyi ve gürültüyü de içerecektir. Bu, daha büyük miktarda veri nedeniyle makine öğrenimi algoritmasının eğitim sürecini daha zor hale getirecektir (Scher ve Messori, 2018).

Bu bağlamda konu ile ilgili tüm uzmanların katıldığı bir ekiple değerlendirme toplantısı yapılmış ve bir levendin konik çözgü hazırlama makinesinde sarılabilmesi için gerekli işlem süresi, Çizelge 3.1’de verilen parametrelere bağlı olarak değişkenlik gösterebileceği belirlenmiştir.

Çizelge 3.1 : Tahmin parametreleri

Parametre	Açıklama	Tür	Aralık
Çözgü tel sayısı	Kumaşın enine göre cm’deki toplam tel (iplik) sayısıdır.	Int	(286, 44880)
İplik Cinsi	Çözgü ipliği olarak kullanılan ipliğin türüdür. (20 tür)	Kategorik	
İplik Denye	9000 metre uzunluğundaki ipliğin ağırlığı (gr)	Int	(20, 2500)
Çözgü (tarak) eni	Kumaş eni (cm)	Int	(95, 360)
Levente alma devri	Tambura sarılan çözgü ipliklerinin, dokuma leventine aktarma hızıdır. (m/dk)	Int	(50, 250)
Çözme Devri	Cağlıktaki ipliklerin tambura sarılma hızıdır. (m/dk)	Int	(80, 700)
Levent metre	Dokuma leventine sarılan ipliklerin sipariş metrajıdır. (m)	Int	(70, 19250)
Toplam kalba sayısı	İstenilen kumaş enini oluşturmak için yan yana sarılan bantların sayısıdır.	Int	(1, 40)
Cağlık dizildi mi?	Bobinlerin cağlığa takılma işleminin yapıldığı (1) ya da yapılmadığı (0) durumu ifade eder.	Int	(0, 1)
Tarak atıldı mı?	Öncül işin çözgü eni (kalba genişliği), ardıl iş ile uyumsuz ise ardıl işe ait tarak değiştirilme işleminin yapılıp (1) yapılmadığı (0) durumu ifade eder.	Int	(0, 1)
MakinaID	Çözgü hazırlama işleminin yapıldığı makine numarasıdır. (5 makine)	Kategorik	
İşlem süresi	Bağımlı (çıktı) değişken (saat)		

Konik çözgü makinesinde iş emrinde belirtilen çözgü ipliğinin bir levente sarılma süresi; iş emrindeki barkod numarasının okutulması ile başlayıp, sarım sonrası ilgili levente ait üretim etiketinin ERP programından çıkartılması ile sonlanmaktadır. Bu sürenin içinde her iş emri için bir defa yapılan çağlık dizilimi ve tarak atma (hazırlık işlemi) dâhildir. Yani bir iş emrinde beş levent sarımı planlanmış ise ilk leventin işlem süresi belirtilen hazırlık işlemlerinden dolayı daha uzundur. Ayrıca bu süreye sarım devam ederken karşılaşılan ipin kopması nedeni ile yaşanan duruşlar, biten bobin yerine yeni bobinin takılması (bobin değişimi) ve kısa süreli makine arızaları gibi süreler de bir leventin sarımı için kaydedilen çevrim süresine dahildir.

3.5.4 Veri önileme

Makine öğrenimi yoluyla daha doğru tahminler oluşturmak için önemli bir adım olan veri ön işlemede, olası hatalı, eksik veya tutarsız veriler (aykırı değerlerin) tespit edilir. Bu aşamada yapılacak işlemler tercih edilen makine öğrenmesi algoritmasına bağlı olarak da değişkenlik gösterir. Örneğin, bazı algoritmalar (Rastgele Orman ailesi gibi) girdilerinde boş değerleri desteklemezken, diğerleri bunları işleyebilir. Veya bazı makine öğrenmesi algoritmaları aykırı değerlerin varlığında diğerlerinden daha sağlamdır (Chibani and Coudert, 2020). İşlem süreleri ve ilişkili parametreler ERP sisteminden toplanmaktadır. 2018-2022 yıllarına ait ham veri setlerinde, hatalı/eksik veri girişi veya birim dönüştürme gerekliliğinden ötürü düzenleme gerekmektedir. Örneğin iplik numaralandırma sistemi olarak denye seçilmiştir. Ancak bu alanda İngiliz pamuk ipliği sistemi (Ne) cinsinden veriler tespit edilmiştir. Farklı birimdeki bu veriler denyeye dönüştürülmüştür. Operatörlerin sayısal bir alana metinsel veriler girmesi sebebi ile oluşan tutarsızlıklar giderilmiştir. Son olarak her iplik cinsi için 30 kayıtlı altındaki kayıtlar tamamen silinmiştir. ERP sistemine geçişin yaşandığı 2018 yılına ait veriler birçok parametre için eksik olduğundan silinmiştir. Veri önilemeye dair yapılan işlemler aşağıda maddeler halinde sunulmuştur:

- Veri Temizleme (Gürültülü veri (*Noisy data*) ve eksik veri analizi (*Missing data analysis*))

“Çözgü plan numarası” ilgili satıra ait bir ID rolü üstlendiğinden veri setinden çıkarılmıştır. Veriler ERP sisteminden alındığı için sayısal alanlara metinsel ifadeler girilmiş veya boş bırakılmıştır. Metinsel ifadeler ve boşluklar temizlenmiştir. Sistemden 18375 satır veri alınmış, 7232 satırında işlem süre bilgisi olmadığından

silinmiştir. Temizleme sonrası 7558 satır veri kalmıştır. İşlem sonrası değişken türlerine ilişkin bilgi Şekil 3.16’da sunulmuştur.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7558 entries, 0 to 7557
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Çzg_Tel_Sayisi        7558 non-null   int64
1   İplik_Cinsi           7558 non-null   object
2   İplik_Denye           7558 non-null   int64
3   Cozgu_Eni             7558 non-null   int64
4   Levente_Alma_Devri    7558 non-null   int64
5   Cozme_Devri           7558 non-null   int64
6   Levent_Metre          7558 non-null   int64
7   Toplam_Kalba_Sys      7558 non-null   int64
8   Caglik_Dizildi_mi     7558 non-null   int64
9   Tarak_Atildi_mi       7558 non-null   int64
10  Makina                 7558 non-null   object
11  Islem_Sure             7558 non-null   float64
dtypes: float64(1), int64(9), object(2)
memory usage: 708.7+ KB
```

Şekil 3.16: Değişken türleri.

- Değişken dönüşümleri

Diğer bir ön işleme adımı kategorik değişkenlerin numerik hale getirilmesidir. Bunun için “sklearn.preprocessing” kütüphanesinden faydalanılmıştır. İki tip dönüşüm yaklaşımı bulunmaktadır.

One hot Encoding: Kategorik değişkenin olduğu sütunu birden çok sütuna bölerek kategorik verileri sayısal verilere dönüştürür. Sayılar, ilgili sütundaki değere bağlı olarak 1’ler ve 0’larla değiştirilir. Örneğin şehir kategorik değişkenine ait değerleri içeren tek kolon aşağıdaki şekilde dönüştürülmektedir. Kategorik değerler çok sayıda olduğunda yüksek bellek tüketimine sebep olacağından bu yöntem tavsiye edilmez. Genel işleyişi Şekil 3.17 de resmedilmiştir.

Şehir	Şehir	Malatya	Eskişehir	Elazığ	Bursa
Malatya	Malatya	1	0	0	0
Eskişehir	Eskişehir	0	1	0	0
Elazığ	Elazığ	0	0	1	0
Bursa	Bursa	0	0	0	1

Şekil

3.17: One hot encoder'ın genel işleyişi.

Label encoding: Kategorik değişkeni ifade eden bir sütundaki her değeri bir sayıya dönüştürür. Kategorik değerlere ait sıralamanın önemli olmadığı ve kategorik değer çeşitliliği fazla olduğu durumlarda kullanımı avantajlıdır. Genel işleyişi Şekil 3.18'de sunulmuştur.

Şehir	Şehir
Malatya	0
Eskişehir	1
Elazığ	2
Bursa	3

Şekil 3.18: Label encoder genel işleyişi.

3.5.5 Model seçimi ve değerlendirme

Makine öğrenimi prosedürleri, verilerden öğrenerek modeller uydurur (*fit*) ve oluşturulan modeller ile tahminde bulunur. Ancak her veri için uygun evrensel bir model yoktur ve veriye göre farklı makine öğrenmesi teknikleri birbirlerine üstünlük sağlayabilmektedir. Bu nedenle, tipik bir veri analizinde çok önemli bir adım, bir dizi aday modeli değerlendirmek ve ardından en uygun olanı seçmektir (Ding ve diğ., 2018). Özetle, bir eğitim veri kümesi için bir dizi aday makine öğrenimi modeli arasından son bir makine öğrenimi modelinin seçildiği bu süreçte model seçimi denmektedir (Brownlee, 2019).

Son birkaç on yılda, çok çeşitli model seçim yöntemleri önerilmiştir. Yüzde bölme (*percentage split*) ve çapraz doğrulama (*cross-validation*) yöntemleri en yaygın seçim yöntemlerindendir. Yüzde bölme yönteminde tüm veriler test ve eğitim olmak üzere manuel bir şekilde iki sete ayrılır. Öğrenme süreci için eğitim veri seti, performans değerlendirme için test veri seti kullanılmaktadır. Ancak eğitim ve test veri setlerinin

dağılımlarındaki farklılık, aykırı değerlerin eşit dağılmaması vb. nedenlerle değerlendirme sonuçları güvenilir olmayabilir. Bu nedenle çapraz doğrulama yöntemi geliştirilmiştir. Yinelemeli bir süreç olan çapraz doğrulama metodunda veriler yaklaşık olarak eşit büyüklükte k ayırık kümeye bölünür. Her yinelemede 9 ayırık küme eğitim verisi ve bir küme test verisi olarak kullanılır. Çapraz doğrulama yöntemi yüzde bölme yöntemine göre daha güvenilir sonuçlar verir. Ancak her yinelemede öğrenme ve test işlemi yapıldığından sonuçlanması uzun zaman alabilmektedir (Seçkin ve diğ, 2019).

Bu çalışmada modellerin tahmin doğruluğunu belirlemek için 10 kat çapraz doğrulama kullanılmıştır. Çalışmada kullanılan regresyon algoritmaları, yaygın olarak kullanılan beş performans metriği açısından çizelge 3.2'deki gibi karşılaştırılır.

Çizelge 3.2 : Performans metrikleri

Metric	Equation
Root dean squared error (RMSE)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Normalized root mean squared error (NRMSE)	$\frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{y_{max} - y_{min}} * 100$
Mean absolute error (MAE)	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Mean absolute percantage error (MAPE)	$\frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right * 100$
R^2	$1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}$

Burada y_i gerçekleşen işlem süresi, $\hat{y}_i$ ve $\bar{y}_i$ sırasıyla tahmin edilen ve ortalama işlem süresidir. n, test veri setinde dikkate alınan gözlem sayısıdır y_{max} ve y_{min} sırasıyla gerçekleşen değerler arasından en büyük ve en küçük değerlerdir.

3.5.6 Hiperparametre ayarı

Hiper-parametre, öğrenme süreci başlamadan önce ayarlanan başka bir deyişle modelin dışında olan, değeri verilerden tahmin edilemeyen ve model mimarisini tanımlayan parametrelerdir. Hiper-parametre ayarlama/optimizasyonu, bir makine

öğrenimi modelinin davranışını kontrol etmenin önemli bir parçasıdır (Navas, 2022). Makine öğrenimi algoritmalarının performansı büyük ölçüde Hiper-parametrelerine bağlıdır. Bu nedenle, daha büyük bir doğrulukla öngörülebilir bir model tasarlanabilmesi açısından Hiper-parametrelerin ayarlanması kritik bir adımdır. Ayrıca, Hiper-parametrelerin ayarlanması, modelin makine öğreniminde ciddi bir sorun olan aşırı öğrenmeden muzdarip olmasını önler (Dash ve diğ., 2021).

Bu çalışmada kullanılan 12 makine öğrenmesi algoritmasına ait hiper-parametrelerin ayarlanabilmesi için ızgara arama ve rastgele arama yöntemleri seçilmiştir. Izgara arama, arama uzayındaki her olası hiper-parametre kombinasyonunu değerlendirir (James ve diğ., 2011). Her bir kombinasyon için, bu Hiper-parametre değerleri kullanılarak bir model oluşturulur ve değerlendirilir. En yüksek doğruluğa sahip hiper-parametre kombinasyonu rapor edilir (Villalobos-Arias ve diğ., 2020). Özellikle çok fazla hiper-parametre yine çok sayıda seviye için optimize edildiği durumlarda ızgara aramasından hesaplama açısından pahalı olabilir. Bu durumda diğer popüler bir hiper-parametre ayarlama tekniği olan rastgele arama tercih edilir. Rastgele arama, arama uzayının bir alt kümesini örnekleyen bir yaklaşım takip eder (James ve diğ., 2012). Bu çalışmada kullanılan her bir teknik için araştırılan hiper-parametreler ile ilgili bilgi Çizelge 3.3'te sunulmuştur:

Çizelge 3.3 : Seçilen algoritmalara ait hiper-parametre değerleri

Approach	Hyperparameter	Possible Values	Optimal Values
PCR	num_components	(1 ,2 ,..., 39)	35
PLS	num_components	(1 ,2 ,..., 39)	36
RR	alpha	10^x , x=(5, 4.9, 4.8, ..., -0.8, -0.9, -1)	0.005
Lasso	alpha	10^x , x=(5, 4.9, 4.8, ..., -0.8, -0.9, -1)	0.00004
ElasticNet			
KNN	k	1 ,2 ,..., 30)	5
RT	max_depth	(8, 10, 20, 30, 40, None)	20
	max_features	auto, sqrt	auto
	max_leaf_nodes	(120, 140, 160, 190, 210)	210
	min_samples_leaf	(4, 6, 8, 10, 15)	4
	min_samples_split	(2, 5, 8, 10, 20)	10
MLP	learning_rate_init	(0.00001, 0.01, 0.1, 0.5)	0.001
	activation	relu, logistic, identity, tanh	relu
	hidden_layer_sizes	[(3), (6), (4, 4), (20,20), (40, 40), (50, 50), (60, 60)]	(50, 50)
	alpha	(0.1, 0.01, 0.001, 0.0001)	0.0001
Random Forest	max_depth	(2, 4, 6, 8, 10)	10
	max_features	(5, 10, 15, 20, 25, 30, 35)	35
	n_estimators	(200, 500, 1000, 2000)	2000
LightGBM	colsample_bytree	(0.2, 0.4, 0.6, 0.8, 1)	0.8
	learning_rate	(0.07, 0.09, 1.10, 1.30, 1.50)	0.09
	max_depth	(1, 3, 5, 7 ,9)	3
	n_estimators	(1000, 1200, 1600, 1800, 2000)	1800
CatBoost	iterations	(200, 500, 1000, 2000)	2000
	learning_rate	(0.01, 0.03, 0.05, 0.1)	0.05
	depth	(3, 4, 5, 6, 7, 8)	5

4. BULGULAR VE TARTIŞMA

Bu bölümde çözgü hazırlama işlem süresinin tahmin edilmesi problemi için 12 farklı makine öğrenmesi algoritması test edilmiştir. Beş farklı metriğe göre karşılaştırmalı sonuçlar Çizelge 4.1’de sunulmuştur. Tahmin için scikit-learn kütüphanesi ve Google Colaboratory (Colab) notebook kullanılmıştır.

Çizelge 4.1: Farklı makine öğrenimi modelleri arasında performans karşılaştırma

Doğrusal Makine Öğrenmesi Metotları												
	MLR		PCR		PLC		Ridge		Lasso		ElasticNet	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	0,826	0,930	0,826	0,930	0,815	0,910	0,826	0,929	0,826	0,929	0,899	0,988
NRMS	5,47	4,55	5,47	7,16	5,47	4,55	5,54	4,65	5,54	4,65	6,03	4,95
E	%	%	%	%	%	%	%	%	%	%	%	%
MAE	0,591	0,607	0,591	607	0,586	0,595	0,591	0,608	0,591	0,607	0,661	0,669
MAPE	12,70	12,67	12,69	12,67	12,60	12,70	12,69	12,69	12,69	12,67	14,54	14,52
	%	%	%	%	%	%	%	%	%	%	%	%
R ²	0,838	0,838	0,838	0,838	0,842	0,821	0,838	0,837	0,838	0,838	0,808	0,808
Doğrusal Olmayan Makine Öğrenmesi Metotları												
	KNN		RT		MLP		Random Forest		LightGBM		CatBoost	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	0,999	1,235	0,702	1,099	0,491	0,945	0,536	0,836	0,427	0,732	0,416	0,730
NRMS	6,70	6,18	3,70	4,18	3,29	4,73	3,59	4,18	2,86	3,66	2,79	3,65
E	%	%	%	%	%	%	%	%	%	%	%	%
MAE	0,663	0,802	0,510	0,680	0,366	0,529	0,376	0,504	0,238	0,345	0,310	0,395
MAPE	14,30	17,25	11,20	14,22	8,18	10,75	8,31	10,77	5,54	7,58	7,07	8,53
	%	%	%	%	%	%	%	%	%	%	%	%
R ²	0,762	0,762	0,882	0,882	0,969	0,987	0,932	0,932	0,957	0,957	0,959	0,959

Çalışmada kullanılan tüm algoritmalar hem eğitim seti hem de test seti üzerinden değerlendirilmiş ve Çizelge 4.1’de karşılaştırmalı bir şekilde sunulmuştur. Bu şekilde makine öğrenmesinin en temel sorunlardan biri olan aşırı öğrenme durumunun oluşup oluşmadığı gözlemlenmeye çalışılmıştır. Bir algoritmanın eğitim setinde yüksek, ancak test setinde düşük performans göstermesi aşırı öğrenme göstergesidir. Bu durum oluşturulan modelin genelleme yeteneğinin düşük olduğu ve yeni veriler için başarılı sonuçlar vermeyeceği anlamına gelmektedir. Çizelge 4.1’de sunulan sonuçlar KNN

algoritması ile elde edilen sonuçlardaki küçük farklılık dışında tüm algoritmaların eğitim ve test setleri için yaklaşık performans gösterdiği gözlemlenmiştir.

MAPE değerinin %20'nin altında olması, modelin uydurma ve tahminde iyi bir doğruluk oranına sahip olduğunu gösterir. Bu bağlamda analiz edilen tüm yöntemler doğru sonuçlar verme yeteneğine sahip olduğu ve her birinin gerçek hayat şartlarında uygulanabilir olduğu gözükmemektedir. Tüm performans ölçütleri açısından diğerlerinden sürekli olarak daha iyi olan tek bir model yoktur. Ancak genel olarak nonlineer makine öğrenmesi algoritmalarının tüm metriklerde daha iyi sonuçlar verdiği gözlemlenmiştir. Özellikle boosting machine algoritmaları kaliteli sonuçlar vermiştir. CatBoost yöntemi ile LightGBM yöntemleri RMSE ve NRMSE metrikleri bağlamında neredeyse aynı performansı göstermektedir. Ancak MAPE ve MAE bağlamında LightGBM üstünlük sağlamıştır. KNN algoritması NRMSE hariç tüm metriklerde en kötü sonuç vermiştir. En yüksek NRMSE değeri PCR algoritması ile elde edilmiştir.

5. SONUÇ VE ÖNERİLER

Bu çalışmada dokuma kumaş üretim aşamalarının önemli bileşenlerinden olan çözgü hazırlama sürecine odaklanılmıştır. Dokuma makinelerinde kullanılmak üzere çözgü ipliklerinin leventlere sarıldığı ilgili süreçte bir leventin hazırlanması için gereken işlem süresi birçok parametreye bağlı olarak değişmektedir. Öncelikle konunun uzmanları ile görüşmeler yapılmış ve neticesinde işlem süresini etkileyebilecek 11 öznitelik belirlenmiştir. Daha sonra çözgü hazırlama işlem süresinin tahmin edilmesi için 6 doğrusal ve 6 doğrusal olmayan toplam 12 farklı makine öğrenmesi algoritması test edilmiştir. Deneysel çalışma işletmeye ait 2018-2022 yılları arası ERP sisteminden elde edilen veriler ile yapılmıştır. Veri ön işleme neticesinde elenen veriler sonrası yaklaşık 8000 satırlık veri kullanılmıştır. Deneysel çalışması sonucu elde edilen sonuçlar Boosting machine algoritmalarının belirgin bir şekilde diğer algoritmalara üstünlük sağladığı görülmüştür. Geliştirilen tahmin modelinin firma ERP sistemine entegrasyonu planlanmaktadır. İşlem süresinin daha doğru tahmini ile dokuma tezgâhlarındaki - dokuma için gerekli çözgü leventlerinin beklenmesinden kaynaklı - duruşların azalması beklenmektedir. Ayrıca, daha doğru planlama ile çözgü makinelerinin kullanım oranının artması ve neticesinde fasona imalat maliyetlerinin azalması beklenmektedir.

KAYNAKLAR

- Alidaee, B., & Womer, N. K.** (1999). Scheduling with time dependent processing times: review and extensions. *Journal of the Operational Research Society*, 50, 711-720.
- Altman, N. S.** (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185.
- Alzahrani, J.** (2020). Job-shop scheduling optimization with stochastic processing times. *International Journal of Engineering Technologies and Management Research*, 6(1), 73-83.
- Backus, P., Janakiram, M., Mowzoon, S., Runger, C., & Bhargava, A.** (2006). Factory cycle-time prediction with a data-mining approach. *IEEE Transactions on Semiconductor Manufacturing*, 19(2), 252-258..
- Bento, C.** (2021). Multilayer perceptron explained with a real-life example and python code: Sentiment analysis. Towards Data Science.
- Binaman, A.** (2022). İstatistiksel makine öğrenme yöntemleriyle bankaların derecelendirilmesi: Türkiye bankacılık sistemi uygulaması.
- Bishop, C. M.** (2007). Pattern recognition and machine learning (information science and statistics).
- Bozdemir, M.** (2022). Makine çevirisi ile Türkçe Sözel İfadelerin Python Sözdiziminin Oluşturulması (Doctoral dissertation, Bursa Uludag University (Turkey)).
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J.** (2017). Classification And Regression Trees. 678 Routledge. New York.
- Brownlee, J.** (2019). A gentle introduction to model selection for machine learning. ML Mastery.
- Burggräf, P., Wagner, J., Lück, K., & Adlon, T.** (2017). Cost-benefit analysis for disruption prevention in low-volume assembly. *Production engineering*, 11, 331-342.
- Business Dictionary.** Accessed: August 15, 2022. [Online]. Available: <http://www.businessdictionary.com/>
- Bocewicz, G., Nielsen, I. E., & Banaszak, Z. A.** (2016). Production flows scheduling subject to fuzzy processing time constraints. *International Journal of Computer Integrated Manufacturing*, 29(10), 1105-1127.
- Cai, J., & Lei, D.** (2021). A cooperated shuffled frog-leaping algorithm for distributed energy-efficient hybrid flow shop scheduling with fuzzy processing time. *Complex & Intelligent Systems*, 7(5), 2235-2253.

- Canesche, M., Bragança, L., Neto, O. P. V., Nacif, J. A., & Ferreira, R.** (2021, May). Google Colab CAD4U: Hands-on cloud laboratories for digital design. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE.
- CatBoost documentantation**, (2022) Catboost. <https://catboost.ai/>
- Chen, C. Y., George, E. I., & Tardif, V.** (2001). A Bayesian model of cycle time prediction. *IIE Transactions*, 33(10), 921-930.
- Chen, T.** (2006). A hybrid SOM-BPN approach to lot output time prediction in a wafer fab. *Neural Processing Letters*, 24, 271-288.
- Chen, T.** (2007). An intelligent hybrid system for wafer lot output time prediction. *Advanced Engineering Informatics*, 21(1), 55-65.
- Chen, T., Wang, Y. C., & Tsai, H. R.** (2009). Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a SOM-FBPN-ensemble approach with multiple buckets and partial normalization. *The International Journal of Advanced Manufacturing Technology*, 42, 1206-1216.
- Chen, T. C. T., & Wang, Y. C.** An Explainable Deep-Learning Approach for Job Cycle Time Prediction. Available at SSRN 4136028.
- Cheng, Y., Chen, K., Sun, H., Zhang, Y., & Tao, F.** (2018). Data and knowledge mining with big data towards smart production. *Journal of Industrial Information Integration*, 9, 1-13.
- Chibani, S., & Coudert, F. X.** (2020). Machine learning approaches for the prediction of materials properties. *Apl Materials*, 8(8).
- Chien, C. F., Hsiao, C. W., Meng, C., Hong, K. T., & Wang, S. T.** (2005, September). Cycle time prediction and control based on production line status and manufacturing data mining. In *ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005.* (pp. 327-330). IEEE.
- Choudhary, A. K., Harding, J. A., & Tiwari, M. K.** (2009). Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20, 501-521.
- Daniels, R. L., & Kouvelis, P.** (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management science*, 41(2), 363-376.
- Dash, R. K., Nguyen, T. N., Cengiz, K., & Sharma, A.** (2021). Fine-tuned support vector regression model for stock predictions. *Neural Computing and Applications*, 1-15.
- Ding, J., Tarokh, V., & Yang, Y.** (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6), 16-34.
- Dorogush, A.V., A. Gulin, G. Gusev, L. Ostroumova Prokhorenkova, and A. Vorobev.** Catboost: unbiased boosting with categorical features. arXiv preprint arXiv:1706.09516, 2017.
- Dorogush, A.V., Ershov, V., Gulin, A.,** (2018). CatBoost: gradient boosting with categorical features. 1–7

- Ebiç, O.** Tek bobinden çözgü hazırlama makinesinin tasarımı ve değerlendirilmesi (Master's thesis, Eğitim Bilimleri Enstitüsü).
- El-Dereny, M., & Rashwan, N. I.** (2011). Solving multicollinearity problem using ridge regression models. *International Journal of Contemporary Mathematical Sciences*, 6(12), 585-600.
- Fidanoğlu, I., & Akdeniz, F.** (2010) İstatistiksel Daraltıcı (Shrinkage) Model Ve Uygulamaları.
- Fitrianto, A., & Yik, L. C.** (2014). Performance Of Ridge Regression Estimator Methods On small sample size by varying correlation coefficients: a simulation study. *Journal of mathematics and Statistics*, 10(1), 25.
- Gao, K. Z., Suganthan, P. N., Pan, Q. K., Chua, T. J., Chong, C. S., & Cai, T. X.** (2016). An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. *Expert Systems with Applications*, 65, 52-67.
- Garthwaite, P. H.** (1994). An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425), 122-127.
- Ghezavati, V., & Saidi-Mehrabad, M.** (2010). Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time. *The International Journal of Advanced Manufacturing Technology*, 48, 701-717.
- Gujjar, J. P., Kumar, H. P., & Chiplunkar, N. N.** (2021). Image classification and prediction using transfer learning in colab notebook. *Global Transitions Proceedings*, 2(2), 382-385.
- Gunasekaran, A., Patel, C., & Tirtiroglu, E.** (2001). Performance measures and metrics in a supply chain environment. *International journal of operations & production Management*, 21(1/2), 71-87.
- Gupta, J. N., & Gupta, S. K.** (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14(4), 387-393.
- Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W., & Monostori, L.** (2018). Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *IFAC-PapersOnLine*, 51(11), 1029-1034.
- Gyulai, D., Pfeiffer, A., Bergmann, J., & Gallina, V.** (2018). Online lead time prediction supporting situation-aware production control. *Procedia CIRP*, 78, 190-195.
- Han, Y., Gong, D., Jin, Y., & Pan, Q. K.** (2016). Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time. *Applied Soft Computing*, 42, 229-245.
- Hans, C.** (2011). Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 106(496), 1383-1393.
- He, J., Chu, F., Zheng, F., & Liu, M.** (2021). A green-oriented bi-objective disassembly line balancing problem with stochastic task processing times. *Annals of Operations Research*, 296(1-2), 71-93.

- Horng, S. C., Lin, S. S., & Yang, F. Y.** (2012). Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Systems with Applications*, 39(3), 3603-3610.
- Hornik, K.** (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2), 251-257.
- Huang, J., Chang, Q., & Arinez, J.** (2020). Product completion time prediction using a hybrid approach combining deep learning and system model. *Journal of Manufacturing Systems*, 57, 311-322.
- İhtiyar, M.** (2019). Çözgü Makineleri İçin Gerginlik Kontrol Sistemi Tasarımı (Doctoral dissertation, Bursa Uludag University (Turkey)).
- Jabeur, S. B., Gharib, C., Mefteh-Wali, S., & Arfi, W. B.** (2021). CatBoost model and artificial intelligence techniques for corporate failure prediction. *Technological Forecasting and Social Change*, 166, 120658.
- Jamrus, T., Chien, C. F., Gen, M., & Sethanan, K.** (2017). Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 31(1), 32-41.
- James, G., Witten, D., Hastie, T., & Tibshirani, R.** (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Bergstra, J., & Bengio, Y.** (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B.** (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- Jia, Z., Yan, J., Leung, J. Y., Li, K., & Chen, H.** (2019). Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities. *Applied Soft Computing*, 75, 548-561.
- Joo, B. J., Shim, S. O., Chua, T. J., & Cai, T. X.** (2018). Multi-level job scheduling under processing time uncertainty. *Computers & Industrial Engineering*, 120, 480-487.
- Karaj, A.** (2023). Kardiyovasküler hastalıklarının teşhisine yönelik makine öğrenmesi algoritmaları ile karar destek sistemi tasarımı.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y.** (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Khaled, M. S., Shaban, I. A., Karam, A., Hussain, M., Zahran, I., & Hussein, M.** (2022). An Analysis of Research Trends in the Sustainability of Production Planning. *Energies*, 15(2), 483.
- Kuhn, M., & Johnson, K.** (2013). Applied predictive modeling (Vol. 26, p. 13). New York: Springer.

- Singh, M. K.** (2014). Industrial practices in weaving preparatory. Woodhead Publishing India Pvt. Limited.
- Kunnathur AS and Gupta SK** (1990). Minimizing the makespan with late start penalties added to processing times in a Lei, D. (2011). Population-based neighborhood search for job shop scheduling with interval processing time. *Computers & Industrial Engineering*, 61(4), 1200-1208.
- KURNAZ, F. S.** (2020). Temel Bileşenlere Dayalı Bazı Regresyon Yöntemleri. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 10(3), 581-591.
- Li, X., Gao, L., Wang, W., Wang, C., & Wen, L.** (2019). Particle swarm optimization hybridized with genetic algorithm for uncertain integrated process planning and scheduling with interval processing time. *Computers & Industrial Engineering*, 135, 1036-1046.
- LightGBM documentantion,** (2022). Lightgbm. [lightgbm.readthedocs.io](https://lightgbm.readthedocs.io/en/v3.3.2/)
<https://lightgbm.readthedocs.io/en/v3.3.2/>
- Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., & Monostori, L.** (2018). Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia Cirp*, 72, 1051-1056.
- Lödding, H.** (2005). Verfahren der Fertigungssteuerung (Vol. 2). Berlin: Springer.
- Machado, M. R., Karray, S., & de Sousa, I. T.** (2019, August). LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry. In *2019 14th International Conference on Computer Science & Education (ICCSE)* (pp. 1111-1116). IEEE.
- Masand, B., Linoff, G., & Waltz, D.** (1992, June). Classifying news stories using memory based reasoning. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 59-65).
- Meidan, Y., Lerner, B., Rabinowitz, G., & Hassoun, M.** (2011). Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining. *IEEE transactions on semiconductor manufacturing*, 24(2), 237-248.
- Melnikov, A., Kordzanganeh, M., Alodjants, A., & Lee, R. K.** (2023). Quantum machine learning: From physics to software engineering. *Advances in Physics: X*, 8(1), 2165452.
- Bakanlığı, T. M. E.** (2007). MEGEP (Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi). Eğlence Hizmetleri Hareket ve Ritim, Ankara.
- Moore, R. E.** (1979). Methods and applications of interval analysis. Society for Industrial and Applied Mathematics.
- Mosheiov, G.** (1996). Λ -shaped policies to schedule deteriorating jobs. *Journal of the Operational Research Society*, 47(9), 1184-1191.

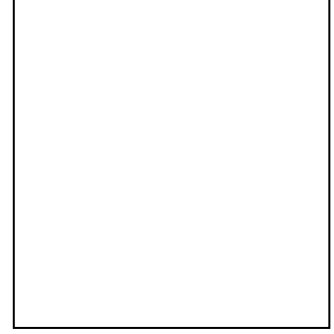
- Muratlar, E.R.,** (2020). Lightgbm. veribilimiokulu.com. <https://www.veribilimiokulu.com/lightgbm/>
- Nagasawa, K., Ikeda, Y., & Irohara, T.** (2015). Robust flow shop scheduling with random processing times for reduction of peak power consumption. *Simulation modelling practice and theory*, 59, 102-113.
- Navas Juan,** (2022). What is hyperparameter tuning?. [www.anyscale.com](https://www.anyscale.com/blog/what-is-hyperparameter-tuning) <https://www.anyscale.com/blog/what-is-hyperparameter-tuning>
- Nyhuis, P., & Wiendahl, H. P.** (2012). *Logistische Kennlinien: Grundlagen, Werkzeuge und Anwendungen*. Springer-Verlag.
- Quinlan, J. R.** (1986). Induction of decision trees. *Machine learning*, 1, 81-106.
- Quinlan, J. R.** (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3), 221-234.
- Ouyang, F. S., Guo, B. L., Ouyang, L. Z., Liu, Z. W., Lin, S. J., Meng, W., ... & Yang, S. M.** (2019). Comparison between linear and nonlinear machine-learning algorithms for the classification of thyroid nodules. *European journal of radiology*, 113, 251-257.
- Osmanoğlu, U. Ö.,** (2021). Makine Öğrenmesi Sınıflama Algoritmalarıyla Kalp Yetersizliği Mortalitesinin Tahminlemesi, Eskişehir.
- Öztemel, E.** (2012). *Yapay Sinir Ağları*, Papatya Yayıncılık Eğitim, 3. Basım, İstanbul.
- Özdemir, E. S.** (2019). Makine Öğrenmesi Kullanımıyla Ev Tipi Klimalarda Parametre Ve Enerji Verimliliği Optimizasyonu (Doctoral dissertation, Enerji Enstitüsü).
- Phatak, A., & De Jong, S.** (1997). The geometry of partial least squares. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 11(4), 311-338.
- Raaymakers, W. H., & Weijters, A. J. M. M.** (2003). Makespan estimation in batch process industries: A comparison between regression analysis and neural networks. *European Journal of Operational Research*, 145(1), 14-30.
- Rainer, C.** (2013). Data mining as technique to generate planning rules for manufacturing control in a complex production system: A case study from a manufacturer of aluminum products. In *Robust Manufacturing Control: Proceedings of the CIRP Sponsored Conference RoMaC 2012*, Bremen, Germany, 18th-20th June 2012 (pp. 203-214). Springer Berlin Heidelberg.
- Sakawa, M., & Mori, T.** (1999). An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Computers & industrial engineering*, 36(2), 325-341.
- Scikit-learn developers,** (2007,2023). Linear Models. [Scikit-learn.org. https://scikit-learn.org/stable/modules/linear_model.html#elastic-net](https://scikit-learn.org/stable/modules/linear_model.html#elastic-net)
- Seçkin, M., Seçkin, A. Ç., & Coşkun, A.** (2019). Production fault simulation and forecasting from time series data with machine learning in glove textile industry. *Journal of Engineered Fibers and Fabrics*, 14, 1558925019883462.

- Scher, S., & Messori, G.** (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, 144(717), 2830-2841.
- Schuh, G., & Stich, V.** (2012). Konzeptentwicklung in der Produktionsplanung und-steuerung. *Produktionsplanung und-steuerung 2: Evolution der PPS*, 149-415.
- Shrivastava, N. A., Khosravi, A., & Panigrahi, B. K.** (2015). Prediction interval estimation of electricity prices using PSO-tuned support vector machines. *IEEE Transactions on Industrial Informatics*, 11(2), 322-331.
- Singh, A., Thakur, N., & Sharma, A.** (2016, March). A review of supervised machine learning algorithms. In 2016 3rd international conference on computing for sustainable global development (INDIACom) (pp. 1310-1315). Ieee.
- Sun, C., Dominguez-Caballero, J., Ward, R., Ayvar-Soberanis, S., & Curtis, D.** (2022). Machining cycle time prediction: Data-driven modelling of machine tool feedrate behavior with neural networks. *Robotics and Computer-Integrated Manufacturing*, 75, 102293.
- Sümer, K. K.** (2022). Endüstri 4.0 Açısından İstatistiki Makine Öğrenmesi Ve Büyük Verinin Firmaların Rekabet Gücüne Etkisinin İncelenmesi. *Eurasian Econometrics Statistics & Emprical Economics Journal*, 107-120.
- Şanlıtürk, E.** (2018). Makine Öğrenme Algoritmalarıyla Hatalı Ürün Tahmini. Published Master's Thesis). İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü İşletme Mühendisliği Anabilim Dalı, İstanbul.
- Thomas, L. J., & McClain, J. O.** (1993). An overview of production planning. *Handbooks in operations research and management science*, 4, 333-370.
- Tibshirani, R.** (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267-288.
- Tirkel, I.** (2011, May). Cycle time prediction in wafer fabrication line by applying data mining methods. In 2011 IEEE/SEMI Advanced Semiconductor Manufacturing Conference (pp. 1-5). IEEE.
- Torkul, O., Gülseçen, S., Uyaroğlu, Y., Çağıl, G., & Uçar, M. K.** (2017). Mühendislikte yapay zekâ ve uygulamaları.
- Xu, Y., & Goodacre, R.** (2018). On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing*, 2(3), 249-262.
- Villalobos-Arias, L., Quesada-López, C., Guevara-Coto, J., Martínez, A., & Jenkins, M.** (2020, November). Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation. In *Proceedings of the 16th ACM international conference on predictive models and data analytics in software engineering* (pp. 31-40).

- Wang, J., Yang, J., Zhang, J., Wang, X., & Zhang, W.** (2018). Big data driven cycle time parallel prediction for production planning in wafer manufacturing. *Enterprise information systems*, 12(6), 714-732.
- Wang, J., Zhang, J., & Wang, X.** (2018). A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. *IEEE Transactions on Semiconductor Manufacturing*, 31(1), 173-182.7
- Wold, H.** (1966). Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, 391-420.
- Zarte, M., Pechmann, A., & Nunes, I. L.** (2021). Fuzzy Inference model for decision support in sustainable production planning processes—a case study. *Sustainability*, 13(3), 1355.
- Zou, H., & Hastie, T.** (2003). Regression shrinkage and selection via the elastic net, with applications to microarrays. *JR Stat Soc Ser B*, 67, 301-20.



ÖZGEÇMİŞ



Ad-Soyad : Müzeyyen Göksel

Doğum Tarihi ve Yeri :

E-posta :

ÖĞRENİM DURUMU:

- **Lisans** : 2021, Osmangazi Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü
- **Yüksek Lisans** : 2023, Bursa Teknik Üniversitesi, Akıllı Sistemler Mühendisliği Anabilim Dalı, Yüksek Lisans Program

MESLEKİ DENEYİM VE ÖDÜLLER:

- Termal Seramik, Planlama ve Bilgi İşlem Mühendisi
- Küçükçalık Tekstil A.Ş., Sistem ve Süreç Geliştirme Yönetmeni
- İbrahim Ortakçı Cam A.Ş., Operasyon Müdürü
- Doğtaş Kelebek Mobilya, Dış Ticaret Kalite Müdürü

TEZDEN TÜRETİLEN ESERLER, SUNUMLAR VE PATENTLER:

Demir Y., Göksel M., (2023). Çözümleri Hazırlama İşlem Süresinin Doğrusal Regresyon Modelleri ile Tahmini, 1st International Conference on Pioneer and Innovative Studies, June 5-7, 2023